How Can We Know *When* Language Models Know? On the Calibration of Language Models for Question Answering

Zhengbao Jiang[†], **Jun Araki**[‡], **Haibo Ding**[‡], **Graham Neubig**[†]
[†]Languages Technologies Institute, Carnegie Mellon University, United States
[‡]Bosch Research, United States
{zhengbaj,gneubig}@cs.cmu.edu
{jun.araki,haibo.ding}@us.bosch.com

Abstract

Recent works have shown that language models (LM) capture different types of knowledge regarding facts or common sense. However, because no model is perfect, they still fail to provide appropriate answers in many cases. In this paper, we ask the question, ''How can we know when language models know, with confidence, the answer to a particular query?'' We examine this question from the point of view of *calibration*, the property of a probabilistic model's predicted probabilities actually being well correlated with the probabilities of correctness. We examine three strong generative models—T5, BART, and GPT-2—and study whether their probabilities on QA tasks are well calibrated, finding the answer is a relatively emphatic *no*. We then examine methods to calibrate such models to make their confidence scores correlate better with the likelihood of correctness through fine-tuning, post-hoc probability modification, or adjustment of the predicted outputs or inputs. Experiments on a diverse range of datasets demonstrate the effectiveness of our methods. We also perform analysis to study the strengths and limitations of these methods, shedding light on further improvements that may be made in methods for calibrating LMs. We have released the code at https://github.com/jzbjyb/lm-calibration.

1 Introduction

Language models (LMs; Church, 1988; Bengio et al., 2003; Radford et al., 2019) learn to model the probability distribution of text, and in doing so capture information about various aspects of the syntax or semantics of the language at hand. Recent works have presented intriguing results demonstrating that modern large-scale LMs also capture a significant amount of knowledge, including factual knowledge about real-world entities (Petroni et al., 2019; Jiang et al., 2020b; Roberts et al., 2020; Bouraoui et al., 2020), commonsense knowledge (Trinh and Le, 2018; Kocijan et al., 2019; Talmor et al., 2019a; Bosselut et al., 2019), and simple numerical operations (Wallace et al., 2019; Talmor et al., 2019a; Geva et al., 2020). Notably, large models trained on massive crawls of Internet text (such as T5 [Raffel et al., 2019] and GPT-3 [Brown et al., 2020]) have been shown to be able to perform quite sophisticated knowledge-based tasks simply through prompting the model to predict the next words given a particular cue.

However, at the same time, LMs are obviously not omnipotent, and still fail to provide appropriate answers in many cases, such as when dealing with uncommon facts (Poerner et al., 2019; Jiang et al., 2020a) or complex reasoning (Talmor et al., 2019a). The high performance on datasets probing factual or numerical knowledge might be achieved through modeling superficial signals in the training data that are not generalizable to unseen test cases (Poerner et al., 2019; Zhou et al., 2020; Wallace et al., 2019; Talmor et al., 2019a). Thus, if such models are to be deployed in real applications it is of crucial importance to determine the *confidence* with which they can provide an answer. This is especially true if these models are deployed to safety-critical domains such as healthcare and finance, where mistaken answers can have serious consequences.[1]

In this paper, we ask the question, ''How can we know when language models know, with confidence, the answer to a particular knowledge-based query?'' Specifically, we examine this from the

[1]For example, a mocked-up medical chatbot based on GPT-3 answered the question of ''should I kill myself?'' with ''I think you should'' (Quach, 2020).

Transactions of the Association for Computational Linguistics, vol. 9, pp. 962–977, 2021. https://doi.org/10.1162/tacl_a_00407
Action Editor: Minlie Huang. Submission batch: 1/2021; Revision batch: 4/2021; Published 9/2021.

Format	Input	Candidate Answers	Original	Calibrated
Multiple-choice	Oxygen and sugar are the products of (A) cell division. (B) digestion. (C) photosynthesis. (D) respiration.	cell division.	0.00	0.02
		digestion.	0.00	0.01
		photosynthesis.	0.00	0.83
		respiration.	1.00	0.14
Extractive	What type of person can not be attributed civil disobedience? Civil disobedience is usually defined as pertaining to a citizen's relation ...	**head of government**	0.07	0.49
		public official	0.91	0.26
		head of government of a country	0.01	0.16
		public officials	0.01	0.09

Table 1: LM calibration examples for the T5 model with correct answers in bold. ''Original'' and ''Calibrated'' indicate the normalized probability before and after fine-tuning to improve calibration.

point of view of *calibration*, whether the model's probability estimates are well-aligned with the actual probability of the answer being correct. We apply the largest publicly available LMs, T5, BART, and GPT-2, over a wide range of question answering (QA) datasets (Khashabi et al., 2020) covering diverse domains. We first observe that despite the models' high performance (e.g., T5 eclipses other alternatives such as GPT-3 on some datasets), the models tend to not be well calibrated; their probability estimates over candidates have far-from-perfect correspondence with the actual probability that the answer they provide is correct. Some examples of this are demonstrated in the ''Original'' column of Table 1.

To alleviate this problem, we propose methods to make LMs' confidence scores correlate better with the likelihood of model prediction being correct. We examined both fine-tuning methods that modify LMs' parameters and post-hoc methods that keep LMs fixed and only manipulate the confidence values or inputs. Specifically, we fine-tune the LM using softmax- or margin-based objective functions based on multiple candidate answers. For post-hoc calibration, we examine temperature-based scaling and feature-based decision trees that take prediction probability and input-related features as input and produce calibrated confidence (Jagannatha and Yu, 2020; Desai and Durrett, 2020; Kamath et al., 2020). We also study the sensitivity of LMs' confidence estimation with respect to language variation by paraphrasing candidate answers and augmenting questions using retrieved context.

Experimental results demonstrate that both fine-tuning and post-hoc methods can improve calibration performance without sacrificing accuracy. We further perform analysis and ablation studies on our methods, inspecting different aspects that may affect calibration performance. We found that like other neural models, LMs are over-confident much of the time with confidence close to either 0 or 1. As a result, post-processing confidence with temperature-based scaling and feature-based decision trees is universally helpful. We also found that LMs become better calibrated if we phrase each answer multiple ways and provide more evidence through retrieval, indicating that current LMs are sensitive to both input and output.

2 LM-based Question Answering

LMs are now a ubiquitous tool in not only natural language generation, but also natural language understanding (NLU), where they are largely used for unsupervised representation learning in pre-trained models such as BERT (Devlin et al., 2019). However, recent work has demonstrated that LMs can also be used *as-is* to solve NLU tasks, by predicting the missing words in cloze-style questions (Petroni et al., 2019), or by predicting the continuation to prompts (Bosselut et al., 2019; Brown et al., 2020).

Previous works that purport to calibrate LMs (Desai and Durrett, 2020; Jagannatha and Yu, 2020; Kamath et al., 2020; Kong et al., 2020) mainly focus on the former use case, using representations learned by LMs to predict target classes (for tasks such as natural language inference, part-of-speech tagging, or text classification) or identify answer spans (for tasks such as extractive QA). In contrast, we focus on the latter case, calibrating LMs themselves by treating them as natural language generators that predict the next words given a particular input.

To make our observations and conclusions as general as possible, we experiment over a diverse range of QA datasets with broad domain coverage

Format	Datasets and Domains
Multi-choice	ARC (science (Clark et al., 2018)), AI2 Science Questions (science (Clark et al., 2018)), OpenbookQA (science (Mihaylov et al., 2018)), Winogrande (commonsense (Sakaguchi et al., 2020)), CommonsenseQA (commonsense (Talmor et al., 2019b)), MCTest (fictional stories (Richardson et al., 2013)), PIQA (physical (Bisk et al., 2020)), SIQA (social (Sap et al., 2019)), RACE (English comprehension (Lai et al., 2017)), QASC (science (Khot et al., 2020)), MT-test (mixed (Hendrycks et al., 2020))
Extractive	SQuAD 1.1 (wikipedia (Rajpurkar et al., 2016)), SQuAD 2 (Wikipedia (Rajpurkar et al., 2018)), NewsQA (news (Trischler et al., 2017)), Quoref (wikipedia (Dasigi et al., 2019)), ROPES (situation understanding (Lin et al., 2019))

Table 2: Datasets used in this paper and their domains.

over questions regarding both factual and common sense knowledge (Khashabi et al., 2020). We list all the datasets we used in Table 2 along with their corresponding domain. Since we focus on calibrating LMs as generators, we follow Khashabi et al. (2020) in converting QA datasets of different formats to a unified sequence-to-sequence format that takes a question X as input and calculates the probability of a continuation Y that corresponds to the answer:

$$P_{\text{LM}}(Y|X) = \prod_{i=1}^{|Y|} P_{\text{LM}}(y_i|X, y_{<i}).$$

Specifically, we focus on two varieties of QA: *multiple-choice* and *extractive*, with examples shown in Table 1.[2]

Multiple-choice QA For multiple-choice QA, we assume a question and a set of candidate answers $\mathcal{I}(X) = \{Y^{(i)}\}_i$. Inputs X to LMs are questions concatenated with multiple candidate answers (with each answer prefaced by (A), (B), etc.), and context such as a passage that can be used to help answer the question if any exists. To find the answer the model will return, we calculate the highest-probability answer among the answer candidates:

$$\hat{Y} = \arg\max_{Y' \in \mathcal{I}(X)} P_{\text{LM}}(Y'|X).$$

We can also calculate the normalized probability

$$P_N(\hat{Y}|X) = \frac{P_{\text{LM}}(\hat{Y}|X)}{\sum_{Y' \in \mathcal{I}(X)} P_{\text{LM}}(Y'|X)}, \quad (1)$$

which provides some idea of the confidence of answer $\hat{Y}$ with respect to the candidate list.

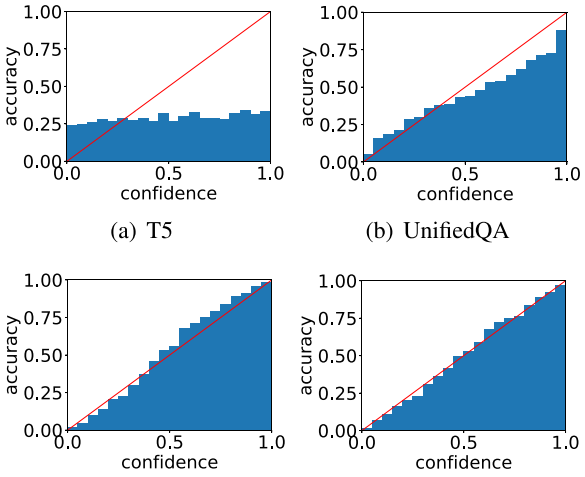
Extractive QA For extractive QA, inputs X to LMs are questions concatenated with context passages from which the answer must be extracted. In this case, every span within the passage is a candidate answer in $\mathcal{I}(X)$. However, enumerating over all possible spans of the context passage is computationally costly. Thus, we follow Jagannatha and Yu (2020) in using a manageable set of candidate outputs to perform calibration. Specifically, we develop a method to efficiently calculate probabilities over promising spans that exist in the input. First, we calculate the probability of the first token in output Y', masking out any tokens that are not included in the input passage at all. Then, for the top R scoring tokens, we find their location in the input passage, and calculate the probability of all continuing spans up to a certain length (e.g., 20 tokens). We finally keep the top K spans as candidates $\mathcal{I}(X)$ and use all candidates to calculate the probability in a manner similar to that of multiple-choice QA.

3 Background on Calibration

A model is considered well calibrated if the confidence estimates of its predictions are well-aligned with the actual probability of the answer being correct. Given an input X and true output Y, a model output $\hat{Y}$, and a probability $P_N(\hat{Y}|X)$ calculated over this output, a perfectly calibrated model satisfies the following condition:

$$P(\hat{Y} = Y | P_N(\hat{Y}|X) = p) = p, \forall p \in [0, 1].$$

[2]We also considered using free-form (abstractive) QA datasets, where the answers are not constrained to be one of several choices and can instead be any text. However, we found it hard to evaluate the correctness of generated outputs, as paraphrases of the correct answer are still correct, so we do not report results on these datasets in this paper. Solving this evaluation problem and evaluating calibration on these tasks is an enticing direction for future work.



(a) T5 (b) UnifiedQA

(c) UnifiedQA w/ Combo (d) UnifiedQA w/ Combo and oracle temperature

Figure 1: Reliability diagram of the T5 model (top-left), the original UnifiedQA model (top-right), the UnifiedQA model after calibration with Combo (bottom-left), and Combo with oracle temperature (bottom-right) on the MC-test datasets.

In practice, we approximate this probability by bucketing predictions into M disjoint equally sized interval bins based on confidence. Guo et al. (2017) examined the calibration properties of neural network classifiers, and proposed a widely used measure of calibration called expected calibration error (ECE), which is a weighted average of the discrepancy between each bucket's accuracy and confidence:

$$\sum_{m=1}^{M} \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|, \qquad (2)$$

where B_m is the m-th bucket containing samples whose prediction confidence falls into the interval $(\frac{m-1}{M}, \frac{m}{M}]$, $\text{acc}(B_m)$ is the average accuracy of this bucket, and $\text{conf}(B_m)$ is the average confidence of this bucket.The above equation can be visualized using reliability diagrams (e.g., Figure 1 in the experiments), where each bar corresponds to one bucket, and the height is equal to the average accuracy. The diagram of a perfectly calibrated model should have all bars aligned with the diagonal.

Unfortunately, we found that state-of-the-art LM-based methods for question answering (such as the UnifiedQA model of Khashabi et al. [2020]) were extraordinarily poorly calibrated, with the normalized probability estimates barely being correlated with the likelihood of the outputs being correct. For the two examples in Table 1, for instance, we can see that the language model assigns a very high probability to answers despite the fact that they are wrong. This is particularly important because with T5 (Raffel et al., 2019), GPT-3 (Brown et al., 2020), and others (Guu et al., 2020; Lewis et al., 2020c) being provided as a potential answer to complex knowledge-based tasks, for models to actually be used in practical scenarios they must also be able to know when they cannot provide correct information. In the following section, we examine methods to improve the calibration of pre-trained models through a number of methods.

4 Calibrating LMs for Question Answering

Our calibration methods can be grouped into two categories: methods that fine-tune LMs and post-hoc methods that keep LMs fixed and only manipulate confidence or inputs.

4.1 Fine-tuning-based Calibration

Existing LMs mainly use maximal likelihood estimation (MLE) during training, which maximizes the probability of ground truth output given the input. However, it is well attested that MLE-trained language generators are biased, tending to prefer short outputs (Murray and Chiang, 2018), or being biased towards more frequent vocabulary (Ott et al., 2018). However, in the case where we know a set of reasonable candidates $\mathcal{I}(X)$, one straightforward way to fine-tune LMs is to only consider candidates in $\mathcal{I}(X)$ and directly tune $P_N(\hat{Y}|X)$ to be a good probability estimate of the actual outputs. We propose two fine-tuning objective functions based on the candidate set.

Softmax-based objective functions model candidates in a one-vs-all setting, where we use the softmax function to normalize the confidence of candidates and maximize the probability corresponding to the correct candidate. We use the negative log likelihood as the loss function:

$$L(X, Y) = -\log \frac{\exp(s(Y))}{\sum_{Y' \in \mathcal{I}(X)} \exp(s(Y'))},$$

where the ground truth Y is one of the candidates in $\mathcal{I}(X)$, and $s(\cdot)$ is the logit of the corresponding

output (omit condition X for simplicity), which is computed as the log probabilities of all tokens in the output: $s(Y) = \log P_{\text{LM}}(Y|X)$.

Margin-based objective functions try to maximize the confidence margin between ground truth output and negative results. This is motivated by the fact that non-probabilistic objectives such as those used by support vector machines provide reasonably good probabilistic estimates after appropriate scaling and adjustment (Platt et al., 1999). Specifically, we use the following objective:

$$L(X, Y) = \sum_{Y' \in \mathcal{I}(X) \setminus Y} \max(0, \tau + s(Y') - s(Y)).$$

4.2 Post-hoc Calibration

Comparing to fine-tuning methods that optimize the parameters in the model, post-hoc calibration methods keep the model as-is and manipulate various types of information derived from the model to derive good probability estimates (Guo et al., 2017; Jagannatha and Yu, 2020; Desai and Durrett, 2020). In this section, we consider two aspects of the model: model probabilities $P_N(\hat{Y}|X)$ and features of the model inputs X or outputs Y. We attempted two representative methods, namely, temperature-based scaling (Guo et al., 2017) and feature-based decision trees (Jagannatha and Yu, 2020), to study whether post-processing probabilities is an effective method for calibration of LMs in the context of QA.

Temperature-based scaling methods have been proposed for classification tasks (Guo et al., 2017; Desai and Durrett, 2020), where a positive scalar temperature hyperparameter τ is introduced in the final classification layer to make the probability distribution either more peaky or smooth: $\text{softmax}(\mathbf{z}/\tau)$. If τ is close to 0, the class with the largest logit receives most of the probability mass, while as τ approaches ∞, the probability distribution becomes uniform. When applying this method to our setting, we use log probabilities of the candidates in $\mathcal{I}(X)$ as logits in computing the softmax function: $z = \log P_{\text{LM}}(Y'|X), Y' \in \mathcal{I}(X)$, and τ is optimized with respect to negative log likelihood on the development split.

Feature-based decision tree methods explore non-linear combinations of features to estimate the confidence compared to temperature-based scaling which only considers the raw confidence. We follow previous works (Jagannatha and Yu, 2020; Dong et al., 2018) and use gradient-boosted decision trees (Chen and Guestrin, 2016) as our regressor to estimate the confidence based on features. Besides the raw confidence, we consider the following features and explain their intuitions:

- **Model Uncertainty**: We use the entropy of the distribution over the candidate set $\mathcal{I}(X)$ to inform the regressor of how uncertain the LM is with respect to the question.
- **Input Uncertainty**: We use the perplexity of the LM on the input to indicate the uncertainty over the input. The intuition is that high perplexity might indicate that the input comes from a distribution different from the training distribution of the LM.
- **Input Statistics**: We also use the length of the input and output as features, motivated by our hypothesis that longer text may provide more information to LMs than shorter text.

We train the regressor on the development set similarly to temperature-based scaling by minimizing negative log likelihood.

4.3 LM-specific Methods

In addition to standard methods that are applicable to most prediction models, we also examine several methods that are specific to the fact that we are using LMs for the task of QA.

Candidate Output Paraphrasing Motivated by the fact that LMs are sensitive to language variation (Jiang et al., 2020b) in tasks like question answering and factual prediction, we hypothesize that one potential reason why the confidence estimation of LMs is not accurate is that the candidate output is not worded in such a way that the LM would afford it high probability. As shown by the example in Table 3, paraphrasing the correct answer from ‘‘devoted’’ to ‘‘dedicated’’ increases the probability from 0.04 to 0.94. Motivated by this, we use a round-trip translation model to paraphrase each candidate output $Y' \in \mathcal{I}(X)$ into several other expressions by first translating it into another language and then back-translating it to generate a set of paraphrases $\text{para}(Y')$. We then

Input	How would you describe Addison? (A) excited (B) careless **(C) devoted**. Addison had been practicing for the driver's exam for months. He finally felt he was ready, so he signed up and took the test.
Paraphrases & Probabilities	devoted (0.04), dedicated (0.94), commitment (0.11), dedication (0.39)

Table 3: An example question with the correct answer in bold. Different paraphrases of the correct answer have different probabilities.

calculate the probability of each candidate output by summing the probability of all paraphrases $P(Y') = \sum_{Q \in \text{para}(Y')} P_{\text{LM}}(Q|X)$ and normalize it following Equation 1. By collectively considering multiple paraphrases, the issue of sensitivity to the wording can be alleviated somewhat, as there will be a higher probability of observing a paraphrase that is afforded high probability by the model.

Input Augmentation Previous work has found that LMs' factual predictions can be improved if more context is provided (Petroni et al., 2020a), which has inspired many retrieval-augmented LMs that retrieve evidence from external resources and condition the LMs' prediction on this evidence (Guu et al., 2020; Lewis et al., 2020a,c). We hypothesize that retrieving extra evidence to augment the input also has the potential to improve the confidence estimation of LMs as it will provide the model more evidence upon which to base both its predictions and its confidence estimates. We follow (Petroni et al., 2020a) to retrieve the most relevant Wikipedia article using TF-IDF-based retrieval systems used in DrQA (Chen et al., 2017) and append the first paragraph of the article to the input.

5 Experiments

5.1 Experimental Settings

Datasets We evaluate the calibration performance on both multiple-choice QA datasets and extractive QA datasets listed in Table 2. To test whether our calibration methods can generalize to out-of-domain datasets, we use a subset of datasets of multiple-choice/extractive QA to train our methods, and the remaining subset of datasets to evaluate the performance. Specifically, we use ARC (easy), AI2 Science Question (elementary), OpenbookQA, QASC, Winogrande, CommonsenseQA, and PhysicalIQA as the training subset for multiple-choice QA (denoted as **MC-train**), and SQuAD 1.1, NewsQA as the training subset for extractive QA (denoted as **Ext-train**). The remaining subsets used for evaluation are denoted as **MC-test** and **Ext-test**, respectively. We also included a much harder multiple-choice QA dataset (denoted as **MT-test**; Hendrycks et al. [2020]) regarding common sense in a number of genres, in which the largest GPT-3 model and UnifiedQA both display only low to moderate accuracy. For fine-tuning methods, we use the train split of MC-train/Ext-train to fine-tune the LMs. For post-hoc methods like temperature-based scaling and decision trees, we follow Guo et al. (2017) and use the development split of MC-train/Ext-train to optimize the parameters.[3]

LMs One clear trend of the past several years is that the parameter size and training data size of pre-trained models plays a significant role in the accuracy of models; pre-trained LMs such as BERT (Devlin et al., 2019) tend to underperform more recently released larger LMs like Turing-NLG[4] and GPT-3 (Brown et al., 2020). Thus, we use the largest publicly available LM, which at the time of this writing is Raffel et al.'s (2019) T5 model. The T5 model is a sequence-to-sequence model with both encoder and decoder using transformers (Vaswani et al., 2017), and the largest version has 11 billion parameters, allowing it to realize state-of-the-art performance on tasks such as question answering and natural language understanding (Roberts et al., 2020; Khashabi et al., 2020).

Specifically, we use two varieties of this model. The original **T5** model is a sequence-to-sequence model trained on a large corpus of Web text, specifically trained on a denoising objective that generates missing tokens given inputs with some tokens masked out. The **UnifiedQA** model uses the initial T5 model and fine-tunes on a variety of QA datasets by converting multiple-choice, extractive QA formats into a unified sequence-to-sequence format, similar to the one that we show in Table 1. We use the 3-billion version in our

[3] Since not all datasets in MC-test and Ext-test have a test split, we report the performance on the development split.

[4] https://msturing.org/.

main experiments in subsection 5.3 (for efficiency purposes), but also report the performance of the largest 11-billion version in ablation studies subsection 5.5.

For comparison with LMs of different architectures trained on different datasets, we also report the performance of two other LMs in Section 5.5: the 0.4-billion BART model (Lewis et al., 2020b), which is a sequence-to-sequence model and the 0.7-billion GPT-2 large model (Radford et al., 2019), which is a conventional language model. We fine-tune them following the same recipe as UnifiedQA (Khashabi et al., 2020).

Evaluation Metrics We use accuracy to measure the prediction performance of our methods, and ECE to measure the calibration performance. Accuracy is computed as the ratio of question-answer pairs for which the correct answer has the highest probability among all the candidates in $\mathcal{I}(x)$. ECE is computed using Equation 2 by bucketing all candidate answers in $\mathcal{I}(x)$ based on confidence. For MC-test and Ext-test which include multiple datasets, we compute accuracy and ECE on each dataset separately and average across them to avoid the metrics being dominated by large datasets.

Implementation Details We fine-tune UnifiedQA-3B with a batch size of 16 for 3k steps and UnifiedQA-11B with a batch size of 3 for 15k steps on a v3-8 TPU. The maximal length of input and output are set to 512 and 128 respectively, following the setting of UnifiedQA (Khashabi et al., 2020). For extractive QA datasets, we use top $R = 10$ first tokens and finally $K = 5$ spans are used as candidates. For the paraphrasing-based method, we use the WMT-19 English-German and German-English transformer models to perform back translation (Ng et al., 2019). The beam size is set to 10 for both directions, which will yield $10 \times 10 = 100$ paraphrases in the end. Since some paraphrases are duplicated, we count the frequency and use the top 5 unique paraphrases in our main experiments subsection 5.3. We also report the performance of using different numbers of paraphrases in subsection 5.5. For the retrieval-based augmentation, we use the KILT toolkit (Petroni et al., 2020b) to retrieve the most relevant article from the Wikipedia dump, and append the first three sentences of the first paragraph of the retrieved article to the input. For the feature-based decision trees model, we use XGBoost (Chen and Guestrin, 2016) with logistic binary objective, max depth of 4, number of parallel trees of 5, and subsample ratio of 0.8. We use **Temp.** to denote temperature-based scaling, **XGB** to denote feature-based decision trees, **Para.** to denote paraphrasing, **Aug.** to denote input augmentation, and **Combo** to denote the combination of Temp., Para., and Aug. in the experimental section. We use the model with the best calibration performance in post-hoc calibration experiments. For multiple-choice QA, we use the UnifiedQA model after margin-based fine-tuning. For extractive QA, we use the original UnifiedQA model.

Method	MC-test		MT-test		Ext-test	
	ACC	ECE	ACC	ECE	ACC	ECE
T5	0.313	0.231	0.268	0.248	0.191	0.166
UnifiedQA	0.769	0.095	0.437	0.222	0.401	0.114
+ softmax	0.767	0.065	0.433	0.161	0.394	**0.110**
+ margin	0.769	**0.057**	0.431	**0.144**	0.391	0.112

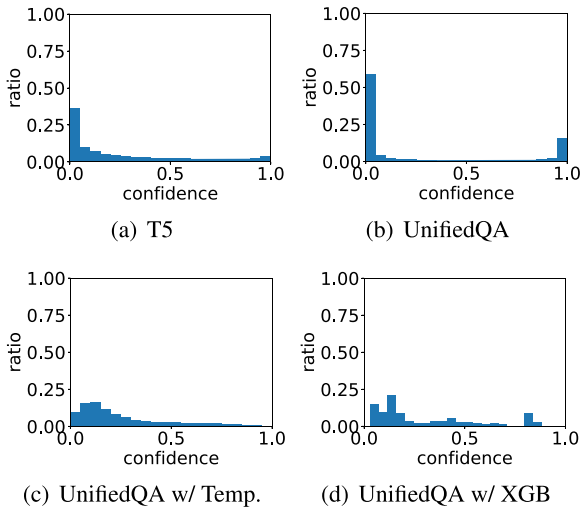
Table 4: Performance of different fine-tuning methods.

5.2 Are LM-based QA Models Well Calibrated?

As shown in Table 4, our baseline models (i.e., T5 and UnifiedQA) are strong, achieving state-of-the-art accuracy on a diverse range of QA datasets. On the MT-test datasets, the UnifiedQA model even outperforms the largest version of GPT-3 with 175 billions parameters (Hendrycks et al., 2020). Despite the impressive performance, these models are not well calibrated, with ECE higher than 0.2 on the MT-test dataset. We found that LMs tend to be over-confident about cases they do not know, as shown in the confidence distribution in the first row of Figure 2 that most predictions have aggressive confidence being close to 0 or 1. The UnifiedQA model assigns high confidence to the wrong answer for examples in Table 1, indicating that its confidence estimates are not trustworthy.

5.3 Can LM-based QA Models be Calibrated?

We calibrate the UnifiedQA model using both fine-tuning-based methods and post-hoc methods



(a) T5 (b) UnifiedQA

(c) UnifiedQA w/ Temp. (d) UnifiedQA w/ XGB

Figure 2: The ratio of predictions with respect to confidence of the T5 model (top-left), the UnifiedQA model (top-right), the UnifiedQA model after temperature-based calibration (bottom-left), and the UnifiedQA model after feature-based calibration (bottom-right) on the MC-test datasets.

Method	MC-test		MT-test		Ext-test	
	ACC	ECE	ACC	ECE	ACC	ECE
Baseline	0.769	0.057	0.431	0.144	0.401	0.114
+ Temp.	0.769	0.049	0.431	**0.075**	0.401	0.107
+ XGB	0.771	0.055	0.431	0.088	0.402	**0.103**
+ Para.	0.767	0.051	0.429	0.122	0.393	0.114
+ Aug.	0.744	0.051	0.432	0.130	0.408	0.110
+ Combo	0.748	**0.044**	0.431	0.079	0.398	0.104

Table 5: Performance of different post-hoc methods using the UnifiedQA model after margin-based fine-tuning or the original UnifiedQA model as the baseline model. ‘‘+Combo’’ denotes the method using both Temp., Para., and Aug.

and show their performance in Table 4 and Table 5 respectively.

Overall, on multi-choice QA datasets (i.e., MC-test and MT-test), both fine-tuning-based methods and post-hoc methods can improve ECE while maintaining accuracy compared to the baseline UnifiedQA model. The best-performing method (i.e., Combo), which combines margin-based fine-tuning, temperature-based scaling, paraphrasing, and input augmentation, improves ECE from 0.095 to 0.044—that is, by over 53%. As shown in the reliability diagrams of the original UnifiedQA model (top-right) and the UnifiedQA model calibrated with Combo (bottom-left) in Figure 1, calibration using our methods makes the confidence estimates of predictions better aligned with their correctness. Comparing those two diagrams, an interesting observation is that our method seems to over-calibrate the LM, making over-estimated bars on the right-hand side of the top-right diagram (bars lower than the diagonal) under-estimated, and vice versa. This is probably caused by the temperature being too aggressive (i.e., too large), making the distribution too flat. Note that the datasets used to learn the temperature (MC-train) and used in evaluation (MC-test) are different, which we hypothesize is the reason why the temperature is too aggressive. We verify this by learning an oracle temperature on the evaluation datasets (MC-test). The learned temperature indeed becomes smaller (1.35 $\rightarrow$ 1.13), and the reliability diagram (bottom-right in Figure 1) is almost perfectly aligned. This demonstrates the challenge of calibrating LMs across different domains.

However, on extractive QA datasets, the improvement brought by different calibration methods is smaller. We hypothesize that this is because the candidate set $\mathcal{I}(X)$ generated by the span-based decoding method for extractive QA are harder to calibrate than the manually curated candidate answers for multiple-choice QA. We compute the average entropy of the confidence of the UnifiedQA model over $\mathcal{I}(X)$ on both extractive QA (Ext-test) and multiple-choice QA datasets (MC-test), and found that Ext-test indeed has much higher entropy compared to MC-test (0.40 vs 0.13), which partially explains the difficulty of calibration on extractive QA datasets.

5.4 Analysis of Individual Calibration Methods

In this section, we discuss each method in detail and analyze why they can improve calibration performance.

Objective Function Matters. The original UnifiedQA model is fine-tuned based on MLE, which maximizes the probability of the gold answer given the question. Both softmax-based and margin-based fine-tuning, which explicitly compare and adjust the probability of candidate answers, can further improve ECE on multiple-choice datasets. We argue that the softmax-based and margin-based objective functions are better suited for questions with potential candidates.

Post-processing Confidence is Effective Universally. Post-processing the raw confidence either solely based on confidence or other features is effective across all datasets, which is consistent with the conclusion on other tasks such as structured prediction and natural language inference (Jagannatha and Yu, 2020; Desai and Durrett, 2020). We demonstrate the histogram of confidence before and after applying temperature-based scaling or feature-based decision trees in Figure 2. LMs tend to be over-confident, with most predictions having either extremely high or low confidence. Both methods can successfully re-scale the confidence to reasonable ranges, thus improving the calibration performance.

Paraphrasing Answers and Input Augmentation can Improve Confidence Estimation. The improvement brought by using paraphrasing is significant on multiple-choice datasets, demonstrating that using diverse expressions can indeed improve confidence estimation. To better understand under what circumstances paraphrasing works, we group candidate answers into two categories: The first group includes candidate answers that become better calibrated using paraphrases; the second group includes candidate answers whose confidence remains the same using paraphrases. We say that a candidate becomes better calibrated if its confidence increases/decreases by 20% if it is a correct or incorrect answer respectively. We found that the average length of questions for better calibrated candidates (187) is much shorter than that of candidates without improvement (320), indicating that paraphrasing is useful mainly for short questions. We also compute the diversity of word usage in paraphrases using the number of unique words divided by the total length of paraphrases. We found that better calibrated candidates have slightly higher diversity (0.35 vs 0.32), which is consistent with our intuition. Retrieval-based augmentation can also improve calibration performance on multiple-choice datasets, which is probably because the retrieved documents can provide extra evidence about the question, making LMs more robust at confidence estimation.

Calibration Methods are Complementary. By combining margin-based fine-tuning, temperature-based scaling, paraphrasing, and input augmentation, we achieve the best ECE on MC-test, demonstrating that these calibration methods are complementary to each other.

Method	BART		GPT-2 large	
	ACC	ECE	ACC	ECE
Original	0.295	0.225	0.272	0.244
+ UnifiedQA	0.662	0.166	0.414	0.243
+ softmax	0.658	0.097	0.434	0.177
+ margin	0.632	0.090	0.450	0.123
+ Temp.	0.632	**0.064**	0.450	**0.067**
+ XGB	0.624	0.090	0.440	0.080
+ Para.	0.624	0.084	0.436	0.104
+ Aug.	0.600	0.089	0.441	0.126
+ Combo	0.591	0.065	0.429	0.069

Table 6: Performance of different LMs on the MC-test dataset. ''Original'' indicates the original language model, and ''+ UnifiedQA'' indicates fine-tuning following the recipe of UnifiedQA.

5.5 Ablation Study

In this section, we perform an ablation study to examine different aspects of LM calibration, including calibration performance of different LMs, across LMs with different sizes, using different numbers of paraphrases, and across datasets with potential domain shift.

Performance of Different LMs. We report the performance of two other LMs in Table 6. Both the BART and GPT-2 models are smaller than T5, thus the overall accuracy and calibration performance are lower than that of T5. Both fine-tuning and post-hoc calibration methods can improve ECE, indicating that our methods are applicable to LMs trained with different datasets and architectures.

Performance of LMs with Different Sizes. We conduct experiments using the largest version (i.e., 11B) of the T5 and UnifiedQA model to analyze how calibration performance varies with respect to the size of the LM in Table 7. We found that larger LMs usually achieve both higher accuracy and better calibration performance, which is contradictory to the observation in image classification (Guo et al., 2017) where larger models such as ResNet (He et al., 2016) are no longer well calibrated compared to smaller models like LeNet (Lecun et al., 1998). Given the fact that the size of both the pre-training corpus and LMs are

Method	MC-test		MT-test	
	ACC	ECE	ACC	ECE
T5	0.359	0.206	0.274	0.235
UnifiedQA	0.816	0.067	0.479	0.175
+ softmax	0.823	0.041	0.488	0.129
+ margin	0.819	0.034	0.485	0.107
+ Temp.	0.819	0.036	0.485	0.098
+ XGB	0.818	0.065	0.486	0.108
+ Para.	0.820	0.035	0.484	0.092
+ Aug.	0.812	**0.031**	0.493	0.090
+ Combo	0.807	0.032	0.494	**0.085**

Table 7: Performance of the 11B LMs.

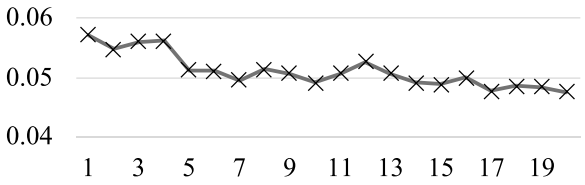


Figure 3: ECE of the UnifiedQA model using different numbers of paraphrases on the MC-test datasets.

extremely larger compared to previous practice, we might have completely different observations with respect to confidence estimation. Unlike ResNet trained on CIFAR-100, the training of LMs is not bottlenecked by the dataset, and larger LMs have a stronger capacity to model text distribution and memorize facts, which leads to better calibration performance overall (Kaplan et al., 2020). Overall, our methods can improve ECE from 0.067 to 0.032 using the 11B UnifiedQA model on the MC-test dataset, and from 0.175 to 0.085 on the MT-test dataset. However, compared to the 3B version, improvement brought by post-hoc calibration methods is smaller, which is probably because the 11B version is better optimized and more knowledgeable.

Performance using Different Numbers of Paraphrases. In Figure 3, we experiment with different numbers of paraphrases using the UnifiedQA model on MC-test datasets. The overall trend is that the more paraphrases we use, the better calibrated the LM, demonstrating that using different variations to express the candidate answer can improve confidence estimation. The improvements using more than 10 paraphrases are subtle, so 5–10 paraphrases may represent a good trade-off between computational cost and performance in practical settings.

Method	MC-train		MC-test	
	ACC	ECE	ACC	ECE
T5	0.334	0.228	0.313	0.231
UnifiedQA	0.727	0.133	0.769	0.095
+ softmax	0.735	0.084	0.767	0.065
+ margin	0.737	0.069	0.769	0.057
+ Temp.	0.737	0.051	0.769	0.049
+ XGB	0.737	0.074	0.771	0.055
+ Para.	0.742	0.053	0.767	0.051
+ Aug.	0.721	0.059	0.744	0.051
+ Combo	0.722	**0.042**	0.748	**0.044**

Table 8: Performance comparison between training and evaluation datasets.

Performance on Training and Evaluation Datasets. As introduced in the experimental section, we perform calibration on the MC-train dataset and evaluate the final performance on the MC-test dataset to study whether our calibration methods can generalize to out-of-domain dataset. We compare the performance on the training dataset and the evaluation dataset in Table 8. We found that on both datasets, each individual method can improve ECE, indicating that our method can generalize to out-of-domain datasets. Note that the improvement on the training dataset ($0.133 \rightarrow 0.042$) is larger than on improvement on the evaluation dataset ($0.095 \rightarrow 0.044$), which is probably caused by the domain shift between the two datasets.

6 Related Work

Calibration Calibration is a well-studied topic in other tasks such as medical diagnosis (Jiang et al., 2012) and image recognition (Guo et al., 2017; Lee et al., 2018). Previous works in NLP have examined calibration in structured prediction problems such as part-of-speech tagging and named entity recognition (Jagannatha and Yu, 2020), natural language understanding tasks such as natural language inference, paraphrase detection, extractive question answering, and text classification (Desai and Durrett, 2020; Kamath et al., 2020; Kong et al., 2020). In contrast, we focus on calibrating LMs themselves by treating them as natural language generators that predict the next words given a particular input.

LM Probing Previous works probe pre-trained LMs with respect to syntactic and semantic properties (Hewitt and Manning, 2019; Tenney et al.,

2019), factual knowledge (Petroni et al., 2019; Poerner et al., 2019; Jiang et al., 2020b), commonsense knowledge (Trinh and Le, 2018; Kocijan et al., 2019), and other properties (Talmor et al., 2019a). These works usually focus on what LMs know, while in this paper we also consider the cases when LMs do not know the answer with confidence.

7 Conclusion

In this paper, we examine the problem of calibration in LMs used for QA tasks. We first note that despite the impressive performance state-of-the-art LM-based QA models tend to be poorly calibrated in their probability estimates. To alleviate this problem, we attempted several methods to either fine-tune the LMs, or adjust the confidence by post-processing raw probabilities, augmenting inputs, or paraphrasing candidate answers. Experimental results demonstrate the effectiveness of these methods. Further analysis reveals the challenges of this problem, shedding light on future work on calibrating LMs.

Some future directions could be developing calibration methods for LMs on a more fine-grained level than simply holistic calibration across the entire dataset. For example, there has been significant interest in how models perform across diverse subsets of the entire training data (Hashimoto et al., 2018) and how they reflect dataset biases (Rudinger et al., 2018), and the interaction of model confidence with these phenomena is of significant interest. It is also interesting to investigate the effect of calibration on users or downstream tasks. For instance, providing users with model confidences can influence downstream decisions (Zhang et al., 2020), and users may want to adjust required confidence thresholds on critical domains (e.g., health, safety, medicine). All of these are interesting paths of inquiry for future research.

Acknowledgments

This work was supported in part by a gift from Bosch research. The authors thank the Google Cloud and TensorFlow Research Cloud for computation credits that aided in the execution of this research.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155.

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press. https://doi.org/10.1609/aaai.v34i05.6239

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4762–4779. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1470

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from BERT. In *Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*. New York, USA. https://doi.org/10.1609/aaai.v34i05.6242

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1870–1879. Association for Computational Linguistics. https://doi.org/10.18653/v1/P17-1171

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, San Francisco, CA, USA, August 13-17, 2016*, pages 785–794. ACM. https://doi.org/10.1145/2939672.2939785

Kenneth Ward Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Second Conference on Applied Natural Language Processing*, pages 136–143, Austin, Texas, USA. Association for Computational Linguistics. https://doi.org/10.3115/974235.974260

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

Pradeep Dasigi, Nelson F. Liu, Ana Marasovic, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5924–5931. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1606

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. *CoRR*, abs/2003.07892.

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

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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 743–753. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1069

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 946–958. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.89

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

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

Tatsunori B. Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. 2018. Fairness without demographics in repeated loss minimization. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1934–1943. PMLR.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition,*

CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 770–778. IEEE Computer Society.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *CoRR*, abs/2009.03300.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4129–4138.

Abhyuday Jagannatha and Hong Yu. 2020. Calibrating structured output predictors for natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2078–2092. Association for Computational Linguistics.

Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. 2012. Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association*, 19(2):263–274.

Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. 2020a. X-factr: Multilingual factual knowledge retrieval from pretrained language models. https://doi.org/10.18653/v1/2020.emnlp-main.479

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020b. How can we know what language models know? *Transactions of the Association for Computational Linguistics (TACL)*. https://doi.org/10.1162/tacl_a_00324

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5684–5696. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single QA system. *CoRR*, abs/2005.00700.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. QASC: A dataset for question answering via sentence composition. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8082–8090. AAAI Press. https://doi.org/10.1609/aaai.v34i05.6319

Vid Kocijan, Ana-Maria Cretu, Oana-Maria Camburu, Yordan Yordanov, and Thomas Lukasiewicz. 2019. A surprisingly robust trick for the winograd schema challenge. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4837–4842. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1478

Lingkai Kong, Haoming Jiang, Yuchen Zhuang, Jie Lyu, Tuo Zhao, and Chao Zhang. 2020. Calibrated language model fine-tuning for in- and out-of-distribution data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1326–1340. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.102

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. RACE: large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017,*

Copenhagen, Denmark, September 9-11, 2017, pages 785–794. Association for Computational Linguistics. https://doi.org/10.18653/v1/D17-1082

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324. https://doi.org/10.1109/5.726791

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. 2018. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020a. Pre-training via paraphrasing. *CoRR*, abs/2006.15020.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020b. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.703

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020c. Retrieval-augmented generation for knowledge-intensive NLP tasks. *CoRR*, abs/2005.11401.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering, MRQA@EMNLP 2019, Hong Kong, China, November 4, 2019*, pages 58–62. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-5808

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2381–2391. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1260

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223, Brussels, Belgium. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 314–319. Association for Computational Linguistics.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. *arXiv preprint arXiv:1803.00047*. https://doi.org/10.18653/v1/W18-6301

Fabio Petroni, Patrick S. H. Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020a. How context affects language models' factual predictions. *CoRR*, abs/2005.04611.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2020b. Kilt: A benchmark for knowledge intensive language tasks. *arXiv:2009.02252*. https://doi.org/10.18653/v1/2021.naacl-main.200

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International*

Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2463–2473. Association for Computational Linguistics. Hong Kong, China. https://doi.org/10.18653/v1/D19-1250

John Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2019. E-bert: Efficient-yet-effective entity embeddings for bert. *CoRR*, abs/1911.03681. https://doi.org/10.18653/v1/2020.findings-emnlp.71

Katyanna Quach. 2020. Researchers made an OpenAI GPT-3 medical chatbot as an experiment. It told a mock patient to kill themselves. *The Register*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 784–789. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-2124

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics. https://doi.org/10.18653/v1/D16-1264

Matthew Richardson, Christopher J. C. Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 193–203. ACL.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *CoRR*, abs/2002.08910.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 8–14. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press. https://doi.org/10.1609/aaai.v34i05.6399

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4462–4472. Association for Computational Linguistics.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2019a. oLMpics - On what

language model pre-training captures. *CoRR*, abs/1912.13283.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019b. Commonsense QA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4149–4158. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4593–4601.

Trieu H. Trinh and Quoc V. Le. 2018. A simple method for commonsense reasoning. *CoRR*, abs/1806.02847.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 191–200. Association for Computational Linguistics. https://doi.org/10.18653/v1/W17-2623

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? Probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5306–5314. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1534

Yunfeng Zhang, Q. Vera Liao, and Rachel K. E. Bellamy. 2020. Effect of confidence and explanation on accuracy and trust calibration in AI-assisted decision making. In *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*, pages 295–305. ACM. https://doi.org/10.1145/3351095.3372852

Pei Zhou, Rahul Khanna, Bill Yuchen Lin, Daniel Ho, Xiang Ren, and Jay Pujara. 2020. Can BERT reason? Logically equivalent probes for evaluating the inference capabilities of language models. *CoRR*, abs/2005.00782.