Calibration of Machine Reading Systems at Scale

Anonymous ACL submission

Abstract

In typical machine learning systems, an estimate of the probability of the prediction is used to assess the system's confidence in the prediction. This confidence measure is usually uncalibrated; i.e. the system's confidence in the prediction does not match the true probability of the predicted output. In this paper, we present an investigation into calibrating open setting machine reading systems such as open-domain question answering and claim verification systems. We show that calibrating such complex systems which contain discrete retrieval and deep reading components is challenging and current calibration techniques fail to scale to these settings. We propose simple extensions to existing calibration approaches that allows us to adapt them to these settings. Our experimental results reveal that the approach works well, and can be useful to selectively predict answers when question answering systems are posed with unanswerable or out-of-the-training distribution questions.

1 Introduction

With recent advances in machine reading, there has been a surge of interest in practical applications of the technology such as open-domain question answering (Karpukhin et al., 2020; Lee et al., 2019) and claim verification (Thorne et al., 2018b). Due to various scale limitations in practical settings, these systems are seldom trained end-to-end. Such systems typically make use of a RETRIEVER alongside a READER – the evidence is first retrieved from a large corpus and is then used by a machine reading model to provide an answer.

As these systems are increasingly being deployed in the real world, it is important that they are not only accurate but also trustworthy. A way to make these systems trustworthy is to indicate when they are likely to be incorrect by providing a calibrated `confidence` measure in addition to the prediction. A naive solution for this is to use the system's output probability as the confidence. However, this confidence score is often `uncalibrated` (Kuleshov and Liang, 2015; Guo et al., 2017); i.e. it is not representative of the true correctness likelihood.[1]

Previous work (Jiang et al., 2020; Jagannatha and Yu, 2020; Desai and Durrett, 2020) has shown that large language models especially suffer from miscalibration. Thus, several methods have been proposed to calibrate language models based on gradient-based calibration methods such as temperature scaling (Guo et al., 2017) and feature-based forecasters (Kuleshov and Liang, 2015). While gradient-based calibration is intuitive and easy to implement, feature-based forecasters require manual feature engineering.

In this work, we contribute a simple method to calibrate practical RETRIEVER - READER machine reading pipelines. These systems typically include a hard retrieval step which makes gradient-based calibration infeasible. Thus, we make use of the Gumbel machinery (Jang et al., 2017; Maddison et al., 2017); specifically the Gumbel top-K procedure of Vieira (2014); Xie and Ermon (2019) to obtain a differentiable sampling routine for the retrieval step. This sampler can then be combined with any gradient-based calibration technique such as Platt's scaling.

We conduct experiments on three different models – a generative and extractive open-domain question answering model and a claim verification model. We find that calibrating the RETRIEVER and the READER jointly is better than calibrating only the READER or the RETRIEVER . We also show that our approach can produce calibrated scores that can be used to selectively abstain from answering questions that are contrived or ill-posed or questions that are out-of-the-training distribu-

[1] For a perfectly calibrated system, given 100 answer predictions, each with a confidence of 0.7, we expect that 70 should be correct.

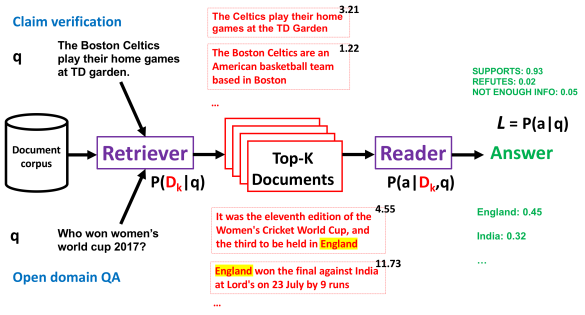

Figure 1: General architecture of the two machine reading systems considered in this paper. a) Claim verification (top half) and b) Open-domain QA (bottom half). The systems follow the same architecture and are composed of a retriever and a reader. Given the query, the retriever retrieves a set of K documents from the corpus along with scores for each of them. The reader then takes these as input and produces the output: a veracity label for claim verification and an answer span for the QA model. This can be seen as a probabilistic model with latent retrieval (D_k shown in red). The goal of this paper is to calibrate the final output probabilities $P(a|q)$.

tion. Finally, we also demonstrate how the calibration of such a system works – the calibration techniques lower the confidence of the predicted answer when the question is unanswerable or when the retriever is not able to retrieve any relevant evidence for answering the question.

2 Preliminaries

2.1 Machine Reading at Scale

Practical real-world machine reading systems such as open-domain question answering systems (Chen et al., 2017) (Karpukhin et al., 2020) (Izacard and Grave, 2020b) or claim verification systems (Hanselowski et al., 2018) rely on an information retrieval (IR) component called a RETRIEVER to reduce the search space over a large corpus of documents. This smaller set of documents is then passed to a READER model that reasons over the text and produces an answer. This setting, where the READER is not given labeled documents is referred to, in the literature, as an open-domain setting.

We now proceed to define the pipeline formally for a machine reading system in the open-domain setting.

Let $\mathscr{D} = \{d_1, \ldots, d_N\}$ denote the given corpus of documents. Let q denote the user query (a question or a claim). We denote the answer to the question or the veracity label of the claim as a. The retreiver model takes in q and scores all the documents $d \in \mathscr{D}$ to produce a set of scores:

$$\text{RETRIEVER}(d_1, \ldots, d_N|q) \longrightarrow S_{d_1}, \ldots, S_{d_N} \quad (1)$$

This formulation of the RETRIEVER is generic. This allows our method to work with any IR model such as the traditional BM25 model (Wikipedia contributors, 2004) to more modern methods such as Dense Passage Retrieval (DPR) by Karpukhin et al. (2020).

The documents are then sorted based on the scores and the k top-scoring documents are chosen. We call this set of top-K documents D_k. D_k is then given to a READER model which extracts the answer or predicts a veracity label for the claim, a. The READER can vary depending on the task. For extractive QA, the READER produces a score for each span (s_i) in the documents provided to it.

$$\text{READER}\ (q, D_k) \longrightarrow S^{Read}(s_i), s_i \in D_k \quad (2)$$

In claim verification, the READER produces a score for each veracity label: SUPPORTED, REFUTED or NOT ENOUGH INFO, which indicate whether the claim can be verified by the given set of documents.

$$\begin{aligned}\text{READER}\ (D_k, q) \longrightarrow & S_{\text{SUPPORTED}}, \\ & S_{\text{REFUTED}}, \\ & S_{\text{NOT ENOUGH INFO}},\end{aligned}$$

2.2 Calibration

We summarize below the calibration framework by Kuleshov and Liang (2015) in the context of machine reading. Given a query q, true output a, model output $\hat{a}$, and probability $P(\hat{a}|q)$ calculated over this output, a perfectly calibrated model satisfies the following condition:

$$\mathbb{P}(\hat{a} = a | P(\hat{a}|q) = p) = p \quad \forall p \in [0,1] \quad (3)$$

In simple words, for the confidence estimate $P(\hat{a}|q)$ to be calibrated, we require that $P(\hat{a}|q)$ follows the unknown true probability distribution $\mathbb{P}$.

In a multi/binary class setting, a calibrator can be learned to map the output distribution to a calibrated confidence score. However, in a machine reading setting, the space of possible documents retrieved and answers contained in them is usually very large. Thus, we only focus on a specific event set $I(q)$ of interest. The event set $I(q)$ can be defined using the outputs relevant to the deployment requirements of the machine reading model. In our work, we consider all answer candidates in the retrieved set of documents D_k: $I(q) = \{a | a \in \arg\max_{D_k} P(\hat{a}|D_k, q)\}$

2.3 Measuring Calibrated-*ness*

Calibration can be measured by computing the difference in expectation between confidence scores and accuracies.

$$\mathbb{E}_{P(\hat{a}|q)}\Big[\mathbb{P}(\hat{a}=a|P(\hat{a}|q)=p)-p\Big] \quad (4)$$

This is known as expected calibration error (ECE) (Naeini et al., 2015). Practically, ECE is estimated by partitioning the predictions in M equally spaced bins $(B_1 \ldots B_M)$ and taking the weighted average of the difference between the average accuracy and average confidence of the bins.

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |acc(B_m) - conf(B_m)| \quad (5)$$

Reliability Diagrams

Another common tool to visualize model calibration is a reliability diagram. A reliability diagram plots sample accuracy as a function of confidence for each bin. If a model is perfectly calibrated, the confidence and accuracy bars should be identical.

2.4 Calibration methods

The general algorithm used for calibrating classification models involves transforming the logits produced by the model. The parameters for this transformation are trained on a held-out calibration set $C = \{(q_i, a_i)\}_{i=1}^{N}$. This method has been shown to improve the model's ECE without a significant loss in accuracy. In our work, we use negative log-likelihood (NLL) to tune a model $P_\theta(a|q)$ to be a good probability estimate of the output answers:

$$\mathscr{L}_\theta = -\sum_{i=1}^{N} log(P_\theta(a_i|q_i)) \quad (6)$$

ML theory guarantees that NLL is minimized if and only if $P_\theta(a_i|q_i)$ recovers the ground-truth conditional distribution $\mathbb{P}(a|q)$. In the following part of this section, we describe some of these key methods.

Temperature Scaling

Temperature scaling (Guo et al., 2017) is one of the simplest methods for calibration and has been shown to be very effective. Temperature scaling allows the logits of the system's output (Z) to be scaled by a single temperature value τ. This scaling is done before the computation of the softmax.

$$Y = \texttt{softmax}(Z/\tau) \quad (7)$$

We optimize τ by maximizing $\mathscr{L}_\theta$ on the dev set.

Temperature prediction

The temperature prediction approach (Kumar and Sarawagi, 2019) extends temperature scaling to a gradient-based approach. The output logits of the classifier are featurized and passed through an MLP which predicts a temperature value. This temperature value is used to scale the logits. In contrast to temperature scaling which learns one temperature parameter for each example, in this approach, a new temperature value can be learned for each example.

$$\frac{1}{\tau} = \sigma(\texttt{MLP}(Z))$$
$$Y = \texttt{softmax}(Z/\tau)$$

Forecasters

Forecasters were introduced to calibrate structured prediction models (Kuleshov and Liang, 2015; Jagannatha and Yu, 2020). The forecaster approach introduces a feature-rich calibration model that uses various features of the model such as its logits and various uncertainties estimated to predict the confidence score. This approach generally only produces a calibrated score over a smaller set of candidate predictions referred to as the interest set $I(x)$ Previous work has successfully used gradient boosted decision trees **(XGB)** as forecasters.

3 Calibration of Machine Reading Systems

Previous work has looked at calibration in the aspect of machine reading (Jagannatha and Yu, 2020; Jiang et al., 2020). However, they do not consider the open setting in which the evidence document for each query is not provided. We are interested in determining the calibrated probability distribution of the system, $\mathbb{P}(a|q)$. In the first set of methods, we do this by calibrating the confidence of the model $P(\hat{a}=a|q)$. For a machine reading system,

$$P(\hat{a}=a|q) = \underbrace{P(D_k|q)}_{\text{conf of RETRIEVER}} \times \underbrace{P(\hat{a}=a|q,D_k)}_{\text{conf of READER}} \quad (8)$$

We discuss three possible ways to calibrate $P(\hat{a}=a|q)$

ONLY READER One way to calibrate $P(\hat{a}=a|q)$ is to assume that the RETRIEVER is perfectly accurate and perfectly calibrated. We refer to his approach in our results as ONLY READER. In this

approach, we only calibrate $P(\hat{a} = a|q, D_k)$. We can use all the previously mentioned calibration approaches for this task. For extractive QA, the output logits lie over all the possible text spans, while for fact verification we have a single logit per class. In our experiments, we show that this leads to subpar calibration.

INDIVIDUALLY CALIBRATED We explore another possible approach where we calibrate $P(D_k|q)$ and $P(\hat{a} = a|q, D_k)$ indivdually using the objectives of the RETRIEVER and READER individually. We refer to this approach as INDIVIDUALLY CALIBRATED.

The READER is calibrated as discussed in the previous section. For the RETRIEVER we use the scores as the logits to compute the probability and thus can apply the discussed calibration methods again. This method is impractical, as often in an open-domain setting gold labels for the RETRIEVER are not readily available. To overcome this we make use of the less accurate distance supervision objective used to train the RETRIEVER Ẇe show that owing to this mismatch, this approach also results in an uncalibrated system.

JOINTLY CALIBRATED Finally, we discuss our approach to calibrate the entire system using the final objective of the system. We refer to this approach as JOINTLY CALIBRATED. In this approach, we treat the documents retrieved by the retriever as a latent variable D_k.

We rewrite our objective in eqn. 6 as

$$\mathscr{L}_\theta = \sum_{D_k \in \mathscr{D}} P(D_k|q) P(\hat{a} = a|q, D_k) \qquad (9)$$

Clearly, it is infeasible to marginalize over all possible D_k (subsets of the corpus of size k). Thus, we propose a diffentiable sampler for D_k:

$$\mathscr{L}_\theta = \sum_{D_k \sim P_\theta(D_k|q)} P(\hat{a} = a|q, D_k) \qquad (10)$$

To make our calibrator differentiable, we apply the Gumbel–softmax trick (Maddison et al., 2017) and, in particular, its extension to top-K subset selection (Vieira, 2014; Xie and Ermon, 2019). In order to sample a subset of size K according to the categorical distribution given by (10), we use the well-known two-step process to massage categorical sampling into a differentiable sampling procedure which includes: 1) reparameterization of the categorical using Gumbels and 2) softening the argmax into a softmax.

The Gumbel-top-K trick further generalizes this idea and repeats the Gumbel trick K times until we have a set of the desired size. Xie and Ermon (2019) have shown that this procedure is a reasonable relaxation of the Gumbel-top-K. We refer the interested reader to their paper for more details.

4 Experimental Details

Open Domain Question Answering

Extractive We test the described calibration techniques on the open domain QA, using the pretrained models from (Karpukhin et al., 2020). We perform our experiments on the Natural Questions (NQ) dataset (Kwiatkowski et al., 2019). We randomly split our validation set into two equal parts which we will call `calib` and `valid`. We use these splits for training and tuning our calibration models respectively. We use the test set of NQ as our test set (`test`). During inference, we use the RETRIEVER to retrieve top 10 documents which are passed to the READER to extract the answer.

Generative We use the FiD model proposed by (Izacard and Grave, 2020b) for our calibration experiments. As generative models don't produce a confidence over multiple answers, we use the trick described by (Jiang et al., 2020) to generate an interest set. First we calculate the probabilities of the first generated tokens. We mask out any tokens not in the retrieved passages. Next we, select the top R tokens we find their location in the passages and calculate the probability of all continuing spans up to a certain length (of 10 tokens). We then keep the top-10 scoring spans in our candidate set.

Claim Verification

For the claim verification task, we experiment on the FEVER dataset (Thorne et al., 2018a). We use a recently published state-of-the-art model, (Liu et al., 2020), in our calibration experiments. For every test example, we retrieve 5 sentences that are provided to the claim verification model to ascertain the veracity of the claim.

Temperature based methods

For the READER -RETRIEVER setup we require two temperature parameters `t1` and `t2` for the RETRIEVER and READER respectively. We use gradient descent to optimize `t1` and `t2` by maximizing $\mathscr{L}_\theta$ on the `valid` set. For temperature prediction we add a 2-layer MLP that predicts `t1` and `t2` for each example. Once again, the optimization is performed on `valid`.

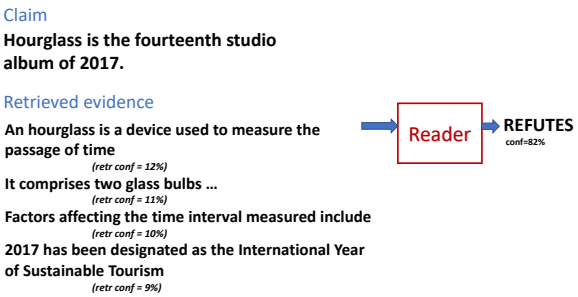

Figure 2: The READER is highly confident about its prediction, but when we incorporate the confidence of the evidence from the RETRIEVER which can identify that the sentences are irrelevant to the claim, the confidence of the prediction can be better calibrated.

Forecaster

For our forecaster, we use gradient boosted decision trees. We train the model to perform binary classification with the model's accuracy as the objective, i.e., if the model's prediction was correct, we assign a positive label to the example. We do not experiment extensively with various features as previous work has done and instead just use the raw logit scores. Similar to Jagannatha and Yu (2020), we create the interest set of the forecaster by choosing the top-3 predictions of the model, i.e., we choose the top-3K choices of the RETRIEVER over which we evaluate our READER and choose the top-3 choices.

Gumbel top-K

For the Gumbel top-K approach required to train the vector scaling and temperature prediction models, we start out with a high temperature value T_0 which we linearly decrease to T_∞. We treat these parameters as hyperparameters.

5 Results

We now present the results of the various calibration techniques in table 1. We also plot the reliability diagrams in Figure 3. We compare all the described calibration algorithms in the three settings discussed. As can be seen, in all the cases there is a benefit to JOINTLY calibrate the RETRIEVER and READER . We give some reasons for why this setting works best in the discussion section below.

5.1 Discussion

Calibrating only the READER

In all our experiments we show that calibrating the READER alone performs worse. We believe that this is because, at train time, the READER is only trained on positive documents. This makes the READER *overconfident* on documents that don't have the answer. This phenomenon has been also been discussed in Clark and Gardner (2017). We show an example in Fig 2. We also notice that adding the RETRIEVER helps more in the QA task than for claim verification. We posit that this is because in the open-domain setting, the QA passage RETRIEVER has a lower accuracy than the sentence RETRIEVER for claim verification.[2]

Calibrating INDIVIDUALLY

Our experimental results show that in almost all cases, it is detrimental to individually calibrate the READER and RETRIEVER . We believe that this is due to the RETRIEVER 's accuracy being misaligned with the final objective. In several cases, such as in QA, supervision for the RETRIEVER is not provided, and instead a distant supervision objective is used where the document is marked as positive when it contains the answer string. We show an example in Figure 1 where, for the question "Who won the women's worldcup in 2017", a document saying "world cup to be held in *England*" would be assigned a positive label as it contains the answer string "England". This mismatch in accuracy for the RETRIEVER can result in an incorrectly calibrated system. This problem has been well discussed in the literature and more recently by Izacard and Grave (2020a)

Reliability plots As can be seen from Figure 3, miscalibration results from the model being overconfident. This is evident with the blue bars being lower than the red – model accuracy is less than model confidence for several bins. We also notice that all calibration techniques address this overconfidence by rescaling the output distribution.

6 Analysis

Next, we attempt to verify the following claims:
C1: The existing approach for calibrating only the reader doesn't result in a good calibration of the overall system. Jointly calibrating the reader and the retriever model is better.
C2: Calibrated ODQA systems do better selective prediction when they are allowed to not provide answers to some questions.
C3: Calibrated ODQA systems are better at handing domain shifts in questions at test time.

[2] `QA, hits@10:0.77, CV, hits@5:0.94`
We use top-10 passages for QA and top-5 sentences for claim verification.

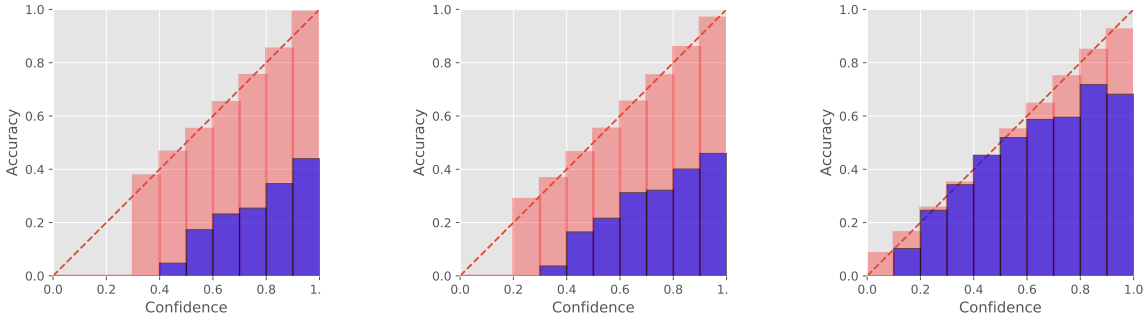

Figure 3: Reliability plots for uncalibrated *versus* INDIVIDUALLY calibrated *versus* JOINTLY calibrated on the GENERATIVE QA task using Temperature Scaling. Blue bars denote bin accuracy, red bars denote bin confidence, difference indicates miscalibration.

Task	Setting	Uncalibrated	Temp scaling	Temp predictor	Forecaster
GENERATIVE QA	GENERATOR	55.1	47.31	45.22	5.40
	INDIVIDUALLY		33.47	35.31	11.35
	JOINTLY		**3.75**	**3.56**	**4.21**
EXTRACTIVE QA	SPAN EXTRACTOR	37.1	8.56	8.11	4.68
	INDIVIDUALLY		10.32	7.42	12.74
	JOINTLY		**2.94**	**2.38**	**2.96**
CLAIM VERIFICATION	CLAIM VERIFIER	7.02	1.42	1.64	1.66
	INDIVIDUALLY		16.35	23.6	26.73
	JOINTLY		**1.15**	**1.30**	**0.98**

Table 1: Values in % ECE, (↓ is better). **INDIVIDUALLY** denotes the retriever and reader have been calibrated separately, while **JOINTLY** indiciates that calibration on a joint objective.

C4: Calibrated ODQA systems are better at handling unanswerable questions at test time.

6.1 Temperature prediction

To try to understand the effectiveness of the temperature predictor model, we probed the temperature predictor model and analyzed the predicted temperature values. We used the model where only the READER is calibrated so as to make the model as simple as possible. The major reason why most of these models remain uncalibrated is that they are overconfident about their prediction thus requiring a $\tau > 1$ for calibration. Our initial assumptions were that the temperature predictor could identify if the model's output scores were peaky or flat and could correct each by predicting an appropriate temperature. To test this, we calculated the correlation between the entropy of the span scores versus τ. Surprisingly, we were able to only find a weak correlation of 0.18. We leave further investigations on how the temperature predictor learns to predict an individual temperature to future work.

6.2 Selective Prediction for Machine Reading

One key use of confidence estimation is selective prediction. The selective prediction setting allows the model to decide whether it wants to make a prediction or abstain on each given test point. Selective prediction has been a long-standing research area in machine learning (Chow, 1957; El-Yaniv et al., 2010).

We investigate how different calibration methods perform on the task of selective prediction. There have been some recent efforts to understand selective prediction for QA models with regard to domain shift; Kamath et al. (2020) investigate how forecasters can be effectively used as calibrators to predict when a model should abstain from providing an answer. We further this investigation in the open-domain setting to see if different calibration techniques can improve the model's performance on the selective prediction task. The evaluation metric used to judge a model's effectiveness in learning to abstain is the area under the risk-coverage curve.

Given an input q, the model's prediction $\hat{a}$ along with the confidence of the prediction $P(\hat{a} = a|q)$ and a threshold τ, our model predicts the the an-

Task	Setting	AURC
EXTRACTIVE QA	UNCALIBRATED	47.39
	FORECASTER	44.21
	TEMP SCALING	43.68
	TEMP PREDICTION	42.64
	BEST POSSIBLE	26.71
GENERATIVE QA	UNCALIBRATED	53.57
	FORECASTER	39.10
	TEMP SCALING	44.85
	TEMP PREDICTION	43.21
	BEST POSSIBLE	22.25
CLAIM VERIFICATION	UNCALIBRATED	11.04
	FORECASTER	3.53
	TEMP SCALING	10.96
	TEMP PREDICTION	9.99
	BEST POSSIBLE	2.76

Table 2: Area under Risk-Coverage curve. ↓ is better

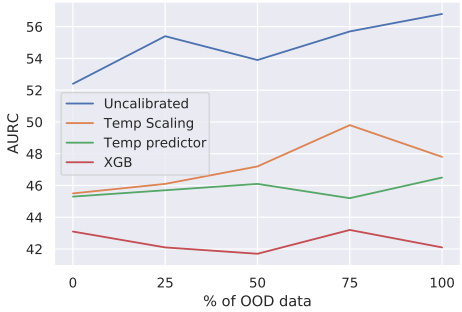

Figure 4: Area under risk coverage curves using different calibration techniques

swer $\hat{a}$ if $P(\hat{a} = a|q) \geq \tau$. For the test set and a value of τ there is an associated *risk*: the fraction of the test set that the model answers incorrectly, and *coverage*: the fraction of the test set the model makes a prediction on. As τ increases, so do the risk and coverage. We plot risk vs coverage as τ varies and report the area under the risk-coverage curve (AURC). Our results are shown in table 2. We can infer from the results that all calibration methods help reduce the AUCR to some extent however the Temperature predictor is able to perform the best on extractive QA while the forecaster is the best on claim verification and generative QA indicating that improving model calibration can also help for the task of selective prediction in the setting of machine reading.

Domain Adaptation With the increasing use of machine reading systems in the wild, a common problem encountered by them is that they are not resilient to inputs that do not come from the distribution of the data they were trained on. A method of selective prediction is often employed, where the model the model can abstain from answering the question. (Kamath et al., 2020) show that training a seperate model to distinguish between in- and out-of- domain helps in doing selective prediction. We show that a well calibrated model is able to perform better on the selective prediction setting even when the calibration step has no access to an out of distribution dataset. In our experiments, we calibrate a trained on NQ FiD model on the FiD dev set. We then evaluate the performance on different splits which contain varying percentages of out-of-distribution data (TriviaQA) (Joshi et al., 2017). We plot the AURC with different splits containing different percentages of OOD questions in figure 4. We notice that an uncalibrated model gets significantly worse when the amount of OOD samples are added. However our calibation techniques are able to mitigate this and apart from maintaining a steady AURC with increasing OOD samples, they are also result in much lower AURC with the Forecaster (XGB) performing the best.

Unanswerable Questions Another challenge that a user facing QA system can encouter is malformed questions. These include questions that were not probably questions, for example a user query containing a named entity which is a question or a question that cannot be answered because it contains a false premise. To investigate if a calibrated model can be used to abstain from answering such questions, we evaluate our approaches on the set of unanswerable questions proposed by (Asai and Choi, 2020). We plot Risk-Coverage curves for different calibration techniques in Figure 5. We find all calibration techniques help in performing selective prediction when compared to an uncalibrated model. However, the Forecaster outperforms all other methods. To exemplify how calibration techniques can help the model abstain we provide two examples in table 3. It can be seen that all calibration techniques are able to lower the confidence of the predicted answer in cases when the question is unasnwerable.

RETRIEVER mistakes Another common seen scenario in an open domain setting is when the RETRIEVER is not able to provide any relevant passages. In such cases because the READER is generally trained on only correct passages, it still produces a high confidence for the incorrect answer. We show that calibration especially methods

Question	Retrieved Passage titles	Model confidences (READER × RETRIEVER)
who do you think you are book pdf	Book of Ryan Ectaco jetBook Comparison of e-book formats	**Uncalibrated**: $0.97 \times 0.40 = 0.39$ **Temp Scaling**:$0.78 \times 0.22 = 0.17$ **Temp Prediction**:$0.63 \times 0.17 = 0.10$ **Forecaster**: 0.11
zombies are a particular challenge for which of the following theories of mind	David Chalmers Theory of mind Philosophy of mind	**Uncalibrated**:$0.99 \times 0.84 = 0.84$ **Temp Scaling**: $0.98 \times 0.40 = 0.40$ **Temp Prediction**: $0.94 \times 0.33 = 0.31$ **Forecaster**: 0.34

Table 3: Examples of unanswerable questions. We show how each calibration approach is able to lower the confidence of the incorrect answer

Question and Passage	Model confidences
how many episodes of corrie has there been Clarkson (TV series):. . . The series ran for **ten** episodes, during a weekly airing schedule . . .	**Uncalibrated**: $0.99 \times 0.49 = 0.49$ **Temp Scaling**:$0.99 \times 0.23 = 0.22$ **Temp Prediction**:$0.90 \times 0.18 = 0.16$ **Forecaster**: 0.06
what is in a pat o brien hurricane Sucker hole: . . . Sucker hole is a colloquial term referring to a short spate of **good weather** . . .	**Uncalibrated**: $0.99 \times 0.49 = 0.49$ **Temp Scaling**:$0.99 \times 0.24 = 0.23$ **Temp Prediction**:$0.95 \times 0.21 = 0.20$ **Forecaster**: 0.07

Table 4: Examples of questions where the RETRIEVER fetches the wrong passages

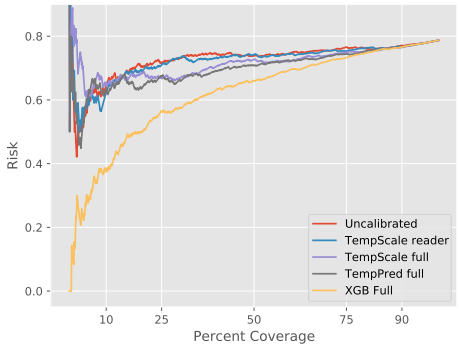

Figure 5: Risk coverage curve for unanswerable questions

that take into account the RETRIEVER confidences can mitigate this by lowering the confidence of the answer. We provide two such examples in table 4

7 Related Work

Obtaining calibrated confidence scores for NLP tasks has recently gained attention. Jagannatha and Yu (2020) and Jiang et al. (2020) study how forecasters can be used and what features can be useful to calibrate the confidence of QA models. Kamath et al. (2020) study calibration in the context of selective answering, i.e., learning when QA models should abstain from answering questions. They show that training a forecaster to predict the model's confidence can perform well when facing a distributional shift. Su et al. (2019) also investigate selective answering using a probe in the model to determine the model's confidence.

Also related to our work is *uncertainity estimation* (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017) as model uncertainities can be seen as confidence scores. In NLP, Xiao and Wang (2019) propose an approach to characterize model and data uncertainties for various NLP problems. Wang et al. (2019) use uncertainty estimation for confidence estimation in MT. Dong et al. (2018) study confidence estimation for semantic parsing. We are the first to study calibration of open-domain machine reading systems.

8 Discussion and Conclusion

In this paper, we analyzed how various calibration techniques can be adopted to open-domain machine reading systems which are now being used in user-facing scenarios. We showed that in such systems that include a retriever, calibrating the system's confidence is not trivial and we proposed a technique that allows calibration of the system *jointly*. Finally, we also provide an analysis on how the calibration techniques can help the model abstain from answering a question especially in settings where the model's prediction can be incorrect due to malformed or out-of-domain questions. While we do not find evidence to prove that one calibration method (e.g. a gradient-based method) is better that the other (e.g. a forecaster approach), it would be important to investigate these questions with more nuanced human studies.

9 Ethical Considerations

In recent years, deep learning approaches have been the main models of choice for practical machine reading systems. However, these systems are often overconfident in their predictions. A calibrated confidence score would help system users better understand the system's decision making. Our work introduces a simple and general way for calibrating these systems. While our models are not tuned for any specific application domain, our methods could be used in sensitive contexts such as legal or healthcare settings, and it is also essential that any work using our method undertake additional quality assurance and robustness testing before using it in their setting. The datasets used in our work do not contain any sensitive information to the best of our knowledge.

References

Akari Asai and Eunsol Choi. 2020. Challenges in information seeking qa: Unanswerable questions and paragraph retrieval. *arXiv preprint arXiv:2010.11915*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Chi-Keung Chow. 1957. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers*, (4):247–254.

Christopher Clark and Matt Gardner. 2017. Simple and effective multi-paragraph reading comprehension. *arXiv preprint arXiv:1710.10723*.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*.

Li Dong, Chris Quirk, and Mirella Lapata. 2018. Confidence modeling for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 743–753, Melbourne, Australia. Association for Computational Linguistics.

Ran El-Yaniv et al. 2010. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(5).

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*.

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018. Ukp-athene: Multi-sentence textual entailment for claim verification. *arXiv preprint arXiv:1809.01479*.

Gautier Izacard and Edouard Grave. 2020a. Distilling knowledge from reader to retriever for question answering. *arXiv preprint arXiv:2012.04584*.

Gautier Izacard and Edouard Grave. 2020b. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Abhyuday Jagannatha and Hong Yu. 2020. Calibrating structured output predictors for natural language processing. *arXiv preprint arXiv:2004.04361*.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know when language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. *arXiv preprint arXiv:2006.09462*.

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

Volodymyr Kuleshov and Percy S Liang. 2015. Calibrated structured prediction. *Advances in Neural Information Processing Systems*, 28:3474–3482.

Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *arXiv preprint arXiv: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.

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

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained fact verification with kernel graph attention network. In *Proceedings of ACL*.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Lixin Su, Jiafeng Guo, Yixin Fan, Yanyan Lan, and Xueqi Cheng. 2019. Controlling risk of web question answering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and verification (fever) shared task. *arXiv preprint arXiv:1811.10971*.

Tim Vieira. 2014. Gumbel-max trick and weighted reservoir sampling.

Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. 2019. Improving back-translation with uncertainty-based confidence estimation. *CoRR*, abs/1909.00157.

Wikipedia contributors. 2004. Okapi bm25— Wikipedia, the free encyclopedia. [Online; accessed 22-July-2004].

Yijun Xiao and William Yang Wang. 2019. Quantifying uncertainties in natural language processing tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7322–7329.

Sang Michael Xie and Stefano Ermon. 2019. Reparameterizable subset sampling via continuous relaxations. *arXiv preprint arXiv:1901.10517*.