Selective Question Answering under Domain Shift

Amita Kamath **Robin Jia** **Percy Liang**
Computer Science Department, Stanford University
{kamatha, robinjia, pliang}@cs.stanford.edu

Abstract

To avoid giving wrong answers, question answering (QA) models need to know when to abstain from answering. Moreover, users often ask questions that diverge from the model's training data, making errors more likely and thus abstention more critical. In this work, we propose the setting of selective question answering under domain shift, in which a QA model is tested on a mixture of in-domain and out-of-domain data, and must answer (i.e., not abstain on) as many questions as possible while maintaining high accuracy. Abstention policies based solely on the model's softmax probabilities fare poorly, since models are overconfident on out-of-domain inputs. Instead, we train a calibrator to identify inputs on which the QA model errs, and abstain when it predicts an error is likely. Crucially, the calibrator benefits from observing the model's behavior on out-of-domain data, even if from a different domain than the test data. We combine this method with a SQuAD-trained QA model and evaluate on mixtures of SQuAD and five other QA datasets. Our method answers 56% of questions while maintaining 80% accuracy; in contrast, directly using the model's probabilities only answers 48% at 80% accuracy.

1 Introduction

Question answering (QA) models have achieved impressive performance when trained and tested on examples from the same dataset, but tend to perform poorly on examples that are out-of-domain (OOD) (Jia and Liang, 2017; Chen et al., 2017; Yogatama et al., 2019; Talmor and Berant, 2019; Fisch et al., 2019). Deployed QA systems in search engines and personal assistants need to gracefully handle OOD inputs, as users often ask questions that fall outside of the system's training distribution. While the ideal system would correctly answer all

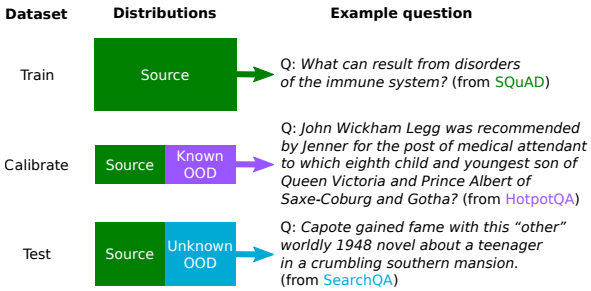

Figure 1: Selective question answering under domain shift with a trained calibrator. First, a QA model is trained only on source data. Then, a calibrator is trained to predict whether the QA model was correct on any given example. The calibrator's training data consists of both previously held-out source data and known OOD data. Finally, the combined selective QA system is tested on a mixture of test data from the source distribution and an unknown OOD distribution.

OOD questions, such perfection is not attainable given limited training data (Geiger et al., 2019). Instead, we aim for a more achievable yet still challenging goal: models should *abstain* when they are likely to err, thus avoiding showing wrong answers to users. This general goal motivates the setting of selective prediction, in which a model outputs both a prediction and a scalar confidence, and abstains on inputs where its confidence is low (El-Yaniv and Wiener, 2010; Geifman and El-Yaniv, 2017).

In this paper, we propose the setting of **selective question answering under domain shift**, which captures two important aspects of real-world QA: (i) test data often diverges from the training distribution, and (ii) systems must know when to abstain. We train a QA model on data from a *source* distribution, then evaluate selective prediction performance on a dataset that includes samples from both the source distribution and an *unknown OOD* distribution. This mixture simulates the likely scenario in which users only sometimes ask questions that are covered by the training distribution. While the sys-

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5684–5696
July 5 - 10, 2020.

tem developer knows nothing about the unknown OOD data, we allow access to a small amount of data from a third *known OOD* distribution (e.g., OOD examples that they can foresee).

We first show that our setting is challenging because model softmax probabilities are unreliable estimates of confidence on out-of-domain data. Prior work has shown that a strong baseline for in-domain selective prediction is MaxProb, a method that abstains based on the probability assigned by the model to its highest probability prediction (Hendrycks and Gimpel, 2017; Lakshminarayanan et al., 2017). We find that MaxProb gives good confidence estimates on in-domain data, but is overconfident on OOD data. Therefore, MaxProb performs poorly in mixed settings: it does not abstain enough on OOD examples, relative to in-domain examples.

We correct for MaxProb's overconfidence by using known OOD data to train a *calibrator*—a classifier trained to predict whether the original QA model is correct or incorrect on a given example (Platt, 1999; Zadrozny and Elkan, 2002). While prior work in NLP trains a calibrator on in-domain data (Dong et al., 2018), we show this does not generalize to unknown OOD data as well as training on a mixture of in-domain and known OOD data. Figure 1 illustrates the problem setup and how the calibrator uses known OOD data. We use a simple random forest calibrator over features derived from the input example and the model's softmax outputs.

We conduct extensive experiments using SQuAD (Rajpurkar et al., 2016) as the source distribution and five other QA datasets as different OOD distributions. We average across all 20 choices of using one as the unknown OOD dataset and another as the known OOD dataset, and test on a uniform mixture of SQuAD and unknown OOD data. On average, the trained calibrator achieves 56.1% coverage (i.e., the system answers 56.1% of test questions) while maintaining 80% accuracy on answered questions, outperforming MaxProb with the same QA model (48.2% coverage at 80% accuracy), using MaxProb and training the QA model on both SQuAD and the known OOD data (51.8% coverage), and training the calibrator only on SQuAD data (53.7% coverage).

In summary, our contributions are as follows:

(1) We propose a novel setting, selective question answering under domain shift, that captures the practical necessity of knowing when to abstain on test data that differs from the training data.

(2) We show that QA models are overconfident on out-of-domain examples relative to in-domain examples, which causes MaxProb to perform poorly in our setting.

(3) We show that out-of-domain data, even from a different distribution than the test data, can improve selective prediction under domain shift when used to train a calibrator.

2 Related Work

Our setting combines extrapolation to out-of-domain data with selective prediction. We also distinguish our setting from the tasks of identifying unanswerable questions and outlier detection.

2.1 Extrapolation to out-of-domain data

Extrapolating from training data to test data from a different distribution is an important challenge for current NLP models (Yogatama et al., 2019). Models trained on many domains may still struggle to generalize to new domains, as these may involve new types of questions or require different reasoning skills (Talmor and Berant, 2019; Fisch et al., 2019). Related work on domain adaptation also tries to generalize to new distributions, but assumes some knowledge about the test distribution, such as unlabeled examples or a few labeled examples (Blitzer et al., 2006; Daume III, 2007); we assume no such access to the test distribution, but instead make the weaker assumption of access to samples from a different OOD distribution.

2.2 Selective prediction

Selective prediction, in which a model can either predict or abstain on each test example, is a long-standing research area in machine learning (Chow, 1957; El-Yaniv and Wiener, 2010; Geifman and El-Yaniv, 2017). In NLP, Dong et al. (2018) use a calibrator to obtain better confidence estimates for semantic parsing. Rodriguez et al. (2019) use a similar approach to decide when to answer QuizBowl questions. These works focus on training and testing models on the same distribution, whereas our training and test distributions differ.

Selective prediction under domain shift. Other fields have recognized the importance of selective prediction under domain shift. In medical applications, models may be trained and tested on different groups of patients, so selective prediction is needed to avoid costly errors (Feng et al., 2019). In computational chemistry, Toplak et al. (2014) use

selective prediction techniques to estimate the set of (possibly out-of-domain) molecules for which a reactivity classifier is reliable. To the best of our knowledge, our work is the first to study selective prediction under domain shift in NLP.

Answer validation. Traditional pipelined systems for open-domain QA often have dedicated systems for answer validation—judging whether a proposed answer is correct. These systems often rely on external knowledge about entities (Magnini et al., 2002; Ko et al., 2007). Knowing when to abstain has been part of past QA shared tasks like RespubliQA (Peñas et al., 2009) and QA4MRE (Peñas et al., 2013). IBM's Watson system for Jeopardy also uses a pipelined approach for answer validation (Gondek et al., 2012). Our work differs by focusing on modern neural QA systems trained end-to-end, rather than pipelined systems, and by viewing the problem of abstention in QA through the lens of selective prediction.

2.3 Related goals and tasks

Calibration. Knowing when to abstain is closely related to calibration—having a model's output probability align with the true probability of its prediction (Platt, 1999). A key distinction is that selective prediction metrics generally depend only on *relative* confidences—systems are judged on their ability to rank correct predictions higher than incorrect predictions (El-Yaniv and Wiener, 2010). In contrast, calibration error depends on the absolute confidence scores. Nonetheless, we will find it useful to analyze calibration in Section 5.3, as miscalibration on some examples but not others does imply poor relative ordering, and therefore poor selective prediction. Ovadia et al. (2019) observe increases in calibration error under domain shift.

Identifying unanswerable questions. In SQuAD 2.0, models must recognize when a paragraph does not entail an answer to a question (Rajpurkar et al., 2018). Sentence selection systems must rank passages that answer a question higher than passages that do not (Wang et al., 2007; Yang et al., 2015). In these cases, the goal is to "abstain" when *no* system (or person) could infer an answer to the given question using the given passage. In contrast, in selective prediction, the model should abstain when *it* would give a wrong answer if forced to make a prediction.

Outlier detection. We distinguish selective prediction under domain shift from outlier detection, the task of detecting out-of-domain examples (Schölkopf et al., 1999; Hendrycks and Gimpel, 2017; Liang et al., 2018). While one could use an outlier detector for selective classification (e.g., by abstaining on all examples flagged as outliers), this would be too conservative, as QA models can often get a non-trivial fraction of OOD examples correct (Talmor and Berant, 2019; Fisch et al., 2019). Hendrycks et al. (2019b) use known OOD data for outlier detection by training models to have high entropy on OOD examples; in contrast, our setting rewards models for predicting correctly on OOD examples, not merely having high entropy.

3 Problem Setup

We formally define the setting of selective prediction under domain shift, starting with some notation for selective prediction in general.

3.1 Selective Prediction

Given an input x, the selective prediction task is to output $(\hat{y}, c)$ where $\hat{y} \in Y(x)$, the set of answer candidates, and $c \in \mathbb{R}$ denotes the model's confidence. Given a threshold $\gamma \in \mathbb{R}$, the overall system predicts $\hat{y}$ if $c \geq \gamma$ and abstain otherwise.

The risk-coverage curve provides a standard way to evaluate selective prediction methods (El-Yaniv and Wiener, 2010). For a test dataset D_{test}, any choice of γ has an associated *coverage*—the fraction of D_{test} the model makes a prediction on—and *risk*—the error on that fraction of D_{test}. As γ decreases, coverage increases, but risk will usually also increase. We plot risk versus coverage and evaluate on the area under this curve (AUC), as well as the maximum possible coverage for a desired risk level. The former metric averages over all γ, painting an overall picture of selective prediction performance, while the latter evaluates at a particular choice of γ corresponding to a specific level of risk tolerance.

3.2 Selective Prediction under Domain Shift

We deviate from prior work by considering the setting where the model's training data D_{train} and test data D_{test} are drawn from different distributions. As our experiments demonstrate, this setting is challenging because standard QA models are overconfident on out-of-domain inputs.

To formally define our setting, we specify three

data distributions. First, p_{source} is the source distribution, from which a large training dataset D_{train} is sampled. Second, q_{unk} is an *unknown OOD distribution*, representing out-of-domain data encountered at test time. The test dataset D_{test} is sampled from p_{test}, a mixture of p_{source} and q_{unk}:

$$p_{\text{test}} = \alpha p_{\text{source}} + (1 - \alpha) q_{\text{unk}} \quad (1)$$

for $\alpha \in (0, 1)$. We choose $\alpha = \frac{1}{2}$, and examine the effect of changing this ratio in Section 5.8. Third, q_{known} is a *known OOD distribution*, representing examples not in p_{source} but from which the system developer has a small dataset D_{calib}.

3.3 Selective Question Answering

While our framework is general, we focus on extractive question answering, as exemplified by SQuAD (Rajpurkar et al., 2016), due to its practical importance and the diverse array of available QA datasets in the same format. The input x is a passage-question pair (p, q), and the set of answer candidates $Y(x)$ is all spans of the passage p. A *base model* f defines a probability distribution $f(y \mid x)$ over $Y(x)$. All selective prediction methods we consider choose $\hat{y} = \arg\max_{y' \in Y(x)} f(y' \mid x)$, but differ in their associated confidence c.

4 Methods

Recall that our setting differs from the standard selective prediction setting in two ways: unknown OOD data drawn from q_{unk} appears at test time, and known OOD data drawn from q_{known} is available to the system. Intuitively, we expect that systems must use the known OOD data to generalize to the unknown OOD data. In this section, we present three standard selective prediction methods for in-domain data, and show how they can be adapted to use data from q_{known}.

4.1 MaxProb

The first method, MaxProb, directly uses the probability assigned by the base model to $\hat{y}$ as an estimate of confidence. Formally, MaxProb with model f estimates confidence on input x as:

$$c_{\text{MaxProb}} = f(\hat{y} \mid x) = \max_{y' \in Y(x)} f(y' \mid x). \quad (2)$$

MaxProb is a strong baseline for our setting. Across many tasks, MaxProb has been shown to distinguish in-domain test examples that the model gets right from ones the model gets wrong (Hendrycks and Gimpel, 2017). MaxProb is also a strong baseline for outlier detection, as it is lower for out-of-domain examples than in-domain examples (Lakshminarayanan et al., 2017; Liang et al., 2018; Hendrycks et al., 2019b). This is desirable for our setting: models make more mistakes on OOD examples, so they should abstain more on OOD examples than in-domain examples.

MaxProb can be used with any base model f. We consider two such choices: a model f_{src} trained only on D_{train}, or a model $f_{\text{src+known}}$ trained on the union of D_{train} and D_{calib}.

4.2 Test-time Dropout

For neural networks, another standard approach to estimate confidence is to use dropout at test time. Gal and Ghahramani (2016) showed that dropout gives good confidence estimates on OOD data.

Given an input x and model f, we compute f on x with K different dropout masks, obtaining prediction distributions $\hat{p}_1, \ldots, \hat{p}_K$, where each $\hat{p}_i$ is a probability distribution over $Y(x)$. We consider two statistics of these $\hat{p}_i$'s that are commonly used as confidence estimates. First, we take the mean of $\hat{p}_i(\hat{y})$ across all i (Lakshminarayanan et al., 2017):

$$c_{\text{DropoutMean}} = \frac{1}{K} \sum_{i=1}^{K} \hat{p}_i(\hat{y}). \quad (3)$$

This can be viewed as ensembling the predictions across all K dropout masks by averaging them.

Second, we take the negative variance of the $\hat{p}_i(\hat{y})$'s (Feinman et al., 2017; Smith and Gal, 2018):

$$c_{\text{DropoutVar}} = -\text{Var}[\hat{p}_1(\hat{y}), \ldots, \hat{p}_K(\hat{y})]. \quad (4)$$

Higher variance corresponds to greater uncertainty, and hence favors abstaining. Like MaxProb, dropout can be used either with f trained only on D_{train}, or on both D_{train} and the known OOD data.

Test-time dropout has practical disadvantages compared to MaxProb. It requires access to internal model representations, whereas MaxProb only requires black box access to the base model (e.g., API calls to a trained model). Dropout also requires K forward passes of the base model, leading to a K-fold increase in runtime.

4.3 Training a calibrator

Our final method trains a calibrator to predict when a base model (trained only on data from p_{source}) is

correct (Platt, 1999; Dong et al., 2018). We differ from prior work by training the calibrator on a mixture of data from p_{source} and q_{known}, anticipating the test-time mixture of p_{source} and q_{unk}. More specifically, we hold out a small number of p_{source} examples from base model training, and train the calibrator on the union of these examples and the q_{known} examples. We define $c_{\text{Calibrator}}$ to be the prediction probability of the calibrator.

The calibrator itself could be any binary classification model. We use a random forest classifier with seven features: passage length, the length of the predicted answer $\hat{y}$, and the top five softmax probabilities output by the model. These features require only a minimal amount of domain knowledge to define. Rodriguez et al. (2019) similarly used multiple softmax probabilities to decide when to answer questions. The simplicity of this model makes the calibrator fast to train when given new data from q_{known}, especially compared to retraining the QA model on that data.

We experiment with four variants of the calibrator. First, to measure the impact of using known OOD data, we change the calibrator's training data: it can be trained either on data from p_{source} only, or both p_{source} and q_{known} data as described. Second, we consider a modification where instead of the model's probabilities, we use probabilities from the mean ensemble over dropout masks, as described in Section 4.2, and also add $c_{\text{DropoutVar}}$ as a feature. As discussed above, dropout features are costly to compute and assume white-box access to the model, but may result in better confidence estimates. Both of these variables can be changed independently, leading to four configurations.

5 Experiments and Analysis

5.1 Experimental Details

Data. We use SQuAD 1.1 (Rajpurkar et al., 2016) as the source dataset and five other datasets as OOD datasets: NewsQA (Trischler et al., 2017), TriviaQA (Joshi et al., 2017), SearchQA (Dunn et al., 2017), HotpotQA (Yang et al., 2018), and Natural Questions (Kwiatkowski et al., 2019).[1] These are all extractive question answering datasets where all questions are answerable; however, they vary widely in the nature of passages (e.g., Wikipedia, news, web snippets), questions (e.g., Jeopardy and trivia questions), and relationship between passages and questions (e.g., whether questions are written based on passages, or passages retrieved based on questions). We used the preprocessed data from the MRQA 2019 shared task (Fisch et al., 2019). For HotpotQA, we focused on multi-hop questions by selecting only "hard" examples, as defined by Yang et al. (2018). In each experiment, two different OOD datasets are chosen as q_{known} and q_{unk}. All results are averaged over all 20 such combinations, unless otherwise specified. We sample 2,000 examples from q_{known} for D_{calib}, and 4,000 SQuAD and 4,000 q_{unk} examples for D_{test}. We evaluate using exact match (EM) accuracy, as defined by SQuAD (Rajpurkar et al., 2016). Additional details can be found in Appendix A.1.

QA model. For our QA model, we use the BERT-base SQuAD 1.1 model trained for 2 epochs (Devlin et al., 2019). We train six models total: one f_{src} and five $f_{\text{src+known}}$'s, one for each OOD dataset.

Selective prediction methods. For test-time dropout, we use $K = 30$ different dropout masks, as in Dong et al. (2018). For our calibrator, we use the random forest implementation from Scikit-learn (Pedregosa et al., 2011). We train on 1,600 SQuAD examples and 1,600 known OOD examples, and use the remaining 400 SQuAD and 400 known OOD examples as a validation set to tune calibrator hyperparameters via grid search. We average our results over 10 random splits of this data. When training the calibrator only on p_{source}, we use 3,200 SQuAD examples for training and 800 for validation, to ensure equal dataset sizes. Additional details can be found in Appendix A.2.

5.2 Main results

Training a calibrator with q_{known} outperforms other methods. Table 1 compares all methods that do not use test-time dropout. Compared to MaxProb with $f_{\text{src+known}}$, the calibrator has 4.3 points and 6.7 points higher coverage at 80% and 90% accuracy respectively, and 1.1 points lower AUC.[2] This demonstrates that training a calibrator is a better use of known OOD data than training a QA model. The calibrator trained on both p_{source} and q_{known} also outperforms the calibrator trained on p_{source} alone by 2.4% coverage at 80% accuracy. All methods perform far worse than the optimal selective predictor with the given base model, though

[1] We consider these different datasets to represent different domains, hence our usage of the term "domain shift."

[2] 95% confidence interval is [1.01, 1.69], using the paired bootstrap test with 1000 bootstrap samples.

	AUC ↓	Cov @ Acc=80% ↑	Cov @ Acc=90% ↑
Train QA model on SQuAD			
MaxProb	20.54	48.23	21.07
Calibrator (p_{source} only)	19.27	53.67	26.68
Calibrator (p_{source} and q_{known})	**18.47**	**56.06**	**29.42**
Best possible	9.64	74.92	66.59
Train QA model on SQuAD + known OOD			
MaxProb	19.61	51.75	22.76
Best possible	8.83	76.80	68.26

Table 1: Results for methods without test-time dropout. The calibrator with access to q_{known} outperforms all other methods. ↓: lower is better. ↑: higher is better.

	AUC ↓	Cov @ Acc=80% ↑	Cov @ Acc=90% ↑
Train QA model on SQuAD			
Test-time dropout (–var)	28.13	24.50	15.40
Test-time dropout (mean)	18.35	57.49	29.55
Calibrator (p_{source} only)	17.84	58.35	34.27
Calibrator (p_{source} and q_{known})	**17.31**	**59.99**	**34.99**
Best possible	9.64	74.92	66.59
Train QA model on SQuAD + known OOD			
Test-time dropout (–var)	26.67	26.74	15.95
Test-time dropout (mean)	17.72	59.60	30.40
Best possible	8.83	76.80	68.26

Table 2: Results for methods that use test-time dropout. Here again, the calibrator with access to q_{known} outperforms all other methods.

achieving this bound may not be realistic.[3]

Test-time dropout improves results but is expensive. Table 2 shows results for methods that use test-time dropout, as described in Section 4.2. The negative variance of $\hat{p}_i(\hat{y})$'s across dropout masks serves poorly as an estimate of confidence, but the mean performs well. The best performance is attained by the calibrator using dropout features, which has 3.9% higher coverage at 80% accuracy than the calibrator with non-dropout features. Since test-time dropout introduces substantial (i.e., K-fold) runtime overhead, our remaining analyses focus on methods without test-time dropout.

The QA model has lower non-trivial accuracy on OOD data. Next, we motivate our focus on selective prediction, as opposed to outlier detection, by showing that the QA model still gets a non-trivial fraction of OOD examples correct. Table 3 shows the (non-selective) exact match scores

[3]As the QA model has fixed accuracy $< 100\%$ on D_{test}, it is impossible to achieve 0% risk at 100% coverage.

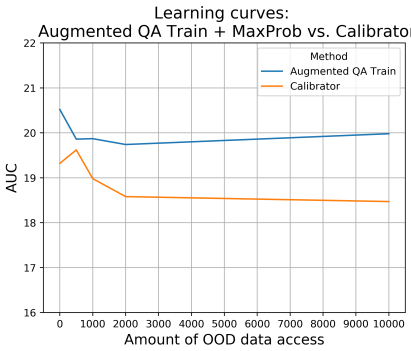

Figure 2: Area under the risk-coverage curve as a function of how much data from q_{known} is available. At all points, using data from q_{known} to train the calibrator is more effective than using it for QA model training.

for all six QA models used in our experiments on all datasets. All models get around 80% accuracy on SQuAD, and around 40% to 50% accuracy on most OOD datasets. Since OOD accuracies are much higher than 0%, abstaining on all OOD examples would be overly conservative.[4] At the same time, since OOD accuracy is worse than in-domain accuracy, a good selective predictor should answer more in-domain examples and fewer OOD examples. Training on 2,000 q_{known} examples does not significantly help the base model extrapolate to other q_{unk} distributions.

Results hold across different amounts of known OOD data. As shown in Figure 2, across all amounts of known OOD data, using it to train and validate the calibrator (in an 80–20 split) performs better than adding all of it to the QA training data and using MaxProb.

5.3 Overconfidence of MaxProb

We now show why MaxProb performs worse in our setting compared to the in-domain setting: it is miscalibrated on out-of-domain examples. Figure 3a shows that MaxProb values are generally lower for OOD examples than in-domain examples, following previously reported trends (Hendrycks and Gimpel, 2017; Liang et al., 2018). However, the MaxProb values are still too high out-of-domain. Figure 3b shows that MaxProb is not well calibrated: it is underconfident in-domain, and overconfident out-of-domain.[5] For example, for a Max-

[4]In Section A.3, we confirm that an outlier detector does not achieve good selective prediction performance.

[5]The in-domain underconfidence is because SQuAD (and some other datasets) provides only one answer at training time, but multiple answers are considered correct at test time. In Ap-

Train Data ↓ / Test Data →	SQuAD	TriviaQA	HotpotQA	NewsQA	Natural Questions	SearchQA
SQuAD only	80.95	48.43	44.88	40.45	42.78	17.98
SQuAD + 2K TriviaQA	81.48	(50.50)	43.95	39.15	47.05	25.23
SQuAD + 2K HotpotQA	81.15	49.35	(53.60)	39.85	48.18	24.40
SQuAD + 2K NewsQA	81.50	50.18	42.88	(44.00)	47.08	20.40
SQuAD + 2K NaturalQuestions	81.48	51.43	44.38	40.90	(54.85)	25.95
SQuAD + 2K SearchQA	81.60	56.58	44.30	40.15	47.05	(59.80)

Table 3: Exact match accuracy for all six QA models on all six test QA datasets. Training on D_{calib} improves accuracy on data from the same dataset (diagonal), but generally does not improve accuracy on data from q_{unk}.

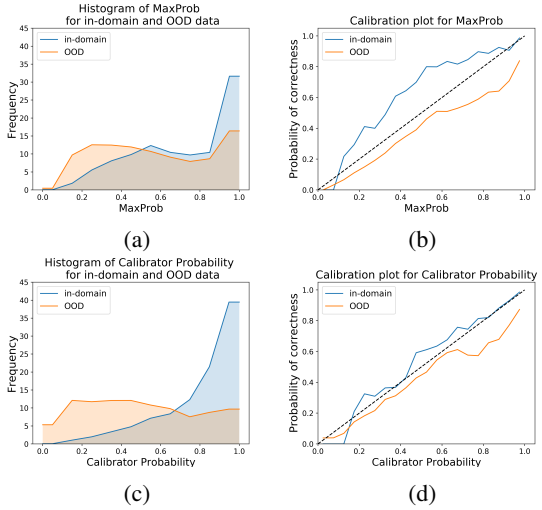

Figure 3: MaxProb is lower on average for OOD data than in-domain data (a), but it is still overconfident on OOD data: when plotting the true probability of correctness vs. MaxProb (b), the OOD curve is below the $y = x$ line, indicating MaxProb overestimates the probability that the prediction is correct. The calibrator assigns lower confidence on OOD data (c) and has a smaller gap between in-domain and OOD curves (d), indicating improved calibration.

Prob of 0.6, the model is about 80% likely to get the question correct if it came from SQuAD (in-domain), and 45% likely to get the question correct if it was OOD. When in-domain and OOD examples are mixed at test time, MaxProb therefore does not abstain enough on the OOD examples. Figure 3d shows that the calibrator is better calibrated, even though it is not trained on any unknown OOD data. In Appendix A.5, we show that the calibrator abstains on more OOD examples than MaxProb.

Our finding that the BERT QA model is not overconfident in-domain aligns with Hendrycks et al. (2019a), who found that pre-trained computer vision models are better calibrated than models trained from scratch, as pre-trained models can be trained for fewer epochs. Our QA model is only trained for two epochs, as is standard for BERT. Our findings also align with Ovadia et al. (2019), who find that computer vision and text classification models are poorly calibrated out-of-domain even when well-calibrated in-domain. Note that miscalibration out-of-domain does not imply poor selective prediction on OOD data, but does imply poor selective prediction in our mixture setting.

pendix A.4, we show that removing multiple answers makes MaxProb well-calibrated in-domain; it stays overconfident out-of-domain.

5.4 Extrapolation between datasets

We next investigated how choice of q_{known} affects generalization of the calibrator to q_{unk}. Figure 4 shows the percentage reduction between MaxProb and optimal AUC achieved by the trained calibrator. The calibrator outperforms MaxProb over all dataset combinations, with larger gains when q_{known} and q_{unk} are similar. For example, samples from TriviaQA help generalization to SearchQA and vice versa; both use web snippets as passages. Samples from NewsQA, the only other non-Wikipedia dataset, are also helpful for both. On the other hand, no other dataset significantly helps generalization to HotpotQA, likely due to HotpotQA's unique focus on multi-hop questions.

5.5 Calibrator feature ablations

We determine the importance of each feature of the calibrator by removing each of its features individually, leaving the rest. From Table 4, we see that the most important features are the softmax probabilities and the passage length. Intuitively, passage length is meaningful both because longer passages have more answer candidates, and because passage length differs greatly between different domains.

5.6 Error analysis

We examined calibrator errors on two pairs of q_{known} and q_{unk}—one similar pair of datasets and one dissimilar. For each, we sampled 100 errors in which the system confidently gave a wrong answer (overconfident), and 100 errors in which the sys-

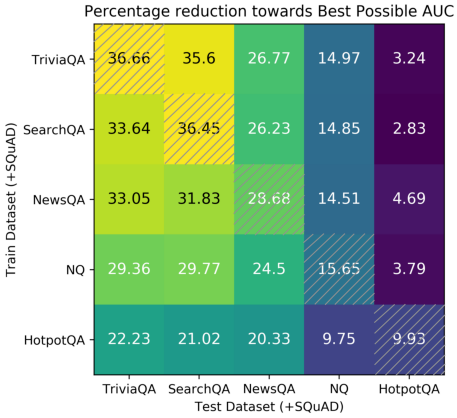

Figure 4: Results for different choices of q_{known} (y-axis) and q_{unk} (x-axis). For each pair, we report the percent AUC improvement of the trained calibrator over MaxProb, relative to the total possible improvement. Datasets that use similar passages (e.g., SearchQA and TriviaQA) help each other the most. Main diagonal elements (shaded) assume access to q_{unk} (see Section 5.9).

	AUC ↓	Cov @ Acc=80% ↑	Cov @ Acc=90% ↑
All features	**18.47**	**56.06**	**29.42**
–Top softmax probability	18.61	55.46	29.27
–2nd:5th highest softmax probabilities	19.11	54.29	26.67
–All softmax probabilities	26.41	24.57	0.08
–Context length	19.79	51.73	24.24
–Prediction length	18.6	55.67	29.30

Table 4: Performance of the calibrator as each of its features is removed individually, leaving the rest. The base model's softmax probabilities are important features, as is passage length.

tem abstained but would have gotten the question correct if it had answered (underconfident). These were sampled from the 1000 most overconfident or underconfident errors, respectively.

q_{known} = NewsQA, q_{unk} = TriviaQA. These two datasets are from different non-Wikipedia sources. 62% of overconfidence errors are due to the model predicting valid alternate answers, or span mismatches—the model predicts a slightly different span than the gold span, and should be considered correct; thus the calibrator was not truly overconfident. This points to the need to improve QA evaluation metrics (Chen et al., 2019). 45% of underconfidence errors are due to the passage requiring coreference resolution over long distances, including with the article title. Neither SQuAD nor NewsQA passages have coreference chains as long or contain titles, so it is unsurprising that the calibrator struggles on these cases. Another 25% of underconfidence errors were cases in which there was insufficient evidence in the paragraph to answer the question (as TriviaQA was constructed via distant supervision), so the calibrator was not incorrect to assign low confidence. 16% of all underconfidence errors also included phrases that would not be common in SQuAD and NewsQA, such as using "*said bye bye*" for "*banned.*"

q_{known} = NewsQA, q_{unk} = HotpotQA. These two datasets are dissimilar from each other in multiple ways. HotpotQA uses short Wikipedia passages and focuses on multi-hop questions; NewsQA has much longer passages from news articles and does not focus on multi-hop questions. 34% of the overconfidence errors are due to valid alternate answers or span mismatches. On 65% of the underconfidence errors, the correct answer was the only span in the passage that could plausibly answer the question, suggesting that the model arrived at the answer due to artifacts in HotpotQA that facilitate guesswork (Chen and Durrett, 2019; Min et al., 2019). In these situations, the calibrator's lack of confidence is therefore justifiable.

5.7 Relationship with Unanswerable Questions

We now study the relationship between selective prediction and identifying unanswerable questions.

Unanswerable questions do not aid selective prediction. We trained a QA model on SQuAD 2.0 (Rajpurkar et al., 2018), which augments SQuAD 1.1 with unanswerable questions. Our trained calibrator with this model gets 18.38 AUC, which is very close to the 18.47 for the model trained on SQuAD 1.1 alone. MaxProb also performed similarly with the SQuAD 2.0 model (20.81 AUC) and SQuAD 1.1 model (20.54 AUC).

Selective prediction methods do not identify unanswerable questions. For both MaxProb and our calibrator, we pick a threshold $\gamma' \in \mathbb{R}$ and predict that a question is unanswerable if the confidence $c < \gamma'$. We choose γ' to maximize SQuAD 2.0 EM score. Both methods perform poorly: the calibrator (averaged over five choices of q_{known}) achieves 54.0 EM, while MaxProb achieves 53.1 EM.[6] These results only weakly outperform the

[6]We evaluate on 4000 questions randomly sampled from the SQuAD 2.0 development set.

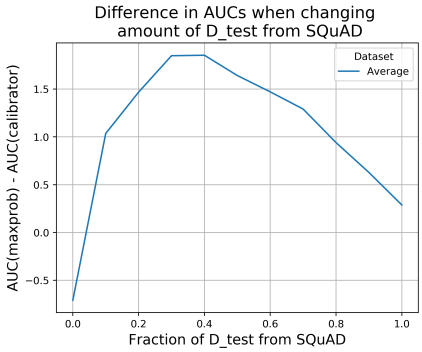

Figure 5: Difference in AUC between calibrator and MaxProb, as a function of how much of D_{test} comes from p_{source} (i.e., SQuAD) instead of q_{unk}, averaged over 5 OOD datasets. The calibrator outperforms MaxProb most when D_{test} is a mixture of p_{source} and q_{unk}.

majority baseline of 48.9 EM.

Taken together, these results indicate that identifying unanswerable questions is a very different task from knowing when to abstain under distribution shift. Our setting focuses on test data that is dissimilar to the training data, but on which the original QA model can still correctly answer a non-trivial fraction of examples. In contrast, unanswerable questions in SQuAD 2.0 look very similar to answerable questions, but a model trained on SQuAD 1.1 gets all of them wrong.

5.8 Changing ratio of in-domain to OOD

Until now, we used $\alpha = \frac{1}{2}$ both for D_{test} and training the calibrator. Now we vary α for both, ranging from using only SQuAD to only OOD data (sampled from q_{known} for D_{calib} and from q_{unk} for D_{test}).

Figure 5 shows the difference in AUC between the trained calibrator and MaxProb. At both ends of the graph, the difference is close to 0, showing that MaxProb performs well in homogeneous settings. However, when the two data sources are mixed, the calibrator outperforms MaxProb significantly. This further supports our claim that MaxProb performs poorly in mixed settings.

5.9 Allowing access to q_{unk}

We note that our findings do not hold in the alternate setting where we have access to samples from q_{unk} (instead of q_{known}). Training the QA model with this OOD data and using MaxProb achieves average AUC of 16.35, whereas training a calibrator achieves 17.87; unsurprisingly, training on examples similar to the test data is helpful. We do not focus on this setting, as our goal is to build selective QA models for unknown distributions.

6 Discussion

In this paper, we propose the setting of selective question answering under domain shift, in which systems must know when to abstain on a mixture of in-domain and unknown OOD examples. Our setting combines two important goals for real-world systems: knowing when to abstain, and handling distribution shift at test time. We show that models are overconfident on OOD examples, leading to poor performance in the our setting, but training a calibrator using other OOD data can help correct for this problem. While we focus on question answering, our framework is general and extends to any prediction task for which graceful handling of out-of-domain inputs is necessary.

Across many tasks, NLP models struggle on out-of-domain inputs. Models trained on standard natural language inference datasets (Bowman et al., 2015) generalize poorly to other distributions (Thorne et al., 2018; Naik et al., 2018). Achieving high accuracy on out-of-domain data may not even be possible if the test data requires abilities that are not learnable from the training data (Geiger et al., 2019). Adversarially chosen ungrammatical text can also cause catastrophic errors (Wallace et al., 2019; Cheng et al., 2020). In all these cases, a more intelligent model would recognize that it should abstain on these inputs.

Traditional NLU systems typically have a natural ability to abstain. SHRDLU recognizes statements that it cannot parse, or that it finds ambiguous (Winograd, 1972). QUALM answers reading comprehension questions by constructing reasoning chains, and abstains if it cannot find one that supports an answer (Lehnert, 1977).

NLP systems deployed in real-world settings inevitably encounter a mixture of familiar and unfamiliar inputs. Our work provides a framework to study how models can more judiciously abstain in these challenging environments.

Reproducibility. All code, data and experiments are available on the Codalab platform at `https://bit.ly/35inCah`.

Acknowledgments. This work was supported by the DARPA ASED program under FA8650-18-2-7882. We thank Ananya Kumar, John Hewitt, Dan Iter, and the anonymous reviewers for their helpful comments and insights.

References

J. Blitzer, R. McDonald, and F. Pereira. 2006. Domain adaptation with structural correspondence learning. In *Empirical Methods in Natural Language Processing (EMNLP)*.

S. Bowman, G. Angeli, C. Potts, and C. D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Empirical Methods in Natural Language Processing (EMNLP)*.

A. Chen, G. Stanovsky, S. Singh, and M. Gardner. 2019. Evaluating question answering evaluation. In *Workshop on Machine Reading for Question Answering (MRQA)*.

D. Chen, A. Fisch, J. Weston, and A. Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Association for Computational Linguistics (ACL)*.

J. Chen and G. Durrett. 2019. Understanding dataset design choices for multi-hop reasoning. In *North American Association for Computational Linguistics (NAACL)*.

M. Cheng, J. Yi, H. Zhang, P. Chen, and C. Hsieh. 2020. Seq2Sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

C. K. Chow. 1957. An optimum character recognition system using decision functions. In *IRE Transactions on Electronic Computers*.

H. Daume III. 2007. Frustratingly easy domain adaptation. In *Association for Computational Linguistics (ACL)*.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Association for Computational Linguistics (ACL)*, pages 4171–4186.

L. Dong, C. Quirk, and M. Lapata. 2018. Confidence modeling for neural semantic parsing. In *Association for Computational Linguistics (ACL)*.

M. Dunn, , L. Sagun, M. Higgins, U. Guney, V. Cirik, and K. Cho. 2017. SearchQA: A new Q&A dataset augmented with context from a search engine. *arXiv*.

R. El-Yaniv and Y. Wiener. 2010. On the foundations of noise-free selective classification. *Journal of Machine Learning Research (JMLR)*, 11.

R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner. 2017. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*.

J. Feng, A. Sondhi, J. Perry, and N. Simon. 2019. Selective prediction-set models with coverage guarantees. *arXiv preprint arXiv:1906.05473*.

A. Fisch, A. Talmor, R. Jia, M. Seo, E. Choi, and D. Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Workshop on Machine Reading for Question Answering (MRQA)*.

Y. Gal and Z. Ghahramani. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*.

Y. Geifman and R. El-Yaniv. 2017. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

A. Geiger, I. Cases, L. Karttunen, and C. Potts. 2019. Posing fair generalization tasks for natural language inference. In *Empirical Methods in Natural Language Processing (EMNLP)*.

D. C. Gondek, A. Lally, A. Kalyanpur, J. W. Murdock, P. A. Duboue, L. Zhang, Y. Pan, Z. M. Qiu, and C. Welty. 2012. A framework for merging and ranking of answers in DeepQA. *IBM Journal of Research and Development*, 56.

D. Hendrycks and K. Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations (ICLR)*.

D. Hendrycks, K. Lee, and M. Mazeika. 2019a. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning (ICML)*.

D. Hendrycks, M. Mazeika, and T. Dietterich. 2019b. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations (ICLR)*.

R. Jia and P. Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Empirical Methods in Natural Language Processing (EMNLP)*.

M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Association for Computational Linguistics (ACL)*.

J. Ko, L. Si, and E. Nyberg. 2007. A probabilistic framework for answer selection in question answering. In *North American Association for Computational Linguistics (NAACL)*.

T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, M. Kelcey, J. Devlin, K. Lee, K. N. Toutanova, L. Jones, M. Chang, A. Dai, J. Uszkoreit, Q. Le, and S. Petrov. 2019. Natural questions: a benchmark for question answering research. In *Association for Computational Linguistics (ACL)*.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*.

W. Lehnert. 1977. *The Process of Question Answering*. Ph.D. thesis, Yale University.

S. Liang, Y. Li, and R. Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations (ICLR)*.

B. Magnini, M. Negri, R. Prevete, and H. Tanev. 2002. Is it the right answer? exploiting web redundancy for answer validation. In *Association for Computational Linguistics (ACL)*.

S. Min, E. Wallace, S. Singh, M. Gardner, H. Hajishirzi, and L. Zettlemoyer. 2019. Compositional questions do not necessitate multi-hop reasoning. In *Association for Computational Linguistics (ACL)*.

A. Naik, A. Ravichander, N. Sadeh, C. Rose, and G. Neubig. 2018. Stress test evaluation for natural language inference. In *International Conference on Computational Linguistics (COLING)*, pages 2340–2353.

Y. Oren, S. Sagawa, T. Hashimoto, and P. Liang. 2019. Distributionally robust language modeling. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems (NeurIPS)*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research (JMLR)*, 12.

A. Peñas, P. Forner, R. Sutcliffe, Álvaro Rodrigo, C. Forăscu, I. Alegria, D. Giampiccolo, N. Moreau, and P. Osenova. 2009. Overview of ResPubliQA 2009: Question answering evaluation over european legislation. In *Cross Language Evaluation Forum*.

A. Peñas, E. Hovy, P. Forner, Álvaro Rodrigo, R. Sutcliffe, and R. Morante. 2013. QA4MRE 2011-2013: Overview of question answering for machine reading evaluation. In *Cross Language Evaluation Forum*.

J. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74.

P. Rajpurkar, R. Jia, and P. Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Association for Computational Linguistics (ACL)*.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing (EMNLP)*.

P. Rodriguez, S. Feng, M. Iyyer, H. He, and J. Boyd-Graber. 2019. Quizbowl: The case for incremental question answering. *arXiv preprint arXiv:1904.04792*.

B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. 1999. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems (NeurIPS)*.

L. Smith and Y. Gal. 2018. Understanding measures of uncertainty for adversarial example detection. In *Uncertainty in Artificial Intelligence (UAI)*.

A. Talmor and J. Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Association for Computational Linguistics (ACL)*.

J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *North American Association for Computational Linguistics (NAACL)*.

M. Toplak, R. Močnik, M. Polajnar, Z. Bosnić, L. Carlsson, C. Hasselgren, J. Demšar, S. Boyer, B. Zupan, and J. Stålring. 2014. Assessment of machine learning reliability methods for quantifying the applicability domain of QSAR regression models. *Journal of Chemical Information and Modeling*, 54.

A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, and K. Suleman. 2017. NewsQA: A machine comprehension dataset. In *Workshop on Representation Learning for NLP*.

E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Empirical Methods in Natural Language Processing (EMNLP)*.

M. Wang, N. A. Smith, and T. Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for QA. In *Empirical Methods in Natural Language Processing (EMNLP)*.

T. Winograd. 1972. *Understanding Natural Language*. Academic Press.

Y. Yang, W. Yih, and C. Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 2013–2018.

Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*.

D. Yogatama, C. de M. d'Autume, J. Connor, T. Kocisky, M. Chrzanowski, L. Kong, A. Lazaridou, W. Ling, L. Yu, C. Dyer, et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.

B. Zadrozny and C. Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 694–699.

A Appendix

A.1 Dataset Sources

The OOD data used in calibrator training and validation was sampled from MRQA training data, and the SQuAD data for the same was sampled from MRQA validation data, to prevent train/test mismatch for the QA model (Fisch et al., 2019). The test data was sampled from a disjoint subset of the MRQA validation data.

A.2 Calibrator Features and Model

We ran experiments including question length and word overlap between the passage and question as calibrator features. However, these features did not improve the validation performance of the calibrator. We hypothesize that they may provide misleading information about a given example, e.g., a long question in SQuAD may provide more opportunities for alignment with the paragraph, making it more likely to be answered correctly, but a long question in HotpotQA may contain a conjunction, which is difficult for the SQuAD-trained model to extrapolate to.

For the calibrator model, we experimented using an MLP and logistic regression. Both were slightly worse than Random Forest.

A.3 Outlier Detection for Selective Prediction

In this section, we study whether outlier detection can be used to perform selective prediction. We train an outlier detector to detect whether or not a given input came from the in-domain dataset (i.e., SQuAD) or is out-of-domain, and use its probability of an example being in-domain for selective prediction. The outlier detection model, training data (a mixture of p_{source} and q_{known}), and features are the same as those of the calibrator. We find that this method does poorly, achieving an AUC of 24.23, Coverage at 80% Accuracy of 37.91%, and Coverage at 90% Accuracy of 14.26%. This shows that, as discussed in Section 2.3 and Section 5.2, this approach is unable to correctly identify the OOD examples that the QA model would get correct.

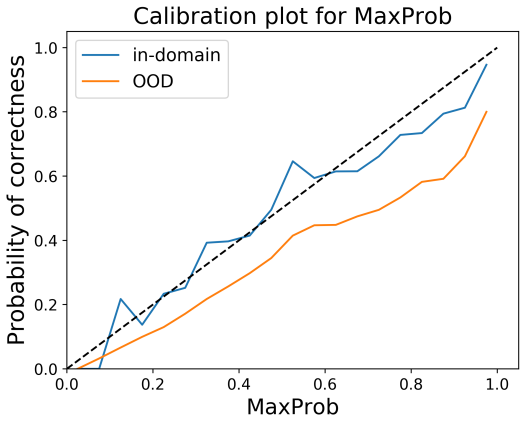

Figure 6: When considering only one answer option as correct, MaxProb is well-calibrated in-domain, but is still overconfident out-of-domain.

A.4 Underconfidence of MaxProb on SQuAD

As noted in Section 5.3, MaxProb is underconfident on SQuAD examples due to the additional correct answer options given at test time but not at train time. When the test time evaluation is restricted to allow only one correct answer, we find that MaxProb is well-calibrated on SQuAD examples (Figure 6). The calibration of the calibrator improves as well (Figure 7). However, we do not retain this restriction for the experiments, as it diverges from standard practice on SQuAD, and EM over multiple spans is a better evaluation metric since there are often multiple answer spans that are equally correct.

A.5 Accuracy and Coverage per Domain

Table 1 in Section 5.2 shows the coverage of MaxProb and the calibrator over the mixed dataset D_{test} while maintaining 80% accuracy and 90% accuracy. In Table 5, we report the fraction of these answered questions that are in-domain or OOD. We also show the accuracy of the QA model on each portion.

Our analysis in Section 5.3 indicated that MaxProb was overconfident on OOD examples, which we expect would make it answer too many OOD questions and too few in-domain questions. Indeed,

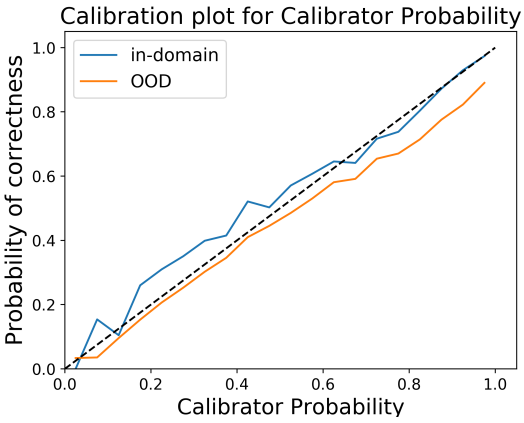

Figure 7: When considering only one answer option as correct, the calibrator is almost perfectly calibrated on both in-domain and out-of-domain examples.

at 80% accuracy, 62% of the examples MaxProb answers are in-domain, compared to 68% for the calibrator. This demonstrates that the calibrator improves over MaxProb by answering more in-domain questions, which it can do because it is less overconfident on the OOD questions.

	MaxProb Accuracy	**MaxProb Coverage**	**Calibrator Accuracy**	**Calibrator Coverage**
At 80% Accuracy				
in-domain	92.45	61.59	89.09	**67.57**
OOD	58.00	38.41	59.55	**32.43**
At 90% Accuracy				
in-domain	97.42	67.85	94.35	**78.72**
OOD	71.20	32.15	72.30	**21.28**

Table 5: Per-domain accuracy and coverage values of MaxProb and the calibrator (p_{source} and q_{known}) at 80% and 90% Accuracy on D_{test}.