# Knowing More About Questions Can Help: Improving Calibration in Question Answering

**Shujian Zhang** **Chengyue Gong** **Eunsol Choi**
The University of Texas at Austin
szhang19@utexas.edu, {cygong,eunsol}@cs.utexas.edu

## Abstract

We study calibration in question answering, estimating whether model correctly predicts answer for each question. Unlike prior work which mainly rely on the model's confidence score, our calibrator incorporates information about the input example (e.g., question and the evidence context). Together with data augmentation via back translation, our simple approach achieves 5-10% gains in calibration accuracy on reading comprehension benchmarks. Furthermore, we present the first calibration study in the open retrieval setting, comparing the calibration accuracy of retrieval-based span prediction models and answer generation models. Here again, our approach shows consistent gains over calibrators relying on the model confidence. Our simple and efficient calibrator can be easily adapted to many tasks and model architectures, showing robust gains in all settings.[1]

## 1 Introduction

Despite rapid progress in AI models, building a question answering (QA) system that can always correctly answer any given query is beyond our reach. Thus, questioners have to interpret the model prediction, deciding whether to trust it. We study providing an accurate estimate of the correctness of model prediction for each example at test time. As making incorrect predictions can be much more costly than making no prediction (e.g., missing diagnosis is much more costly than querying human experts), calibrators can bring practical benefits (Kamath et al., 2020).

Existing work on calibration focuses on mdoel confidence, such as the max probability of the predicted class (Guo et al., 2017; Desai and Durrett, 2020). Unlike classification tasks, question answering explores large output space, either through answer generation (Raffel et al., 2020; Lewis et al., 2020) or selecting a span from provided documents (Rajpurkar et al., 2016). In both settings, optimal decoding is often prohibitively expensive, and heuristic decoding is a standard practice (Seo et al., 2017). Thus, relying on the model's confidence score alone is not sufficient for calibration (Kumar and Sarawagi, 2019).

Nonetheless, prior work (Kamath et al., 2020; Jagannatha and Yu, 2020) relied heavily on model confidence, such as the max probability of the predicted answer, together with a handful of manually crafted features containing little information about the input, such as the length of the question. We empower the calibrator by introducing an input example embedding from a pre-trained language model (Alberti et al., 2019; Liu et al., 2019) fine-tuned on QA supervision data as additional features. With this simple and general feature, calibrator can identify questions regarding rare entities or examples with little lexical overlap between the question and the context. We bring further gains by paraphrasing questions or contexts respectively through back translation (Sennrich et al., 2016), providing lexical variations of the question and the context and enriching the feature space.

We evaluate our calibrator with internal metrics (i.e., calibration accuracy) and external metrics (i.e., impact on QA performance). We first evaluate calibrators in reading comprehension settings introduced in Kamath et al. (2020) – in-domain (Rajpurkar et al., 2016; Kwiatkowski et al., 2019), out of domain (Fisch et al., 2019), and adversarial (Jia and Liang, 2017). Then, we expand calibration study to more challenging open retrieval QA setting (Voorhees and Tice, 2000; Chen et al., 2017), where a system is not provided with an evidence document. We adapt our calibrator for state-of-the-art generation based (Raffel et al., 2020) and extractive (retrieve-and-predict) QA models (Karpukhin

[1]Code is available at https://github.com/szhang42/Calibration_qa.



*Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1958–1970

et al., 2020), showing gains in both models. While calibration accuracy is higher in the generation based model, the extractive method provides better answer coverage above fixed accuracy. Lastly, we use calibrator as a reranker for the answer span candidates in an extractive open retrieval QA model (Karpukhin et al., 2020), showing modest gains. We provide rich ablation studies on design choices for our calibrator, such as the choice of base model to derive input example encoding. Our simple input example embedding from pretrained language models shows consistent gains in all settings and datasets. Without any manual engineering specific to the question answering task, our calibrator could be easily adapted to other tasks with rich output space.

## 2 Problem Definition

We estimate how the models' prediction confidence aligns with the empirical likelihood of correctness (Brier, 1950). Formally, a calibrator $f$ takes the input example $x_i$ and the trained model $M_\theta$ and identifies whether the model's prediction is correct or not. We treat the correctness as binary (i.e., answer string exact match) for simplicity, instead of partial credit (e.g., token level F1 score). We study two settings: reading comprehension (RC) and open retrieval QA. In RC, an input example $x_i$ will be a context $c_i$ and the question $q_i$, and in open domain QA, an input example will be a corpus $C$ and the question $q_i$.

We use the same metrics to evaluate the performance of the calibrator $f$ in the two settings.

### 2.1 Metric: Calibrator performance

**Accuracy:** Given evaluation data $D_{\text{eval}} = \{(x_1, y_1), (x_2, y_2) \dots (x_N, y_N)\}$ and a learned model $M_\theta$, we define the accuracy of the calibrator $f$ as:

$$\text{acc}(f) = \sum_{i=1}^{N} \mathbb{I}\Big\{ f(x_i, M_\theta) = \mathbb{I}[M_\theta(x_i) = y_i] \Big\}.$$

**AUROC:** Based on the above definition of the accuracy of the calibrator $f$, we computes the coverage – fraction of evaluation data $D_{eval}$ that model makes prediction on – and risk, the error at that coverage. We plot risk versus coverage graph, and measure the area under the curve, i.e., AUROC (Area Under the Receiver Operating Characteristic Curve) (Hanley and McNeil, 1982).

### 2.2 Metric: End task performance

We measure how the calibrator performance impacts QA performances. First, we study selective QA setting – where we use calibrator score to decide which examples from $D_{eval}$ to make predictions.

For the extractive model for open retrieval QA (Karpukhin et al., 2020), where multiple answer candidates are given, we further evaluate the performance of calibrator as a reranker and measure the answer span exact match (EM) score.

**Selective QA (coverage at fixed accuracy):** We use the calibrator score to rank the *examples* in the evaluation data. Specifically, we use the calibrator's confidence for the top answer candidate instead of model score to decide which examples in $D_{eval}$ the model answers most confidently. Then, we report the percentage of evaluation data that can be predicted while maintaining threshold accuracy (80%), following prior work (Kamath et al., 2020).

**Open Retrieval QA (top-N accuracy):** We use the calibrator score to rank the *answer candidates* for each evaluation example, similar to how candidate translations are reranked in machine translation (Shen et al., 2004). We first retrieve answer candidates from multiple paragraphs and utilize the calibrator to override the model's prediction. The calibrator scores the top N answer candidates and outputs the answer with the highest confidence score instead of the answer with the highest model score. Our calibrator can be added as last step for any open retrieval QA systems which generates multiple answer candidates without retraining the model. We evaluate the top 1 exact match accuracy and the top 5 exact match accuracy after re ranking with our calibrator score.

## 3 Methods

We propose two general approaches to improve binary calibrator: new feature vector, a dense representation of the input example (Section 3.2) and data augmentation with backtranslation which further improves the new feature vector (Section 3.3). While both are simple, well-established formula for improving end tasks in NLP, neither has been explored in the context of calibration, as prior work assumed model confidence score is the most prominent signal. We follow prior work (Kamath et al., 2020) for calibrator architecture and focus on improving its feature space.

### 3.1 Calibrator Architecture

A binary classifier is trained using the gradient boosting library XGBoost (Chen and Guestrin, 2016), which classifies each test example as correctly answered by the base QA model or not. This calibrator does not share its weights with the base QA models. We finetune the following hyperparameters on the development set: colsample by level, colsample by node, colsample by tree, learning rate, and the number of estimators. All calibrators are trained five times, each with different data partitions and random seeds. We report the variances in the results.

### 3.2 Input Example Embedding Feature From Base QA Model

Prior work uses manually designed features based on the scores to the predicted answer (details in Section 4.2.1). Such features retain little information about input example – e.g, question and the evidence context. Inspired by the recent works in machine learning (e.g. Song et al., 2019; Hendrycks et al., 2019), which use hidden vectors to classify in-domain and out-of-domain data, we introduce an input example embedding, a new feature vector that represent question and (optionally) evidence context to a calibrator.

Our input example embedding is a fixed dimensional vector representing an input example, similar to sentence embeddings (Conneau et al., 2017). It differs in that the representation is taken from the final layer of base QA model, which is trained with supervision from question answering data and it encodes question and (optional) evidence context simultaneously. In Section 6.1, we report minor performance degradation from using embeddings from generic pretrained language model instead.

Each base model processes input example, either query $q_i$ or query, context pair $(q_i, c_i)$ to generate a sequence of hidden vectors, which will be compressed into a fixed dimensional vector to be used as calibrator feature.[2] We denote the input example as a sequence of tokens $\mathbf{t} = (t_0, t_1, \cdots, t_n)$ where $n$ is the length of the input. We pass the sequence $\mathbf{t}$ through base QA model and get $(\mathbf{h}_0, \mathbf{h}_1, \cdots, \mathbf{h}_n)$ where $\mathbf{h}_i$ is the corresponding final-layer hidden state of $t_i$, and $\mathbf{h}_i = (\mathbf{h}_{i,0}, \cdots, \mathbf{h}_{i,m})$ where $m$ is the number of hidden dimensions. Then, we get the $m$-dimensional feature vector

$$\phi(q_i, c_i) = [\frac{1}{n}\sum_{i=1}^{n}\mathbf{h}_{i,0}; \cdots; \frac{1}{n}\sum_{i=1}^{n}\mathbf{h}_{i,m})], \quad (1)$$

where each dimension is an average across the length $n$. We then train a binary classifier using these features as a calibrator. We now describe our base QA models to get this hidden representations.

#### 3.2.1 Base QA Model

We use standard span prediction architecture for RC, and a generation based model and an extractive model for open retrieval QA.

For RC and extractive open retrieval QA model, we use a standard span prediction architecture based on a pretrained language model (Devlin et al., 2018), which predicts start and end index of the answer span separately with softmax layer. The output hidden vector sequence will equal the sum of the length of question and the length of evidence context. For the open retrieval QA setting, the extractive model first retrieves a passage from the corpus and predicts an answer span from it. We use the best model from dense passage retrieval (DPR)(Karpukhin et al., 2020).[3] Specifically, this model retrieves the top 100 retrieved passages as input and trains a span prediction model, which optimizes a softmax cross-entropy loss to select the correct passage among the candidates, and the answer span prediction loss. The model then selects the answer span with the highest answer span score (sum of the start and end logit score) from the passage with the highest passage score. In this setting, $\mathbf{t}$ is a concatenation of question $q_i$ and the context $c_i$.

For generation based model, we use a sequence-to-sequence (seq2seq) model, specifically T5-small (Raffel et al., 2020), which takes the question as an input and generates answer tokens. For this base QA model, $\mathbf{t}$ only consists of the query since the context is not provided.

**Data** For all experiments in RC, we train the model on the SQuAD 1.1 dataset. For open retrieval QA, models are trained on the Natural Questions (NQ) dataset (Kwiatkowski et al., 2019) following the data split from Lee et al. (2019).

### 3.3 Data Augmentation Via Paraphrasing

Paraphrase generation can improve QA models (Yu et al., 2018) by handling language vari-

[2] For simplicity, we write equations with $(q_i, c_i)$ pair as an input, when only query is provided (e.g., generation based open retrieval QA method) $c_i$ is empty.

[3] https://github.com/facebookresearch/DPR

| Task | Setup | Base QA model (train) | Calibrator (train & dev) | Calibrator (test) |
|---|---|---|---|---|
| Standard RC | In domain | SQuAD 1.1 | SQuAD 1.1 + HotpotQA | SQuAD 1.1 + HotpotQA |
| | Out of domain | | SQuAD 1.1 + HotpotQA | Other MRQA |
| | In domain | | SQuAD 1.1 + NQ | SQuAD 1.1 + NQ |
| | Out of domain | | SQuAD 1.1 + NQ | Other MRQA |
| Adversarial RC | In domain | | SQuAD 1.1 Adversarial | SQuAD1.1 Adversarial |
| | Out of domain | | MRQA | SQuAD1.1 Adversarial |
| Unanswerable RC | In domain | | SQuAD 2.0 | SQuAD 2.0 |
| | Out of domain | | MRQA | SQuAD 2.0 |
| Open Retrieval QA | In domain | NQ Training Set | | NQ Test Set |

Table 1: Experiment Configuration. In domain / Out of domain distinguishes whether the training data for calibrator is different from the test data.

ation. Compared to sentence retrieval (Du et al., 2020) and language model based example generation (Anaby-Tavor et al., 2020), backtranslation can capture the ambiguity of questions and answer(Singh et al., 2019). Given a $(q_i, c_i)$ pair, we use back translation (Sennrich et al., 2016) to generate paraphrases of the question $q_i'$ from $q_i$ and the evidence context $c_i'$ from $c_i$.

We use standard transformer-based neural machine translation models (Junczys-Dowmunt et al., 2018) trained on WMT dataset.[4] We first translate the original sentences to a pivot language and then translate them back to the source language. To guarantee translation quality, French and German are used as the pivot languages. We use beam search decoding with beam size as 4 and truncate the context length to 512, as the reading comprehension model truncates the context anyway. We analyze the quality of backtranslation in Section 6.2.

We denote $(q_i', c_i)$ as $\mathbf{t}^q = (t_0^q, \cdots, t_{n_q}^q)$ and $(q_i, c_i')$ as $\mathbf{t}^c = (t_0^c, \cdots, t_{n_c}^c)$. Here, $n_q$ and $n_c$ denote the length after backtranslating the question and context, respectively. For $\mathbf{t}^q$ and $\mathbf{t}^c$, we pass them through the base QA model, get $\mathbf{h}^q$ and $\mathbf{h}^c$, and extract the $m$-dimensional feature vector as in Eqn (1),

$$\begin{aligned}\phi(q_i', c_i) &= [\frac{1}{n_q}\sum_{i=1}^{n_q}\mathbf{h}_{i,0}^q, \cdots, \frac{1}{n_q}\sum_{i=1}^{n_q}\mathbf{h}_{i,m}^q], \\ \phi(q_i, c_i') &= [\frac{1}{n_c}\sum_{i=1}^{n_c}\mathbf{h}_{i,0}^c, \cdots, \frac{1}{n_c}\sum_{i=1}^{n_c}\mathbf{h}_{i,m}^c)].\end{aligned} \tag{2}$$

We use the concatenation of the original input example embedding and backtranslated one, $[\phi(q_i, c_i); \phi(q_i', c_i)]$ and $[\phi(q_i, c_i); \phi(q_i, c_i')]$ as features. Backtranslating both context and question did not bring further gains, thus the results from such a feature set are not presented. We hypothesize that backtranslating context and question together might introduce too severe noise. We do not use data augmentation for open retrieval QA experiments.

[4] https://huggingface.co/transformers/model_doc/marian.html

## 4 Experimental Settings

In this section, we describe the experimental setting, dataset setups and baseline systems. Table 1 summarizes the evaluation scheme. A separate calibrator is trained for each calibrator train data configuration.

### 4.1 Data

For all in-domain reading comprehension experiments, we randomly split the data into training, development, and test (40%,10%,50%), following regression and classification benchmarks (Asuncion and Newman, 2007). Further, we assume only limited supervised data is available for calibrators, simulating a set up where we have a general QA model and small number of annotated data reserved for calibration.

**Standard RC** We test two in domain settings and two out of domain settings. We randomly sample 4K examples from each of the datasets included in the training portion of the MRQA shared task (Fisch et al., 2019) (SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2017), TriviaQA (Joshi et al., 2017), SearchQA (Dunn et al., 2017), HotpotQA (Yang et al., 2018), Natural Questions (Kwiatkowski et al., 2019)). We train two calibrators, one with the SQuAD1.1 + HotpotQA datasets and another with the SQuAD1.1 + NQ datasets. For out of domain evaluation, we use four remaining datasets from MRQA shared task training set.

**Adversarial RC (SQuAD 1.1 Adversarial)** The adversarial examples manipulate the evidence paragraph to change the model prediction but not the gold answer. We sample 2K examples from the development portion of the SQuAD 1.1 (Jia and Liang, 2017) AddSent dataset, which appends an additional sentence that looks similar to the question at the end of the paragraph. For the out-of-domain case, we train the calibrator on 6K examples (1K each sampled from MRQA datasets) and test on adversarial examples.

**Unanswerable RC (SQuAD 2.0)** We sampled 2K examples from the development portion of the SQuAD 2.0 dataset (Rajpurkar et al., 2018), which contains examples where the answer to the question cannot be derived from the provided context. Crowdworkers posed questions that were impossible to answer based on the paragraph alone while referencing entities in the paragraph and ensuring that a plausible answer is present. For out of domain setting, we train the calibrator on 6K examples (1K each sampled from MRQA datasets) and test on SQuAD 2.0 dataset (same as adversarial RC setting).

**Open Retrieval QA** We use the open retrieval version of the NQ (Lee et al., 2019). We split its training data 60% and 40% for calibrator training and validation and use the NQ test set for testing.

### 4.2 Comparison Systems

We summarize the calibrators used in our study in Table 2. All calibrators are trained with the same gradient boosting library XGBoost (Chen and Guestrin, 2016), and they only differ in the feature sets. These calibrators are efficient, trained within a few minutes even with our new feature space.

#### 4.2.1 Reading Comprehension

**MaxProb** is the simplest baseline that relies on the model's confidence score. The model score is the sum of the logit scores of the start and end of the answer span for reading comprehension. For open retrieval question answering, the model first determines the passage with the highest passage-match score and then extracts the answer span from this passage.

Formally, given the set of answer spans $Y$, MaxProb with model $M_\theta$ estimates confidence on input $x_i$ as:

$$\textit{MaxProb} = \max_{y \in Y} M_\theta \left( y \mid x_i \right),$$

| QA model | Calibrator Feature Set | # Features |
|---|---|---|
| RC | MaxProb | 1 |
| | features: Kamath et al. (2020) | 17 |
| | Ours | $m$ |
| | + features | 17 + $m$ |
| | + features + $\phi(q_i, c_i')$ | 17 + $2m$ |
| | + features + $\phi(q_i', c_i)$ | 17 + $2m$ |
| Extractive (DPR) | Unnormalized Scores | 2 |
| | Normalized Scores | 2 |
| | Ours + Normalized Score | 2 + $2m$ |
| Generation (T5) | Likelihood | 1 |
| | Ours + Likelihood | 1 + $m$ |

Table 2: Comparison Systems: different calibrators explored for three base QA models. The last two QA models are for open retrieval QA task. The dimension of question context embedding is $m$ defined in Eqn (1) (eg. $m$ is 768 for reading comprehension).

where $M_\theta(y \mid x_i)$ refers to the model score for candidate answer $y$.

**Kamath et al. (2020)** uses a calibrator based on the following general features: passage length, the predicted answer length, and the top-5 largest softmax probabilities generated by the model. They also use test time dropout (Gal and Ghahramani, 2016): given an input $x_i$ and model $M_\theta$, compute $M_\theta(x_i)$ with $K$ different dropout masks, obtaining prediction distributions $\hat{p}_1, ..., \hat{p}_k$, where each $\hat{p}_i$ is a probability distribution over $Y$. Two options are used as confidence estimates. First, taking the mean of $\hat{p}_i$ (Lakshminarayanan et al., 2017)

$$\textit{Dropout Mean} = \frac{1}{K} \sum_{i=1}^{K} \hat{p}_i.$$

Second, taking the variance of the $\hat{p}_i$ (Feinman et al., 2017; Smith and Gal, 2018)

$$\textit{Dropout Variance} = \text{Var}\left[\hat{p}_1, \ldots, \hat{p}_K\right].$$

The dimension of *MaxProb*, 2th-5th probability, *Dropout Mean*, *Dropout Variance*, context length and prediction length are 1, 4, 5, 5, 1, 1, respectively. In total, this feature set contains 17 features.

**Ours** represents a calibrator that is trained with the question context embedding, $\phi(q_i, c_i)$ in Eqn (1). '+ features' refers to augmenting features from (Kamath et al., 2020), described above. Augmenting the feature sets with question context embeddings from backtranslated questions is denoted as '+$\phi(q_i', c_i)$', and augmenting the feature sets with question context embeddings from backtranslated contexts is denoted as '+$\phi(q_i, c_i')$' from Eqn. (2).

| | In Domain | | | Out of Domain | | |
|---|---|---|---|---|---|---|
| | Calib. Accu | AUROC | Cov@Acc=80% | Calib. Accu | AUROC | Cov@Acc=80% |
| | SQuAD1.1 + HotpotQA | | | SQuAD1.1 + HotpotQA / Other MRQA datasets | | |
| MaxProb | 58.2±0.2 | 58.0±0.3 | 38.4% | 56.8±0.2 | 56.5±0.2 | 38.3% |
| Kamath et al. (2020) | 62.6±0.5 | 62.3±0.7 | 40.9% | 61.2±0.4 | 60.7±0.5 | 39.7% |
| Ours | 65.8±0.3 | 66.8±0.4 | 43.1% | 63.7±0.3 | 64.1±0.3 | 41.6% |
| + features | 67.4±0.5 | 68.5±0.4 | 43.3% | 65.4±0.3 | 66.9±0.3 | 42.7% |
| + features + $\phi(q_i, c_i')$ | **69.2±0.4** | **70.3±0.4** | **44.3%** | **67.6±0.4** | **68.8±0.5** | **43.9%** |
| + features + $\phi(q_i', c_i)$ | 66.8±0.3 | 67.9±0.3 | 42.4% | 64.7±0.4 | 66.2±0.3 | 42.5% |
| | SQuAD1.1 + NQ | | | SQuAD1.1 + NQ / Other MRQA datasets | | |
| MaxProb | 64.8±0.3 | 71.5±0.3 | 49.2% | 61.4±0.2 | 66.7±0.3 | 45.9% |
| Kamath et al. (2020) | 68.5±0.4 | 75.5±0.6 | 53.4% | 64.1±0.6 | 69.2±0.5 | 51.5% |
| Ours | 69.5±0.3 | 76.3±0.5 | 57.8% | 64.3±0.4 | 69.4±0.4 | 54.3% |
| + features | 70.3±0.4 | 77.0±0.3 | 59.1% | 64.9±0.5 | 70.4±0.5 | 56.5% |
| + features + $\phi(q_i, c_i')$ | **73.2±0.4** | **79.4±0.3** | **60.7%** | **66.7±0.5** | **72.1±0.5** | **57.6%** |
| + features + $\phi(q_i', c_i)$ | 72.5±0.4 | 78.7±0.3 | 59.3% | 65.8±0.5 | 71.4±0.5 | 55.9% |

Table 3: Calibration results on standard reading comprehension datasets. In the out of domain setting, we first list the training dataset of calibrator, then the test dataset.

#### 4.2.2 Open Retrieval QA

We consider separate calibrators for two different approaches (Karpukhin et al., 2020; Raffel et al., 2020).

**Extractive (Retrieve-And-Predict)** We consider two baseline calibrators: one takes the product of normalized passage score (normalized across all passage candidates) and answer score (normalized across the top 10 answer spans for each passage), and another takes the product of unnormalized passage and answer scores.

Then, we introduce calibrator augmented with our input example embedding. We include two example embeddings as features: one is the question context embedding as used in the reading comprehension setting (from Eqn 1), and another is the average of the answer span start token representation and the answer span end token representation.

**Generation based (Seq2Seq)** For seq2seq models (Raffel et al., 2020), the output answer space includes all sentences that can be generated with conditional language model. Thus, instead of MaxProb, we use the likelihood of the generated answer (i.e., the product of the conditional probabilities for each token in the generated answer) as a baseline. Then, we introduce calibrator with our input example embedding (from Eqn 1).

## 5 Results

**Calibration** Table 3 reports calibration results on standard reading comprehension datasets. The top block displays the performance of calibrators trained on the SQuAD and HotpotQA datasets, and the bottom block shows the results of calibrators trained on the SQuAD and NQ datasets. In both settings, the our input example embedding works better than the manual feature set. However, two approaches are complementary in all settings. Interestingly, paraphrasing questions shows gains in Natural Questions but not in other datasets. We hypothesize that organically collected search queries contain more ambiguous and ill-defined queries than crowdsourced questions where questions were based directly on the context. Adding paraphrased context embeddings, on the other hand, shows a modest gain across all settings. Unlike QA models have access to millions of parameters, calibrators, even with our feature set, are provided with very limited information. We hypothesize that augmenting the feature set with paraphrased context enabled the calibrator to gain more information about the example, facilitating higher performance.

Table 4 shows the results in more challenging settings: one with adversarial attacks and another containing unanswerable questions. In both settings, we observe sizable gains (5-10% increase in calibration accuracy) for the in domain setting, but the gains are smaller in out of domain settings. Similar to the Natural Questions dataset, in SQuAD 2.0, which includes adversarially designed questions without an answer, paraphrasing the question is more helpful than paraphrasing the context. On the other hand, in the adversarial setting where contexts are manipulated, paraphrasing contexts is more effective. Overall, our new feature vector shows consistent gain across all datasets and settings.

We present the calibration in open retrieval QA in Table 5. Overall, calibrator accuracy is higher

| | In Domain | | | Out of Domain | | |
|---|---|---|---|---|---|---|
| | Calib. Accu | AUROC | Cov@Acc=80% | Calib. Accu | AUROC | Cov@Acc=80% |
| | SQuAD1.1 Adversarial | | | MRQA / SQuAD1.1 Adversarial | | |
| Kamath et al. (2020) | 52.4±0.2 | 53.7±0.4 | 25.4% | 52.4±0.2 | 52.2±0.4 | 24.7% |
| Ours | 61.1±0.4 | 63.2±0.3 | 35.6% | 53.2±0.4 | 53.6±0.3 | 25.3% |
| + features | 61.4±0.6 | 63.5±0.3 | 35.8% | 53.8±0.4 | 54.3±0.4 | 26.8% |
| + features + $\phi(q_i, c_i')$ | **62.8±0.3** | **65.2±0.2** | **37.3%** | **54.9±0.5** | **55.1±0.3** | **27.5%** |
| + features + $\phi(q_i', c_i)$ | 61.6±0.3 | 63.7±0.4 | 35.5% | 53.6±0.5 | 53.9±0.5 | 26.6% |
| | SQuAD2.0 | | | MRQA / SQuAD2.0 | | |
| Kamath et al. (2020) | 57.6±0.3 | 59.2±0.4 | 31.7% | 54.8±0.4 | 56.5±0.5 | 29.6% |
| Ours | 58.9±0.2 | 61.1±0.2 | 33.8% | 55.7±0.3 | 57.4±0.4 | 30.7% |
| + features | 60.1±0.2 | 61.9±0.3 | 34.2% | 56.6±0.4 | 58.3±0.5 | 31.6% |
| + features + $\phi(q_i, c_i')$ | 60.2±0.3 | 61.8±0.3 | 34.1% | 56.4±0.5 | 57.9±0.4 | 31.2% |
| + features + $\phi(q_i', c_i)$ | **62.6±0.4** | **64.3±0.3** | **35.9%** | **58.1±0.4** | **60.4±0.4** | **32.9%** |

Table 4: Calibration results on adversarial and unanswerable SQuAD datasets. In the out of domain setting, we first list the training dataset of calibrator, then the test dataset.

| Model | Answer Acc | Calibrator | Calib. Accu | Calib. AUROC | Cov@Acc=80% |
|---|---|---|---|---|---|
| Extractive (DPR) (Karpukhin et al., 2020) | 41.0 | Unnormalized scores | 65.9±0.2 | 65.2±0.2 | 10.4% |
| | | Normalized scores | 72.2±0.4 | 74.5±0.3 | 28.9% |
| | | Ours (+ Normalized Scores) | **77.3±0.3** | **78.7±0.2** | **30.5%** |
| Generation (T5) (Raffel et al., 2020) | 25.5 | Likelihood | 89.3±0.1 | 86.6±0.1 | 10.4% |
| | | Ours (+ Likelihood) | **91.6±0.3** | **92.9±0.1** | **11.3%** |

Table 5: Calibration results on NQ open retrieval test set for different base QA models and calibration features.

compared to RC, partially because the answer accuracy is substantially lower. For example, with generation based model (T5)'s answer accuracy of 25.5, simply predicting incorrectly for every example will give 74.5 calibration accuracy. In both models, internal confidence scores (Likelihood and Normalized scores) provide reasonable calibrator performance, yet adding our feature set improves the performance. In particular, our calibrator shows a larger gain in the DPR setting. Encouraged by this result, we test our calibrator as an answer candidate reranker for top answer candidates from DPR. Despite high calibration accuracy of generation based approach, selective QA performance (Cov@Acc=80%) is higher with the extractive approach, suggesting comparing calibration performance across models of different accuracy is challenging.

**Answer Reranking** Table 6 shows the results of our calibrator as an answer candidate reranker. The calibrator considers the top 1,000 answer candidates (100 retrieved passages, each with top 10 answer spans) and outputs top candidates based on the calibrator score instead of the model score. We show negligible gains in top 1 accuracy but bigger gains in top 5 accuracy. These small but noticeable gains show potential for using calibrators to improve open retrieval QA performances, where multiple answer candidates are considered.

| | Top 1 EM | Top 5 EM |
|---|---|---|
| DPR | 41.0 | 57.8 |
| Unnormalized scores | 10.3±0.2 | 23.1±0.3 |
| Normalized scores | 41.2±0.1 | 58.6±0.1 |
| Ours (+ Normalized scores) | **41.4±0.1** | **59.0±0.1** |

Table 6: Results on open domain question answering in NQ. The calibrator is used as a reranker for selecting the top answer span out of 1,000 answer spans (10 answer spans per each of 100 retrieved passages).

## 6 Analysis

### 6.1 Task-Agnostic Representation vs. Representation from QA Model

Our study has shown that input example embedding is very useful, adding complementary power to model confidence features. Based on this result, we further ask the question, is it possible to build a calibrator without accessing the model parameters, but only a small amount of calibration training data (which consists of questions, context, and whether the model's prediction is correct or incorrect)? We train a calibrator that does not have any access to the QA model parameters and only takes the model's predictions on a small set of training data (a couple thousand of QA examples). This cali-

| | In Domain | Out of Domain |
|---|---|---|
| | SQuAD1.1 + Hotpot QA | Other MRQA datasets |
| CLS | 63.5±0.4 | 62.3±0.5 |
| Ours | 65.8±0.3 | 63.7±0.3 |
| Diff. | 2.3% | 1.4% |

Table 7: CLS token ablation results, all numbers refer to calibration accuracy. Using CLS token as a feature shows a strong calibration performance, lagging behind question context encoding from the RC model only by a few points. The gap is even smaller in out of domain setting.

brator uses a standard pretrained language model (BERT) to encode [CLS; $(q_i, c_i)$] and takes the final layer hidden representation of the [CLS] token as a feature. Table 7 shows the performance of the [CLS] token classifier. Surprisingly, this calibrator outperforms the MaxProb baseline (in Table 3) in all settings and outperforms Kamath et al. (2020) (in Table 3) in most settings, indicating information about the question and context might be more useful than the QA model's confidence. Using the input example embedding from the QA model shows only 1-3 point gains than using the CLS token embedding. This trend holds for across various settings (more results in Table 11 in Appendix).

### 6.2 Quality of Back Translation

Question paraphrasing (Dong et al., 2017) can improve performances of QA models. Similarly, both question and context paraphrasing improves calibration performance. In this section, we investigate the quality of backtranslation used in our study. We manually inspect 100 question paraphrasing from SQuAD 2.0 dataset. 71 examples maintain the original meaning, 12 examples change its meanings, and 17 examples are hard to distinguish. One common pattern for meaning change is when proper nouns in the original sentences are missing and incorrectly translated (e.g. John Calvin → Jean Calvin).

We study how much variability is introduced during paraphrasing by studying divergence between the original sentence and the paraphrased sentence. We calculate the sentence BLEU score with *NLTK* (Bird et al., 2009), using the original text as source and the back-translated text as target for both question paraphrasing and context paraphrasing. The average sentence BLEU score is larger than 0.55 for all datasets, indicating back-translation introduces relatively minor changes in phrasing.

| | |
|---|---|
| $q$ | In what country is Normandy located? |
| $q'$ | What country is Normandy in? |
| $q$ | When did Edward return? |
| $q'$ | When did Edward come back? |
| $q$ | How would one write T(n) = 7n2 + 15n + 40 in big O notation? |
| $q'$ | How do you write T(n) = 7n2 + 15n + 40? |
| $q$ | What kind of arches does Norman architecture have? |
| $q'$ | What kind of arches does Norman's building have? |

Table 8: Question back translation samples from SQuAD 2.0 dataset. The first row (q) refers to the original question, and the second row ($q'$) refers to back-translated question. In the third example, back translation introduces an error.

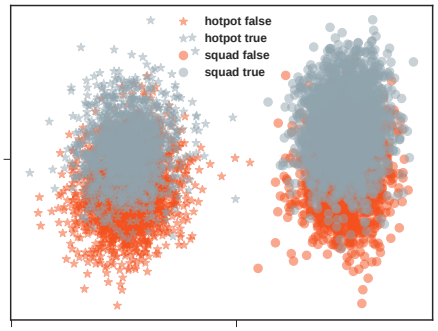


Figure 1: A visualization for the input example embedding from HotPotQA and SQuAD datasets. We denote the data domain by markers with different shapes and denote the correctness with different colors. The X-axis and Y-axis denote the first and second dimensions extracted by linear discriminant analysis, respectively.

**Visualization** Figure 1 shows a visualization of the question context embeddings from HotpotQA and SQuAD. We use linear discriminant analysis (Pedregosa et al., 2011) to plot input example embeddings and observe that embeddings from the same dataset are closer to each other. It demonstrates that embeddings are almost linearly separable between domains, but it is much harder to distinguish correct answers from incorrect ones.

**Choice of Calibrator Architecture** We test if our results are sensitive to the choice of classifiers: XGBoost, logistic regression (LR), and k-nearest

| | In Domain | Out of Domain |
|---|---|---|
| | SQuAD1.1 + NQ | Other MRQA datasts |
| Xgboost | 67.4±0.5 | 65.4±0.3 |
| LR | 66.6±0.3 | 64.7±0.3 |
| KNN | 66.3±0.2 | 64.6±0.3 |

Table 9: Ablation study on different classifiers with features (Ours + features). All numbers refer to calibration accuracy.

| | Lang-uage | In Domain SQuAD1.1 + NQ | Out of Domain Other MRQA datasets |
|---|---|---|---|
| $\phi(q_i, c_i')$ | FR | 73.2±0.4 | 66.7±0.5 |
| $\phi(q_i, c_i')$ | DE | 73.0±0.4 | 66.5±0.5 |
| $\phi(q_i', c_i)$ | FR | 72.5±0.4 | 65.8±0.5 |
| $\phi(q_i', c_i)$ | DE | 72.8±0.3 | 66.1±0.4 |

Table 10: Calibration accuracy for different pivot languages: French vs. German, using calibrator with features (Ours + features).

neighbors (KNN). Table 9 indicates our gains hold across different classifiers. Full experimental results can be found in Appendix.

**Choice of Pivot Language** We test whether the choice of pivot language in backtranslation impacts performances. We find little difference between pivoting through German or French (See Table 10).

## 7 Related Work

**Calibration in NLP** Calibration has become an important topic in NLP as well as general machine learning (Guo et al., 2018; Pleiss et al., 2017; FAN et al., 2021) as confidence scores from calibrators can be useful for the error correction process (Feng and Sears, 2004). Calibration has been studied in natural language inference, commonsense reasoning (Desai and Durrett, 2020; Varshney et al., 2020), dialogue systems (Mielke et al., 2020), semantic parsing (Dong et al., 2018), coreference resolution (Nguyen and O'Connor, 2015) and sequence labeling (Jagannatha and Yu, 2020).

In question answering, Kamath et al. (2020)'s study on selective question answering inspired our work. We measure the calibration performance with calibrator accuracy, AUROC, and coverage at accuracy. Expected Calibration Error (ECE) (Guo et al., 2017) is another commonly used metric for calibration performance, but we consider calibrator as a binary classifier at here. Jagannatha and Yu (2020) also studies calibration in reading comprehension, using language model perplexity and model's confidence as features. Language model perplexity coarsely and indirectly captures information about the question and context. We propose an improved feature space and thoroughly test it in challenging settings, e.g., adversarial RC, unanswerable RC, and open retrieval QA.

**Calibration During Training** Recent work in QA introduces an answer verification step (Tan et al., 2018; Hu et al., 2019; Wang et al., 2020) at the end of the pipeline. During the training, this verifier module takes the questions, answers, or MRC model's state as inputs and determines the answers' validity. Then, the validity score is used to update the model parameters during training. Thus, the validator is jointly trained with the MRC model. While this is conceptually similar to our set up, instead of tying the calibrator into the model, we design a universal post-hoc calibrator that can be easily applied to any model architecture.

**Calibration with Ensembles** Ensemble diversity has been used to improve uncertainty estimation and calibration (e.g. Raftery et al., 2005; Stickland and Murray, 2020). While it is effective, calibration with model ensembling is usually expensive and time consuming (Zhou et al., 2002, 2018). Our calibrator is an offline postprocessing step that does not require further training of the original model.

## 8 Conclusion

We introduce a richer feature space for question answering calibrators with question and context embeddings and paraphrase-augmented inputs. Our work suggests deciding the correctness of a QA system depends on both the semantics of the question-context and the confidence of the model. We thoroughly test our calibrator in domain shift, adversarial, and open domain QA settings. The experiments show noticeable gains in performance across all settings. We further demonstrate our calibrator's general applicability by using it as a reranker in extractive open domain QA. To summarize, our calibrator is simple, effective and general, with potential to be incorporated into existing models or extended for other NLP tasks.

## Acknowledgments

We would like to thank UT Austin NLP group, especially Kaj Bostrom and Greg Durrett for feedback and suggestions.

## References

Chris Alberti, Kenton Lee, and Michael Collins. 2019. A BERT baseline for the natural questions. *arXiv:1901.08634*.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7383–7390.

Arthur Asuncion and David Newman. 2007. Uci machine learning repository.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

G. W. Brier. 1950. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78:1–3.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Annual Meetings of the Association for Computational Linguistics (ACL)*.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. *Empirical Methods in Natural Language Processing (EMNLP)*, abs/2003.07892.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886, Copenhagen, Denmark. Association for Computational Linguistics.

Li Dong, Chris Quirk, and Mirella Lapata. 2018. Confidence modeling for neural semantic parsing. In *Annual Meetings of the Association for Computational Linguistics (ACL)*.

Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Ves Stoyanov, and Alexis Conneau. 2020. Self-training improves pre-training for natural language understanding. *arXiv preprint arXiv:2010.02194*.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

XINJIE FAN, Shujian Zhang, Korawat Tanwisuth, Xiaoning Qian, and Mingyuan Zhou. 2021. Contextual dropout: An efficient sample-dependent dropout module. In *International Conference on Learning Representations*.

Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. 2017. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*.

Jinjuan Feng and Andrew Sears. 2004. Using confidence scores to improve hands-free speech based navigation in continuous dictation systems. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 11(4):329–356.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. *arXiv preprint arXiv:1910.09753*.

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.

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

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2018. On calibration of modern neural networks. *International Conference on Machine Learning (ICML)*.

James A Hanley and Barbara J McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. 2019. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pages 2712–2721.

Minghao Hu, Furu Wei, Yuxing Peng, Zhen Huang, Nan Yang, and Dongsheng Li. 2019. Read+ verify: Machine reading comprehension with unanswerable questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6529–6537.

Abhyuday N. Jagannatha and Hong Yu. 2020. Calibrating structured output predictors for natural language processing. In *Annual Meetings of the Association for Computational Linguistics (ACL)*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Mandar Joshi, Eunsol Choi, Daniel S 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.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. *Annual Meetings of the Association for Computational Linguistics (ACL)*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *Empirical Methods in Natural Language Processing (EMNLP)*.

A. Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *ArXiv*, abs/1903.00802.

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

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems (Neurips)*, pages 6402–6413.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *International Conference on Machine Learning (ICML)*, abs/1906.00300.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, F. Petroni, V. Karpukhin, Naman Goyal, Heinrich Kuttler, M. Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *ArXiv*, abs/2005.11401.

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

Sabrina J. Mielke, Arthur Szlam, Y.-Lan Boureau, and Emily Dinan. 2020. Linguistic calibration through metacognition: aligning dialogue agent responses with expected correctness. *ArXiv*, abs/2012.14983.

Khanh Nguyen and Brendan T. O'Connor. 2015. Posterior calibration and exploratory analysis for natural language processing models. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. 2017. On fairness and calibration. In *Advances in Neural Information Processing Systems (Neurips)*, pages 5680–5689.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, M. Matena, Yanqi Zhou, W. Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

A. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski. 2005. Using bayesian model averaging to calibrate forecast ensembles. *Monthly Weather Review*, 133:1155–1174.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *Annual Meetings of the Association for Computational Linguistics (ACL)*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *Empirical Methods in Natural Language Processing (EMNLP)*.

Rico Sennrich, B. Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, volume abs/1511.06709.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations (ICLR)*, volume abs/1611.01603.

Libin Shen, Anoop Sarkar, and F. Och. 2004. Discriminative reranking for machine translation. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Jasdeep Singh, Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. Xlda: Cross-lingual data augmentation for natural language inference and question answering. *arXiv preprint arXiv:1905.11471*.

Lewis Smith and Yarin Gal. 2018. Understanding measures of uncertainty for adversarial example detection. *arXiv preprint arXiv:1803.08533*.

Jiaming Song, Yang Song, and Stefano Ermon. 2019. Unsupervised out-of-distribution detection with batch normalization. *arXiv preprint arXiv:1910.09115*.

Asa Cooper Stickland and Iain Murray. 2020. Diverse ensembles improve calibration. *ArXiv*, abs/2007.04206.

Chuanqi Tan, Furu Wei, Qingyu Zhou, Nan Yang, Weifeng Lv, and Ming Zhou. 2018. I know there is no answer: modeling answer validation for machine reading comprehension. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 85–97. Springer.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. *ACL 2017*, page 191.

N. Varshney, Swaroop Mishra, and Chitta Baral. 2020. It's better to say "i can't answer" than answering incorrectly: Towards safety critical nlp systems. *ArXiv*, abs/2008.09371.

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Xuguang Wang, Linjun Shou, Ming Gong, Nan Duan, and Daxin Jiang. 2020. No answer is better than wrong answer: A reflection model for document level machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4141–4150.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Adams Wei Yu, David Dohan, Minh-Thang Luong, R. Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. volume abs/1804.09541.

Tianyi Zhou, S. Wang, and J. Bilmes. 2018. Diverse ensemble evolution: Curriculum data-model marriage. In *NeurIPS*.

Z. Zhou, Jianxin Wu, and Wei Tang. 2002. Ensembling neural networks: Many could be better than all. *Artif. Intell.*, 137:239–263.

# Appendix

## A Additional Experimental Results

| | In Domain | | | Out of Domain | | |
|---|---|---|---|---|---|---|
| | Calib. Accu | AUROC | Cov@Acc=80% | Calib. Accu | AUROC | Cov@Acc=80% |
| | SQuAD1.1 + Hotpot | | | SQuAD1.1 + HotpotQA / Other MRQA datasets | | |
| CLS | 63.5±0.4 | 65.2±0.4 | 41.8% | 62.3±0.5 | 62.6±0.3 | 40.3% |
| Ours | 65.8±0.3 | 66.8±0.4 | 43.1% | 63.7±0.3 | 64.1±0.3 | 41.6% |
| Difference | 2.3 | 1.6 | 1.3% | 1.4 | 1.5 | 1.3% |
| | SQuAD1.1 + NQ | | | SQuAD1.1 + NQ / Other MRQA datasets | | |
| CLS | 66.8±0.3 | 74.0±0.5 | 58.5% | 62.8±0.4 | 67.8±0.4 | 57.6% |
| Ours | 69.5±0.3 | 76.3±0.5 | 62.8% | 64.3±0.4 | 69.4±0.4 | 59.3% |
| Difference | 2.7 | 2.3 | 4.3% | 1.5 | 1.6 | 1.7% |

Table 11: CLS token ablation results on reading comprehension.

| | In Domain | | | Out of Domain | | |
|---|---|---|---|---|---|---|
| | Calib. Accu | AUROC | Cov@Acc=80% | Calib. Accu | AUROC | Cov@Acc=80% |
| | SQuAD1.1 + Hotpot | | | SQuAD1.1 + HotpotQA / Other MRQA datasets | | |
| Xgboost | 67.4±0.5 | 68.5±0.4 | 43.3% | 65.4±0.3 | 66.9±0.3 | 42.7% |
| Logistic Regression | 66.6±0.3 | 67.3±0.3 | 42.6% | 64.7±0.3 | 66.1±0.3 | 42.3% |
| KNN | 66.3±0.2 | 67.0±0.3 | 42.1% | 64.6±0.3 | 65.8±0.2 | 41.8% |
| | SQuAD1.1 + NQ | | | SQuAD1.1 + NQ / Other MRQA datasets | | |
| Xgboost | 70.3±0.4 | 77.0±0.3 | 59.1% | 64.9±0.5 | 70.4±0.5 | 56.5% |
| Logistic Regression | 69.7±0.3 | 76.3±0.2 | 58.6% | 64.2±0.4 | 69.7±0.4 | 56.1% |
| KNN | 68.9±0.2 | 75.8±0.2 | 58.3% | 63.8±0.3 | 69.3±0.3 | 55.6% |

Table 12: Ablation study on different classifiers with features (Ours + features).

## B Hyperparameters and Training Details

A binary classifier is trained using the gradient boosting library XGBoost (Chen and Guestrin, 2016). We finetune the following hyper-parameters, colsample by level, colsample by node, colsample by tree, learning rate, and the number of estimators on the development set. We use the following search space: colsample by level/mode/tree is set to the same value and selected from {0.0, 0.1, 0.2, 0.3, 0.4, 0.5}, the learning rate and number of estimators are selected from {0.01, 0.1, 0.2, 0.5} and {5, 25, 50, 100}, respectively. These hyper-parameters are chosen based on the performance on the validation set.

For base QA models, we mostly following the hyperparameters used in the original work (e.g., batch size 32 & learning rate of $5 \times 10^{-5}$ for BERT-base SQuAD 1.1 model). All calibrators are trained five times, each with different data partitions and random seeds. We report the variances in the results. Our calibrator does not share its weights with the base QA models.