SELECTIVE PREDICTION UNDER DOMAIN SHIFT
FOR QUESTION ANSWERING

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Amita Kamath

May 2020

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a thesis for the degree of Master of Science.

(Percy Liang) Principal Advisor

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a thesis for the degree of Master of Science.

Christopher Manning

(Christopher Manning) Secondary Advisor

Approved for the Stanford University Committee on Graduate Studies

Abstract

Machine learning is becoming increasingly prevalent in a wide range of real-world applications. To avoid giving incorrect outputs, which could have dire consequences, these models must know when to abstain from answering. Additionally, machine learning models perform poorly when encountering examples from outside their training distribution, which is an inevitable occurrence for deployed systems. This makes errors more likely, and thus abstention more critical. Question answering (QA) models are a prime example: now widely used in web search engines, their performance suffers on out-of-domain inputs. An abstention policy for QA systems that works despite domain shift is thus required. In this work, we propose the setting of selective prediction under domain shift for question answering, in which a QA model is tested on a mixture of in-domain and out-of-domain data, and must answer (i.e., not abstain on) as many questions as possible while maintaining high accuracy. Abstention policies based solely on model output probabilities fare poorly, since models are overconfident on out-of-domain inputs. Instead, we train a calibrator to identify inputs on which the QA model errs, and abstain when it predicts an error is likely. Crucially, the calibrator benefits from observing the model's behavior on out-of-domain data, even if from a different domain than the test data. We conduct extensive experiments combining this method with a SQuAD-trained QA model and evaluating on mixtures of SQuAD and five other QA datasets. Our method answers 56% of questions while maintaining 80% accuracy; in contrast, directly using the model's probabilities only answers 48% at 80% accuracy.

Acknowledgments

First and foremost, I would like to thank my advisor, Percy Liang, for fostering my growth as an academic researcher. His advice, both technical and non-technical, has been invaluable to me. In addition to setting inspiring standards of academic excellence, Percy has been a constant source of support and guidance, which has resulted in a well-rounded research group that is always willing to discuss ideas, offer support, and provide a wide variety of perspectives.

Of these group members, I would first like to thank my mentor and co-author Robin Jia. I approached Robin to ask for insights about an NLP course project, and my discussions with him motivated me to pursue NLP research. His guidance has been instrumental to the work presented in this thesis, and his mentorship has helped me grow as a researcher and computer scientist.

Thank you also to the other members of Percy's group: John Hewitt, Mina Lee, Chris Donahue, Pang Wei Koh, Shiori Sagawa, Erik Jones, Aditi Raghunathan, Ananya Kumar, Steve Mussman, Fereshte Khani, Michael Xie, Fanny Yang, Nelson Liu, Shyamal Buch, Xinkun Nie, Megha Srivastava and Tastu Hashimoto. It has been my privilege to be a member of p-lambda for the past two years.

I am honored to have Chris Manning as a secondary advisor for this thesis. Chris's NLP course kindled my interest in this field, and I have had the pleasure of being a teaching assistant for the course twice. My interactions with Chris have been motivational, in teaching as well as in research contexts.

I would also like to thank the Stanford NLP group, which has been fundamental to my research journey. Chris Manning, Percy Liang, Dan Jurafsky and Chris Potts have fostered a welcoming and supportive group that has provided me with invaluable feedback and encouragement during presentations and informal discussions.

Words cannot express how grateful I am to have had the opportunity to work with and amongst all of you, in Percy's group and the Stanford NLP Group. Every impromptu coffee chat, late night work session, and whiteboard brainstorming of this experience has been a dream come true for me. I sincerely hope to have the privilege to work with you again in the future.

The work presented in this thesis was supported by the Defense Advanced Research Projects Agency (DARPA) ASED program, under FA8650-18-2-7882, which we gratefully acknowledge.

Finally, I would like to express my gratitude to my family and friends. I would not be where I am today without the constant support of my parents, Kusuma and Ajith Kamath, and my sister, Kiran Kamath. Thank you for always believing in me, and encouraging me to believe in myself.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Machine learning has become ubiquitous in a wide range of real-world systems, from predicting the weather (Gneiting and Raftery, 2005; Brocker, 2009) to aiding medical diagnoses (Khan et al., 2001; Hanczar and Dougherty, 2008; Rajpurkar et al., 2017). However, machine learning models have been shown to perform poorly when encountering examples different from those they were trained on, i.e., under domain shift, an inevitable occurrence for deployed systems. In many situations, it is preferable for the model to produce no output, i.e., to abstain, rather than produce an incorrect output. In this chapter, we introduce selective prediction (Section 1.1), which can be used to allow machine learning models to abstain from producing outputs that they are insufficiently confident in. We then discuss the need for selective prediction methods that work despite domain shift (Section 1.2). Finally, we detail the contributions and structure of this thesis (Section 1.3).

1.1 Selective Prediction

Machine learning models are being used for an increasing number of real-world applications, including safety- and business-critical ones such as aiding medical diagnoses and processing insurance claims (Khan et al., 2001; Hanczar and Dougherty, 2008; Avanzi et al., 2020). For these models, knowing when to abstain rather than producing an erroneous output is critical to prevent severe consequences.

For example, consider a machine learning model trained to read a medical report and output whether or not a patient has a particular disease to aid a medical expert in diagnosis. If the model is insufficiently confident in its prediction, we would vastly prefer it to abstain from producing the output and back-off to the medical expert, rather than return an output which could likely be incorrect, potentially biasing them (Uyumazturk et al., 2019).

This motivates selective prediction, wherein a system outputs a prediction based on the probability distribution produced by the machine learning model, as well as a scalar confidence in that prediction. If the confidence is below some user-defined threshold, the system abstains from returning the

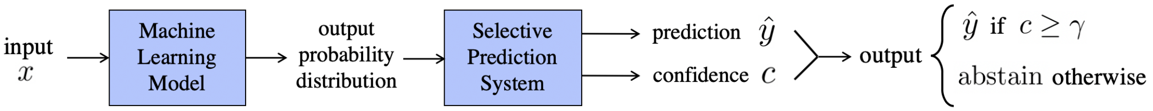


Figure 1.1: Diagram of a model with selective prediction. The system outputs the model prediction $\hat{y}$ if confidence $c \geq \gamma$, the confidence threshold. Else, it abstains.

prediction, as depicted in Figure 1.1.

Selective prediction is of importance to a wide range of research fields, including computational chemistry (Toplak et al., 2014; Zhang and Lee, 2019), finance (El-Yaniv and Pidan, 2011), medical applications (Feng et al., 2019), and Natural Language Processing (NLP) (Dong et al., 2018; Yang et al., 2015; Jurczyk et al., 2016). We discuss these in greater detail in Section 2.1.3.

1.2 Selective Prediction under Domain Shift

While selective prediction is a well-studied field within a single domain, we study it under the practical setting of domain shift. Domain shift is when the examples that a machine learning model is tested on do not always come from the model's training distribution. This is inevitable for systems in production, additionally so because distributions have been shown to change over time (Kramer, 1988). In these settings, model accuracy almost always suffers (Geiger et al., 2019), making selective prediction even more essential.

Going back to our model that predicts whether a patient has a disease given their medical report, assume we give it a report different in some way from those it saw at training time, i.e. a report that is out-of-domain (OOD). For example, the model could have been trained on medical reports written by doctors from one region, and we now give it a report written by a doctor from another.

While ideally, the model could generalize to this OOD input, prior work shows that machine learning models cannot generalize to all OOD inputs given limited training data (Geiger et al., 2019). However, if the confidence estimate of the system is sufficiently low on this OOD sample, the system can abstain if necessary and avoid returning an incorrect output.

Thus, we need a selective prediction method that works despite domain shift, i.e. performs well on both in-domain and OOD inputs, without needing a gold label of whether an input is in-domain or OOD (reflective of test-time conditions).

Selective prediction under domain shift for NLP. Domain shift is a pressing issue in NLP. Models that perform extremely well on benchmark datasets have been shown to err on OOD examples, in ways that suggest they have solved the dataset but not the task itself (Blitzer et al., 2006; Jiang and Zhai, 2007; Jia and Liang, 2017; Yogatama et al., 2019). Selective prediction under domain shift is thus essential to enable NLP systems to abstain on such examples.

1.3 Thesis Overview

In this work we improve the performance of selective prediction under domain shift for the NLP task of Question Answering, assuming only limited access to OOD data separate from the test data. To the best of our knowledge, our work is the first to study selective prediction under domain shift in NLP.

The contributions of this thesis are as follows:

1. We propose a novel setting, selective question answering under domain shift, that captures the practical necessity of knowing when to abstain on test data that differs from the training data.
2. We show that QA models are overconfident on out-of-domain examples relative to in-domain examples, which causes strong baselines using model output probablities to perform poorly in our setting.
3. We show that out-of-domain data, even from a different distribution than the test data, can improve selective prediction under domain shift when used to train a calibrator.

The remainder of the thesis is structured as follows: In Chapter 2, we cover prerequisites necessary to understand selective prediction, domain shift, question answering, as well as related topics such as calibration. In Chapter 3, we discuss selective prediction under domain shift for the task of question answering, proposing a new method to improve performance. In Chapter 4, we conclude the thesis with a discussion on the need for our proposed setting in practical systems, and a description of several interesting directions for future work.

Chapter 2

Background

In this chapter, we discuss selective prediction in Section 2.1, detailing its basic definitions (Section 2.1.1), prior work (Section 2.1.2), and various fields in which it has been studied (Section 2.1.3). We then discuss calibration in Section 2.2 and how it is related to yet distinct from selective prediction. Next, in Section 2.3, we examine prior work in domain shift. Finally, we discuss the NLP task of question answering in Section 2.4, showing why domain shift for this task is an important concern.

2.1 Selective Prediction

2.1.1 Definitions

Given an input x, the selective prediction task is to output $(\hat{y}, c)$ where $\hat{y} \in Y(x)$, the set of answer candidates, and $c \in \mathbb{R}$ denotes the model's confidence. Given a threshold $\gamma \in \mathbb{R}$, the system predicts $\hat{y}$ if $c \geq \gamma$ and abstains otherwise[1], as shown in Figure 1.1.

A standard way to evaluate selective prediction methods is using the risk-coverage curve (El-Yaniv and Wiener, 2010). The *coverage* of the model is the fraction of the test data that the model makes a prediction on at a given γ. The *risk* of the model at that coverage is the fraction of these test inputs for which the prediction made was incorrect. As γ decreases, the coverage tends to increase. However, the risk also increases, as the underlying model does not have perfect accuracy on the test data. Examples of a risk-coverage curve and an optimal risk-coverage curve are given in Figure 2.1.

We evaluate the area under this curve (AUC) as a metric averaging over all γ: the lower the AUC, the better. We also evaluate the maximum possible coverage for a desired risk level, which evaluates at a particular choice of γ corresponding to a specific level of risk tolerance: the higher the coverage at a desired risk level, the better.

[1] where *abstain* $\notin Y(x)$

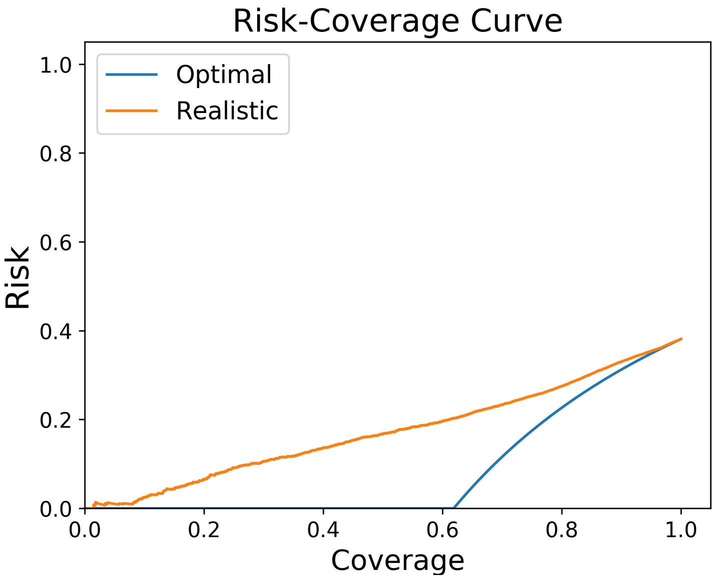


Figure 2.1: Risk-coverage curves in optimal and realistic scenarios. As γ decreases, the coverage increases at the cost of risk. The optimal selective prediction method maintains zero risk until it is forced to answer questions the underlying model gets incorrect at coverage = (1 − total risk), after which the risk increases. In comparison, with the realistic method, the risk increases gradually.

A note on optimality Consider a selective prediction method that assigns a higher confidence to all test data points which an underlying model gets correct, than to the test data points the model gets incorrect. This method is optimal amongst a subset of possible selective predictors that all produce the same $\hat{y}$'s, but different c's. However, its AUC is still greater than 0, as the underlying model's accuracy on the test dataset is not perfect. Say the model error on the full test dataset, i.e. the test error, is R. As γ decreases, the system is forced to predict on data points which the model gets incorrect, and hence risk y increases with coverage x as $y = \dfrac{x - (1 - R)}{x}$. The minimum possible AUC is thus $\displaystyle\int_{1-R}^{1} y \, \mathrm{d}x = R + (1 - R)\log(1 - R)$. In Figure 2.1, the risk on the full dataset, R, is 0.3813. At coverage $= 1 - 0.3813$, the optimal selective prediction method must answer questions the model gets incorrect, reaching the same risk at full coverage as any other selective prediction method. The minimum possible AUC achieved by the optimal selective prediction method here is 8.43.

2.1.2 Prior work

Selective prediction is a long-standing area of focus in machine learning. In this section, we discuss several methods that have been proposed for selective prediction.

Chow (1957) introduces a "rejection channel" to their character recognition system in order to measure the ambiguity of the input, determined by the system probabilities assigned to the various output classes, and rejects inputs accordingly to prevent erroneous outputs. However, this work assumes that the underlying probability distributions are fully known.

El-Yaniv and Wiener (2010) define selective prediction as a risk-coverage tradeoff, proposing the risk-coverage curve discussed in Section 2.1.1. Hendrycks and Gimpel (2017) study MaxProb, a strong baseline for selective prediction. MaxProb is a method that abstains based on the probability assigned by the model to its highest probability label in $Y(x)$.

Ensemble techniques have long been considered for selective prediction. Varshney (2011) study random forests (an ensemble classifier), and abstain from predicting if the average classification score falls within a threshold surrounding the decision boundary. In the era of deep neural networks, Lakshminarayanan et al. (2017) propose that ensembling captures model confidence by aggregating model probabilities over multiple models consistent with the training data.

A significant drawback of ensemble techniques for deep neural networks is that they are costly, requiring the training and querying of multiple ensemble members. Gal and Ghahramani (2016) propose an alternate solution in which an ensemble is instead created during *test time*, by using K different dropout masks in the forward pass through the model to generate K prediction distributions. Statistics over these distributions are then used to calculate a confidence estimate. Although this method performs well on selective prediction and avoids the need to train multiple models, it requires K forward passes of the model, leading to a K-fold increase in runtime. Geifman and El-Yaniv (2017) discuss selective classification techniques as applied to deep neural networks, showing that MaxProb performs as well as or better than test-time dropout on several image classification tasks.

An alternate method that integrates the reject option into the architecture of the model itself is proposed by Geifman and El-Yaniv (2019). The model is trained to optimize both classification and rejection simultaneously. While this is a useful technique to improve selective prediction, it requires changes to the architecture and training of the model on the new objectives. In this work we propose a technique that does not require any changes to the model architecture or training.

2.1.3 Selective prediction in diverse fields

Selective prediction has been studied in a wide variety of fields.

In computational chemistry, selective prediction has been used to determine applicability domain, i.e. the input space in which a model makes reliable predictions about properties of chemicals based on their chemical structure (Toplak et al., 2014; Zhang and Lee, 2019), reducing the need for expensive testing. In finance, selective prediction has been used to obtain confidence estimates for predicting short-term financial trends (El-Yaniv and Pidan, 2011).

In the medical field, selective prediction could be of use for machine learning models in safety-critical applications that necessitate low risk, such as patient diagnosis (Khan et al., 2001; Hanczar and Dougherty, 2008) and ICU prediction tasks (Feng et al., 2019). Confidence estimates of these models could result in safer usage practices, particularly in light of the findings of Uyumazturk et al.

(2019), who show that pathologist diagnoses are significantly biased by a machine learning-based diagnostic assistant, for both correct and incorrect model outputs.

In the field of NLP, traditional systems typically have a natural ability to abstain. SHRDLU recognizes statements that it cannot parse, or that it finds ambiguous (Winograd, 1972). QUALM answers reading comprehension questions by constructing reasoning chains, abstaining if it cannot find any chain that supports an answer (Lehnert, 1977). More recently, selective prediction has been used for semantic parsing (Dong et al., 2018), question answering (Su et al., 2019) and to decide when to answer QuizBowl questions (Rodriguez et al., 2019). Knowing when to abstain is also essential for virtual assistants and answer-triggered systems, when none of the answer candidates for a given question appear correct (Yang et al., 2015; Jurczyk et al., 2016)[2]. In these works, the training and test data come from the same distribution. To the best of our knowledge, our work is the first to study selective prediction under domain shift for NLP.

2.2 Calibration

Calibration is achieved when the prediction probability that a system outputs for an event aligns with the true frequency of that event (Philip, 1982). For example, if a weather forecaster predicts that it will rain for 10 days with probability 0.3, it should rain for approximately 3 of the 10 days. Figure 2.2 shows examples of calibration curves, to be discussed in greater detail in Section 3.5.3.

The importance of model calibration has been shown for clinical settings (Jiang et al., 2012) and meteorological reports (Murphy, 1973; Murphy and Winkler, 1977; DeGroot and Fienberg, 1983; Gneiting and Raftery, 2005; Brocker, 2009). There are several well-established methods to recalibrate model probabilities, including Platt scaling (Platt, 1999) and isotonic regression (Zadrozny and Elkan, 2002). Ovadia et al. (2019) observe increases in calibration error under domain shift.

A key distinction between selective prediction and calibration is that selective prediction metrics generally depend only on *relative* confidences: systems are judged on their ability to rank correct predictions higher than incorrect predictions (El-Yaniv and Wiener, 2010). In contrast, calibration error depends on the absolute confidence scores. Consider an example in which the data points of a test dataset are ordered such that the examples the model gets incorrect all appear before the examples the model gets correct. If confidences are given to each example in a uniformly increasing manner from 0 to 1, the system is poorly calibrated: it is overconfident on nearly all of the examples the model gets incorrect, and underconfident on nearly all of the examples the model gets correct. However, it has optimal selective prediction, because the *relative* ordering of the confidences was correct.

Despite this distinction, we will find it useful to analyze calibration in Section 3.5.3, as being miscalibrated on examples from one domain but not those from another implies poor relative ordering,

[2] Refer to Appendix B for entertaining real-world examples of when models should have abstained!

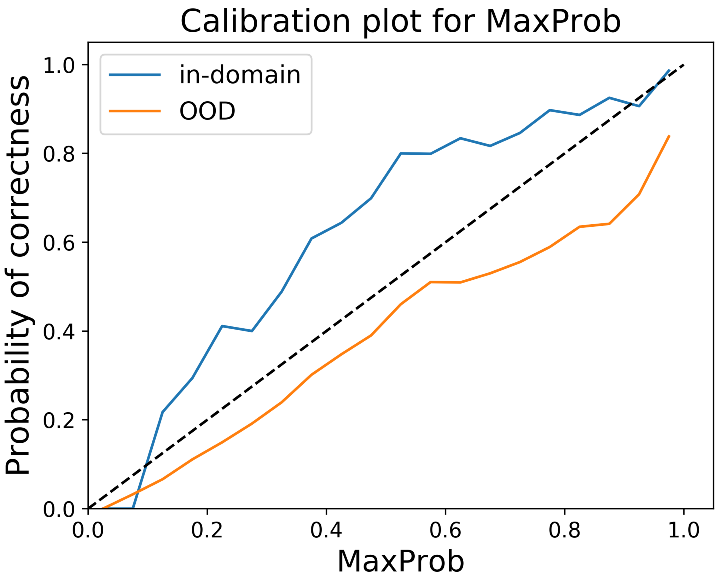


Figure 2.2: Example of calibration curves of a selective prediction method, MaxProb, discussed in Section 3.5.3. The blue line is underconfident, the orange line is overconfident, and the black dashed line represents perfect calibration.

and therefore poor selective prediction. This difference in calibration across domains, in combination with not knowing at test time which domain an example comes from, precludes us from using traditional recalibration techniques to improve selective prediction performance.

2.3 Domain Shift

A common assumption in machine learning literature is that the training and test data are drawn in an i.i.d. manner from the same distribution (Valiant, 1984). However, this does not often hold in practice. In most practical scenarios, it is unrealistic to assume that one has full knowledge of the test distribution at training time — deployed models inevitably face unexpected outputs at test time.

Various types of approaches to tackle domain shift have been proposed. We categorize these approaches into four categories based on two simple criteria:

1. The data the model has access to at train time: whether or not OOD samples were available at train time.
2. The type of data the model is evaluated on at test time: whether the model is being tested on OOD samples alone, or a mixture of in-domain and OOD samples.

These categories are depicted in Figure 2.3, as Categories I, II, III and IV.

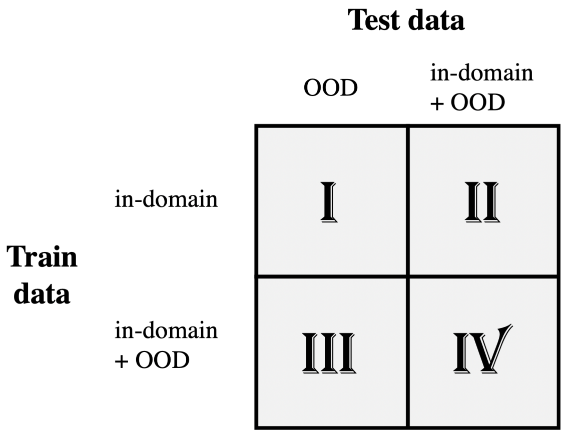


Figure 2.3: The four categories of domain shift discussed in this section, based on the data accessible at training time (Y axis) and type of data the model is evaluated on at test time (X axis).

Category I: trained in-domain, tested OOD. This category is studied in the literature as unsupervised domain adaptation (Quiñonero-Candela et al., 2009), or as zero-shot learning. Here, the goal is to adapt directly from a labeled source domain to an unlabeled target domain. Some methods in this category involve learning domain-invariant representations (Zhao et al., 2019; Shu et al., 2018).

Category II: trained in-domain, tested on mixture. This category evaluates the model on a mixed setting, without having access to any OOD data at training time. This challenging setting is tackled in Hendrycks and Gimpel (2017), in which model probabilities are used to identify OOD examples at test time. Notably, the goal here is outlier detection, and not improving model accuracy on OOD examples.

Category III: trained on mixture, tested OOD. This category is more popularly studied; in which some number of labeled OOD examples are available at training time, and the model must perform well on OOD samples at test time. This is also known as few-shot learning (Fink, 2004; Fei-Fei et al., 2006). In NLP, this setting has been studied for tasks including QA (Talmor and Berant, 2019), language modeling (Vinyals et al., 2016), machine translation (Kaiser et al., 2017), sentiment (Blitzer et al., 2007), and part-of-speech tagging and named entity recognition (Blitzer et al., 2006; Jiang and Zhai, 2007; Daume III, 2007).

Category IV: trained and tested on mixture. This category evaluates the model on a mixed setting, with exposure to OOD data at training time. Importance weighting based methods (Jiayuan et al., 2006; Shimodaira, 2000; Sugiyama et al., 2007) fall under this category, as they assume overlap between the source and target domains. Work tackling goals other than improving model accuracy on the test data include Hendrycks et al. (2019b), which tackles outlier detection, making the looser assumption of having access to OOD data that does not belong to the test dataset. The work outlined in this thesis also makes this looser assumption, and falls under this category, tackling selective prediction — a task for which the mixture setting is especially challenging, as discussed in Section 3.5.3. We detail the specific nature of domain shift we study in Section 3.3.2.

Other types of domain shift. These include concept drift (Kramer, 1988) and gradual domain shift (Bobu et al., 2018; Michael et al., 2018; Markus et al., 2018; Kumar et al., 2020), which make different assumptions about the train and test distributions, primarily with respect to how they change over time, as well as how much of the data accessible is labeled.

2.4 Question Answering

Question answering (QA) is a well-known NLP task that has become popular as a method to evaluate how well machine learning models understand natural language. It is also critical to several industry applications, including search engines (Kwiatkowski et al., 2019) and dialogue systems (Reddy et al., 2018; Choi et al., 2018). We focus on extractive QA, in which a machine learning model is given a passage and a question, and must answer the question by selecting a span from the passage. Several extractive QA datasets are discussed further on in this section.

2.4.1 Extractive Question Answering Datasets

This section discusses some popular extractive QA datasets. These differ from each other significantly in multiple ways: based on where the passages are sourced from, the type of question, the passage and question lengths, as well as the data collection method, which influences the relationship between the passage and question. For all of the below datasets, we consider only answerable questions in this study. The metric we consider is model accuracy, measured based on whether each prediction is an exact match to the corresponding answer(s).[3]

SQuAD. SQuAD (Rajpurkar et al., 2016) sources passages from Wikipedia and questions from crowdworkers who were provided the passage. It was the first large-scale extractive QA dataset. The average question length is 11 tokens, and the average passage length is 137 tokens. An example from SQuAD is shown in Figure 2.4a. SQuAD 2.0 (Rajpurkar et al., 2018) added to SQuAD the concept of "unanswerable questions", which the QA model must recognize as such based on lack of support in the given passage. We discuss the difference between recognizing unanswerable questions and abstaining in Section 3.2.

[3] Some datasets provide multiple annotated answers per question, which acts as a more accurate evaluation metric.

TriviaQA. TriviaQA (Joshi et al., 2017) sources questions from trivia and quiz-league websites, and passages from web snippets returned by a Bing search query. The average question length is 16 tokens, and the average passage length is 784 tokens. An example from TriviaQA is shown in Figure 2.4b.

HotpotQA. In HotpotQA (Yang et al., 2018), each passage consists of two paragraphs from Wikipedia linked by a "bridge entity". The questions are sourced from crowdworkers, who were provided these two paragraphs and requested to write questions requiring multi-hop reasoning to solve. The average question length is 22 tokens, and the average passage length is 232 tokens. An example from HotpotQA is shown in Figure 2.4c.

NewsQA. NewsQA (Trischler et al., 2017) sources passages from CNN news articles and questions from crowdworkers who only see the article headline and summary (although the answers are provided by crowdworkers who can see the full article). The average question length is 8 tokens, and the average passage length is 599 tokens. An example from NewsQA is shown in Figure 2.4d.

Natural Questions. Natural Questions (Kwiatkowski et al., 2019) sources questions from real users' Google search queries. We focus on the "Short Answer" setting, in which the passage is a Wikipedia paragraph containing the answer. Although, like SQuAD, the passages are sourced from Wikipedia, unlike SQuAD, this dataset includes lists and tables. The average question length is 9 tokens, and the average passage length is 153 tokens. An example from Natural Questions is shown in Figure 2.4e.

SearchQA. SearchQA (Dunn et al., 2017) sources questions from the Jeopardy! TV show, which very notably phrases their questions as sentences. The passages consist of web snippets returned by a Google search query. The average question length is 17 tokens, and the average passage length is 749 tokens. An example from SearchQA is shown in Figure 2.4f. Surprisingly, in this example, the question appears verbatim in the passage. This is true for many of the SearchQA examples due to the nature of the data collection.

2.4.2 Prior work

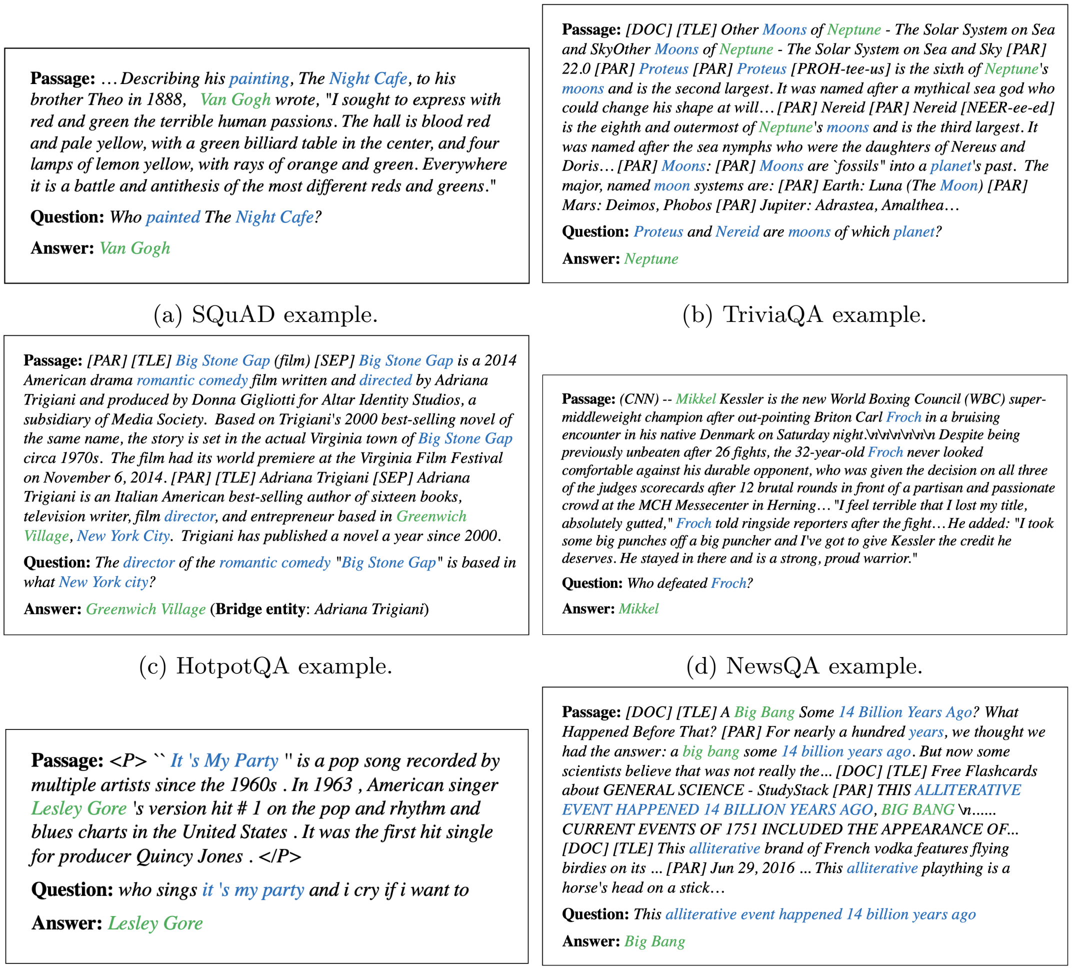
The large number of extractive QA datasets available, as well as the high accuracy models have achieved on each of these, may seem to imply that QA is a solved task. However, recent work shows that QA models that achieve state-of-the-art performance on a particular dataset perform poorly on OOD examples (Fisch et al., 2019; Chen et al., 2017; Jia and Liang, 2017; Talmor and Berant, 2019; Yogatama et al., 2019). Although this may be understandable, given how significantly different these datasets are from each other (as seen in Section 2.4.1), it has increased the focus on making QA systems that can handle domain shift.

Jia and Liang (2017) present adversarial QA examples that fool a model into selecting an incorrect answer span. While training the model on these OOD examples improves performance on them, it does not improve robustness of the model against slightly different types of adversarial inputs — a concerning observation, given that it is unrealistic to expect access to the full test distribution at training time.

In contrast, in the more natural setting of evaluating on different extractive QA datasets, Talmor and Berant (2019) show that training on OOD data may improve the model performance on test data sampled from a separate distribution. However, if the new distribution involves new types of questions or requires different reasoning skills, models trained on multiple domains may still struggle.

Fisch et al. (2019) present a shared task in which the goal is to evaluate how well QA systems can generalize to different datasets. The systems submitted to this task used techniques including data sampling, multi-task learning, adversarial training and ensembling to improve generalization. However, it is made clear that there is significant room for improvement.

Selective prediction under domain shift is thus required to enable the system to abstain on test examples where the QA model may err. We propose our method to achieve this in the next chapter.



Passage: *…Describing his painting, The Night Cafe, to his brother Theo in 1888, Van Gogh wrote, "I sought to express with red and green the terrible human passions. The hall is blood red and pale yellow, with a green billiard table in the center, and four lamps of lemon yellow, with rays of orange and green. Everywhere it is a battle and antithesis of the most different reds and greens."*

Question: *Who painted The Night Cafe?*

Answer: *Van Gogh*

(a) SQuAD example.

Passage: *[DOC] [TLE] Other Moons of Neptune - The Solar System on Sea and SkyOther Moons of Neptune - The Solar System on Sea and Sky [PAR] 22.0 [PAR] Proteus [PAR] Proteus [PROH-tee-us] is the sixth of Neptune's moons and is the second largest. It was named after a mythical sea god who could change his shape at will… [PAR] Nereid [PAR] Nereid [NEER-ee-ed] is the eighth and outermost of Neptune's moons and is the third largest. It was named after the sea nymphs who were the daughters of Nereus and Doris… [PAR] Moons: [PAR] Moons are `fossils" into a planet's past. The major, named moon systems are: [PAR] Earth: Luna (The Moon) [PAR] Mars: Deimos, Phobos [PAR] Jupiter: Adrastea, Amalthea…*

Question: *Proteus and Nereid are moons of which planet?*

Answer: *Neptune*

(b) TriviaQA example.

Passage: *[PAR] [TLE] Big Stone Gap* (film) *[SEP] Big Stone Gap is a 2014 American drama romantic comedy film written and directed by Adriana Trigiani and produced by Donna Gigliotti for Altar Identity Studios, a subsidiary of Media Society. Based on Trigiani's 2000 best-selling novel of the same name, the story is set in the actual Virginia town of Big Stone Gap circa 1970s. The film had its world premiere at the Virginia Film Festival on November 6, 2014. [PAR] [TLE] Adriana Trigiani [SEP] Adriana Trigiani is an Italian American best-selling author of sixteen books, television writer, film director, and entrepreneur based in Greenwich Village, New York City. Trigiani has published a novel a year since 2000.*

Question: *The director of the romantic comedy "Big Stone Gap" is based in what New York city?*

Answer: *Greenwich Village* (**Bridge entity**: *Adriana Trigiani*)

(c) HotpotQA example.

Passage: *(CNN) -- Mikkel Kessler is the new World Boxing Council (WBC) super-middleweight champion after out-pointing Briton Carl Froch in a bruising encounter in his native Denmark on Saturday night.\n\n\n\n\n\n Despite being previously unbeaten after 26 fights, the 32-year-old Froch never looked comfortable against his durable opponent, who was given the decision on all three of the judges scorecards after 12 brutal rounds in front of a partisan and passionate crowd at the MCH Messecenter in Herning… "I feel terrible that I lost my title, absolutely gutted," Froch told ringside reporters after the fight…He added: "I took some big punches off a big puncher and I've got to give Kessler the credit he deserves. He stayed in there and is a strong, proud warrior."*

Question: *Who defeated Froch?*

Answer: *Mikkel*

(d) NewsQA example.

Passage: *<P> `` It 's My Party '' is a pop song recorded by multiple artists since the 1960s . In 1963 , American singer Lesley Gore 's version hit # 1 on the pop and rhythm and blues charts in the United States . It was the first hit single for producer Quincy Jones . </P>*

Question: *who sings it 's my party and i cry if i want to*

Answer: *Lesley Gore*

(e) Natural Questions example.

Passage: *[DOC] [TLE] A Big Bang Some 14 Billion Years Ago? What Happened Before That? [PAR] For nearly a hundred years, we thought we had the answer: a big bang some 14 billion years ago. But now some scientists believe that was not really the... [DOC] [TLE] Free Flashcards about GENERAL SCIENCE - StudyStack [PAR] THIS ALLITERATIVE EVENT HAPPENED 14 BILLION YEARS AGO, BIG BANG \n...... CURRENT EVENTS OF 1751 INCLUDED THE APPEARANCE OF... [DOC] [TLE] This alliterative brand of French vodka features flying birdies on its ... [PAR] Jun 29, 2016 ... This alliterative plaything is a horse's head on a stick…*

Question: *This alliterative event happened 14 billion years ago*

Answer: *Big Bang*

(f) SearchQA example.

Figure 2.4: Examples from different extractive QA datasets. The relevant keywords are shown in blue, and the correct answer in green. Portions of the passage omitted for spatial constraints have been represented using ellipses.

Chapter 3

Selective Question Answering under Domain Shift

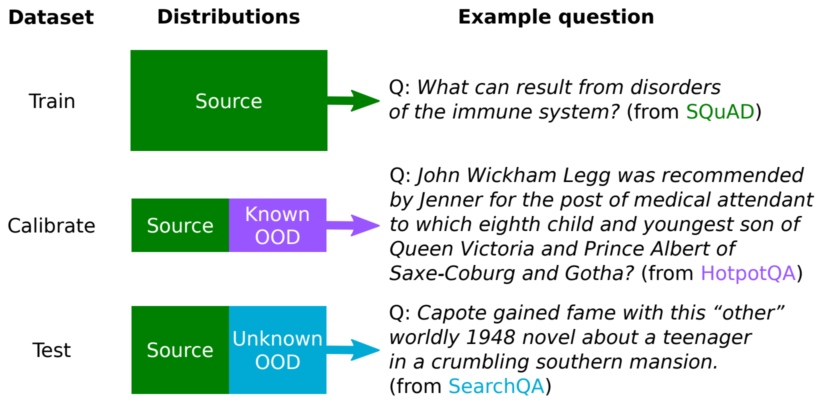
3.1 Introduction

In this chapter, we focus on selective prediction under domain shift for the NLP task of Question Answering (QA), or selective Question Answering under domain shift.[1] As discussed in Section 2.4, QA models have achieved impressive performance when trained and tested on examples from the same dataset, but tend to perform poorly on examples that are out-of-domain (OOD) (Jia and Liang, 2017; Chen et al., 2017; Yogatama et al., 2019; Talmor and Berant, 2019; Fisch et al., 2019).

Deployed QA systems in search engines and personal assistants need to gracefully handle OOD inputs, as users often ask questions that fall outside of the system's training distribution. While the ideal system would correctly answer all OOD questions, such perfection is not attainable given limited training data (Geiger et al., 2019). Instead, we aim for a more achievable yet still challenging goal: models should *abstain* when they are likely to err, thus avoiding showing wrong answers to users. This general goal motivates the setting of selective prediction, as defined in Section 2.1.1.

We propose the setting of **selective question answering under domain shift**, which captures two important aspects of real-world QA: (i) test data often diverges from the training distribution, and (ii) systems must know when to abstain. We train a QA model on data from a *source* distribution, then evaluate selective prediction performance on a dataset that includes samples from both the source distribution and an *unknown OOD* distribution. This mixture simulates the likely scenario in which users only sometimes ask questions that are covered by the training distribution. While the system developer knows nothing about the unknown OOD data, we allow access to a small amount of data from a third *known OOD* distribution (e.g., OOD examples that they can foresee).

[1] We don't refer to the latter using an acronym, for reasons left to the reader to infer.



Q: *What can result from disorders of the immune system?* (from SQuAD)

Q: *John Wickham Legg was recommended by Jenner for the post of medical attendant to which eighth child and youngest son of Queen Victoria and Prince Albert of Saxe-Coburg and Gotha?* (from HotpotQA)

Q: *Capote gained fame with this "other" worldly 1948 novel about a teenager in a crumbling southern mansion.* (from SearchQA)

Figure 3.1: Selective question answering under domain shift with a trained calibrator. First, a QA model is trained only on source data. Then, a calibrator is trained to predict whether the QA model was correct on any given example. The calibrator's training data consists of both previously held-out source data and known OOD data. Finally, the combined selective QA system is tested on a mixture of test data from the source distribution and an unknown OOD distribution.

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

We correct for MaxProb's overconfidence by using known OOD data to train a *calibrator*—a classifier trained to predict whether the original QA model is correct or incorrect on a given example (Platt, 1999; Zadrozny and Elkan, 2002). While prior work in NLP trains a calibrator on in-domain data (Dong et al., 2018), we show this does not generalize to unknown OOD data as well as training on a mixture of in-domain and known OOD data. Figure 3.1 illustrates the problem setup and how the calibrator uses known OOD data. We use a simple random forest calibrator over features derived from the input example and the model's softmax outputs.

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

3.2 Related Goals and Tasks

Calibration. As discussed in Section 2.2, calibration relies on absolute confidence scores, whereas selective prediction relies on the relative ordering of these scores. However, we analyze calibration in Section 3.5.3, as miscalibration on some examples but not others implies poor relative ordering, and therefore poor selective prediction. Ovadia et al. (2019) observe increases in calibration error under domain shift.

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

Identifying unanswerable questions. In SQuAD 2.0, models must recognize when a paragraph does not entail an answer to a question (Rajpurkar et al., 2018). Sentence selection systems must rank passages that answer a question higher than passages that do not (Wang et al., 2007; Yang et al., 2015). In these cases, the goal is to "abstain" when *no* system (or person) could infer an answer to the given question using the given passage. In contrast, in selective prediction, the model should abstain when *it* would give a wrong answer if forced to make a prediction. This is discussed further in Section 3.5.7.

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

3.3 Problem Setup

We formally define the setting of selective prediction under domain shift, starting with some notation for selective prediction in general, in addition to what we covered in Section 2.1.1.

3.3.1 Selective Prediction

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

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

3.3.2 Selective Prediction under Domain Shift

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

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

$$p_{\text{test}} = \alpha p_{\text{source}} + (1 - \alpha) q_{\text{unk}} \tag{3.1}$$

for $\alpha \in (0, 1)$. We choose $\alpha = \frac{1}{2}$, and examine the effect of changing this ratio in Section 3.5.8. Third, q_{known} is a *known OOD distribution*, representing examples not in p_{source} but from which the system developer has a small dataset D_{calib}.

3.3.3 Selective Question Answering

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

3.4 Methods

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

3.4.1 MaxProb

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

$$c_{\text{MaxProb}} = f(\hat{y} \mid x) = \max_{y' \in Y(x)} f(y' \mid x). \tag{3.2}$$

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

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

3.4.2 Test-time Dropout

For neural networks, another standard approach to estimate confidence is to use dropout at test time. Gal and Ghahramani (2016) showed that dropout gives good confidence estimates on OOD data, as discussed briefly in Section 2.1.2.

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

$$c_{\text{DropoutMean}} = \frac{1}{K} \sum_{i=1}^{K} \hat{p}_i(\hat{y}). \tag{3.3}$$

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

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

$$c_{\text{DropoutVar}} = -\text{Var}[\hat{p}_1(\hat{y}), \ldots, \hat{p}_K(\hat{y})]. \tag{3.4}$$

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

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

3.4.3 Training a calibrator

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

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

We experiment with four variants of the calibrator. First, to measure the impact of using known OOD data, we change the calibrator's training data: it can be trained either on data from p_{source} only, or both p_{source} and q_{known} data as described. Second, we consider a modification where instead of the model's probabilities, we use probabilities from the mean ensemble over dropout masks, as described in Section 3.4.2, and also add $c_{\text{DropoutVar}}$ as a feature. As discussed above, dropout features are costly to compute and assume white-box access to the model, but may result in better confidence estimates. Both of these variables can be changed independently, leading to four configurations.

3.5 Experiments and Analysis

3.5.1 Experimental Details

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

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

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

3.5.2 Main results

Training a calibrator with q_{known} outperforms other methods. Table 3.1 compares all methods that do not use test-time dropout. Compared to MaxProb with $f_{\text{src+known}}$, the calibrator has 4.3 points and 6.7 points higher coverage at 80% and 90% accuracy respectively, and 1.1 points lower AUC.[3] This demonstrates that training a calibrator is a better use of known OOD data than training a QA model. The calibrator trained on both p_{source} and q_{known} also outperforms the

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

[3] 95% confidence interval is $[1.01, 1.69]$, using the paired bootstrap test with 1000 bootstrap samples.

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

Table 3.1: Results for methods without test-time dropout. The calibrator with access to q_{known} outperforms all other methods. ↓: lower is better. ↑: higher is better.

calibrator trained on p_{source} alone by 2.4% coverage at 80% accuracy. All methods perform far worse than the optimal selective predictor with the given base model, though achieving this bound may not be realistic.[4]

Test-time dropout improves results but is expensive. Table 3.2 shows results for methods that use test-time dropout, as described in Section 3.4.2. The negative variance of $\hat{p}_i(\hat{y})$'s across dropout masks serves poorly as an estimate of confidence, but the mean performs well. The best performance is attained by the calibrator using dropout features, which has 3.9% higher coverage at 80% accuracy than the calibrator with non-dropout features. Since test-time dropout introduces substantial (i.e., K-fold) runtime overhead, our remaining analyses focus on methods without test-time dropout.

The QA model has lower non-trivial accuracy on OOD data. Next, we motivate our focus on selective prediction, as opposed to outlier detection, by showing that the QA model still gets a non-trivial fraction of OOD examples correct. Table 3.3 shows the (non-selective) exact match scores for all six QA models used in our experiments on all datasets. All models get around 80% accuracy on SQuAD, and around 40% to 50% accuracy on most OOD datasets. Since OOD accuracies are much higher than 0%, abstaining on all OOD examples would be overly conservative.[5] At the same time, since OOD accuracy is worse than in-domain accuracy, a good selective predictor should answer more in-domain examples and fewer OOD examples. Training on 2,000 q_{known} examples does not significantly help the base model extrapolate to other q_{unk} distributions.

[4] As noted in Section 2.1.1, as the QA model has fixed accuracy < 100% on D_{test}, it is impossible to achieve 0% risk at 100% coverage.

[5] In Section A.2, we confirm that an outlier detector does not achieve good selective prediction performance.

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

Table 3.2: Results for methods that use test-time dropout. Here again, the calibrator with access to q_{known} outperforms all other methods.

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

Table 3.3: Exact match accuracy for all six QA models on all six test QA datasets. Training on D_{calib} improves accuracy on data from the same dataset (diagonal), but generally does not improve accuracy on data from q_{unk}.

Results hold across different amounts of known OOD data. As shown in Figure 3.2, across all amounts of known OOD data, using it to train and validate the calibrator (in an 80–20 split) performs better than adding all of it to the QA training data and using MaxProb.

3.5.3 Miscalibration of MaxProb

We now show why MaxProb performs worse in our setting compared to the in-domain setting: it is miscalibrated on out-of-domain examples. Figure 3.3a shows that MaxProb values are generally lower for OOD examples than in-domain examples, following previously reported trends (Hendrycks and Gimpel, 2017; Liang et al., 2018). However, the MaxProb values are still too high out-of-domain. Figure 3.3b shows that MaxProb is not well calibrated: it is underconfident in-domain, and overconfident out-of-domain.[6] For example, for a MaxProb of 0.6, the model is about 80% likely to get the question correct if it came from SQuAD (in-domain), and 45% likely to get the

[6] The in-domain underconfidence is because SQuAD (and some other datasets) provides only one answer at training time, but multiple answers are considered correct at test time. In Appendix A.3, we show that removing multiple answers makes MaxProb well-calibrated in-domain; it stays overconfident out-of-domain.

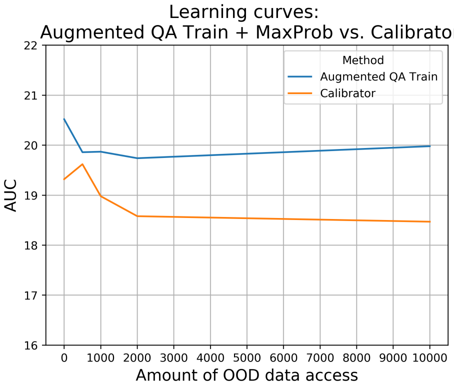


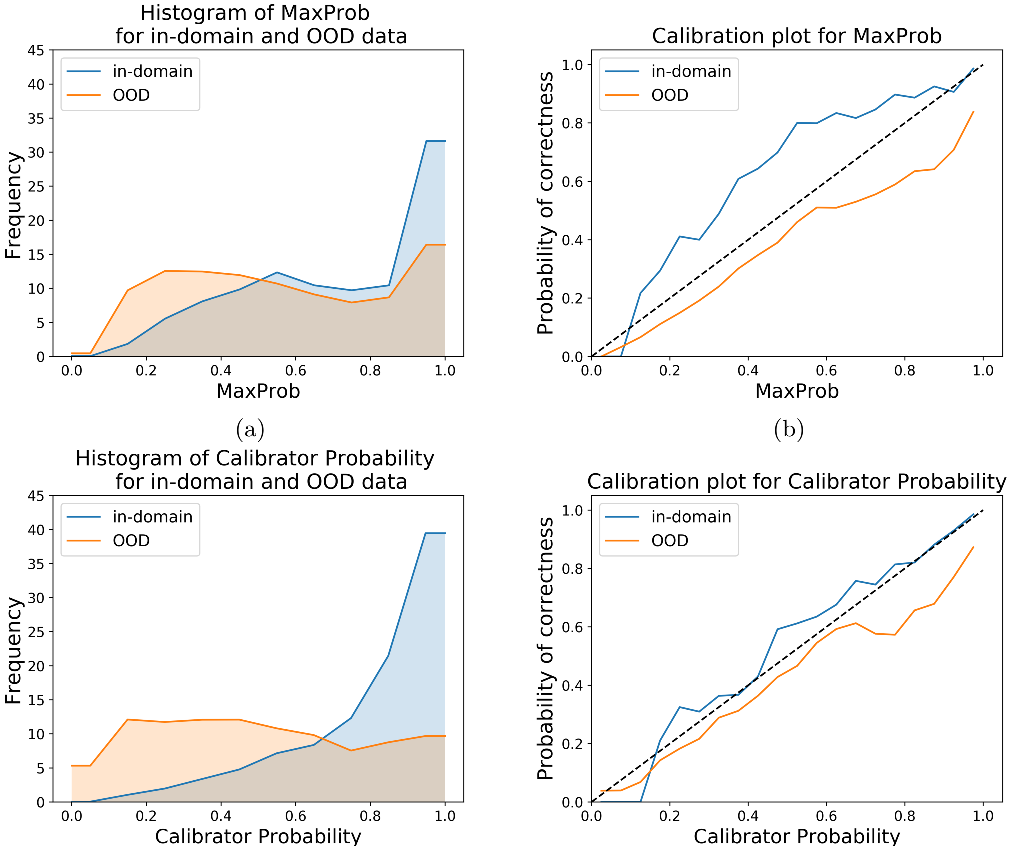
Figure 3.2: Area under the risk-coverage curve as a function of how much data from q_{known} is available. At all points, using data from q_{known} to train the calibrator is more effective than using it for QA model training.

question correct if it was OOD. When in-domain and OOD examples are mixed at test time, MaxProb therefore does not abstain enough on the OOD examples. Figure 3.3d shows that the calibrator is better calibrated, even though it is not trained on any unknown OOD data. In Appendix A.4, we show that the calibrator abstains on more OOD examples than MaxProb.

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

3.5.4 Extrapolation between datasets

We next investigated how choice of q_{known} affects generalization of the calibrator to q_{unk}. Figure 3.4 shows the percentage reduction between MaxProb and optimal AUC achieved by the trained calibrator. The calibrator outperforms MaxProb over all dataset combinations, with larger gains when q_{known} and q_{unk} are similar. For example, samples from TriviaQA help generalization to SearchQA and vice versa; both use web snippets as passages. Samples from NewsQA, the only other non-Wikipedia dataset, are also helpful for both. On the other hand, no other dataset significantly helps generalization to HotpotQA, likely due to HotpotQA's unique focus on multi-hop questions.



(a) (b)

(c) (d)

Figure 3.3: MaxProb is lower on average for OOD data than in-domain data (a), but it is still overconfident on OOD data: when plotting the true probability of correctness vs. MaxProb (b), the OOD curve is below the $y = x$ line, indicating MaxProb overestimates the probability that the prediction is correct. The calibrator assigns lower confidence on OOD data (c) and has a smaller gap between in-domain and OOD curves (d), indicating improved calibration.

3.5.5 Calibrator design decisions

Feature Ablations We determine the importance of each feature of the calibrator by removing each of its features individually, leaving the rest. From Table 3.4, we see that the most important features are the softmax probabilities and the passage length. Intuitively, passage length is meaningful both because longer passages have more answer candidates, and because passage length differs greatly between different domains.

Discarded Features We also experimented with including question length and word overlap between the passage and question as calibrator features. However, these features did not improve the validation performance of the calibrator, as shown in Table 3.5, so we did not include them in our

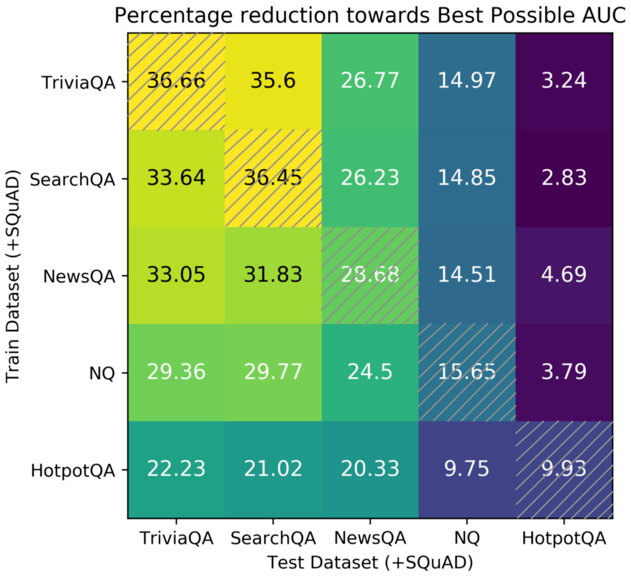


Figure 3.4: Results for different choices of q_{known} (y-axis) and q_{unk} (x-axis). For each pair, we report the percent AUC improvement of the trained calibrator over MaxProb, relative to the total possible improvement. Datasets that use similar passages (e.g., SearchQA and TriviaQA) help each other the most. Main diagonal elements (shaded) assume access to q_{unk} (see Section 3.5.9).

other experiments. We hypothesize that these features may provide misleading information about a given example, e.g., a long question in SQuAD may provide more opportunities for alignment with the paragraph, making it more likely to be answered correctly, but a long question in HotpotQA may contain a conjunction, which is difficult for the SQuAD-trained model to extrapolate to.

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

3.5.6 Error analysis

We examined calibrator errors on two pairs of q_{known} and q_{unk}—one similar pair of datasets and one dissimilar. For each, we sampled 100 errors in which the system confidently gave a wrong answer (overconfident), and 100 errors in which the system abstained but would have gotten the question correct if it had answered (underconfident). These were sampled from the 1000 most overconfident or underconfident errors, respectively.

$q_{\textbf{known}}$ **= NewsQA,** $q_{\textbf{unk}}$ **= TriviaQA.** These two datasets are from different non-Wikipedia sources. 62% of overconfidence errors are due to the model predicting valid alternate answers, or

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

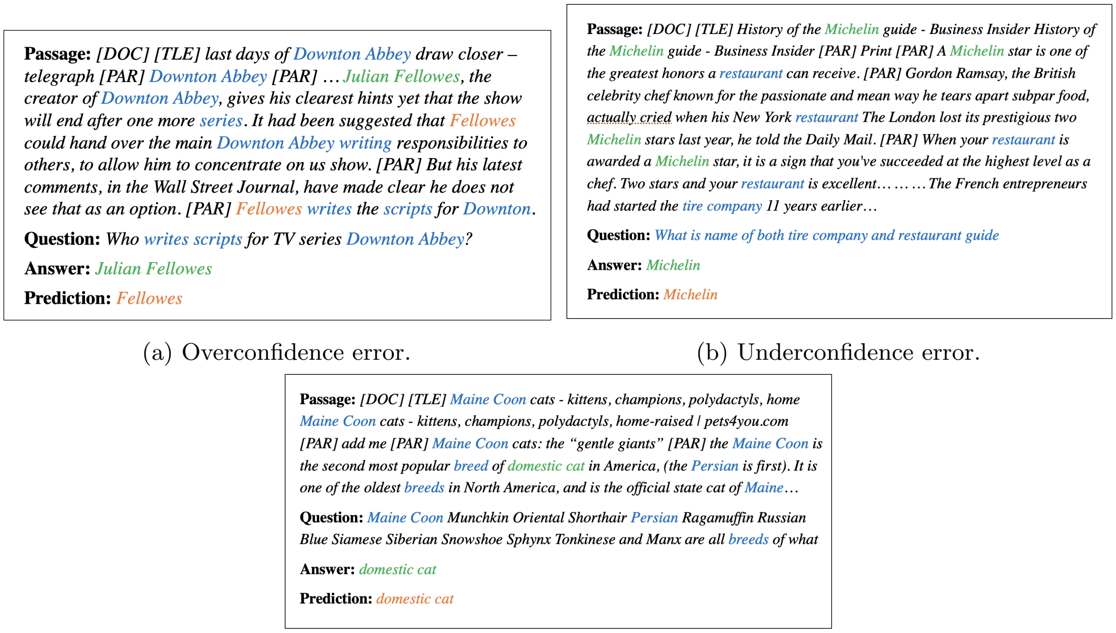
Table 3.4: Performance of the calibrator as each of its features is removed individually, leaving the rest. The base model's softmax probabilities are important features, as is passage length.

	AUC ↓	**Cov @ Acc=80%** ↑	**Cov @ Acc=90%** ↑
Original features	**18.47**	**56.06**	29.42
+ Question length	18.51	55.85	**29.51**
+ Passage-question overlap	18.57	55.51	29.44

Table 3.5: Performance of the calibrator when adding question length and passage-question overlap as features, in order. Although these features improve coverage at 90% accuracy, they do not improve performance when averaged across all thresholds, as shown by the AUC.

span mismatches, as shown in Figure 3.5a—the model predicts a slightly different span than the gold span, and should be considered correct; thus the calibrator was not truly overconfident. This points to the need to improve QA evaluation metrics (Chen et al., 2019). 45% of underconfidence errors are due to the passage requiring coreference resolution over long distances, including with the article title, as shown in Figure 3.5b. Neither SQuAD nor NewsQA passages have coreference chains as long or contain titles, so it is unsurprising that the calibrator struggles on these cases. Another 25% of underconfidence errors were cases in which there was insufficient evidence in the paragraph to answer the question, as shown in Figure 3.5c (as TriviaQA was constructed via distant supervision), so the calibrator was not incorrect to assign low confidence. 16% of all underconfidence errors also included phrases that would not be common in SQuAD and NewsQA, such as using "*said bye bye*" for "*banned.*"

$q_{\textbf{known}}$ = **NewsQA,** $q_{\textbf{unk}}$ = **HotpotQA.** These two datasets are dissimilar from each other in multiple ways. HotpotQA uses short Wikipedia passages and focuses on multi-hop questions; NewsQA has much longer passages from news articles and does not focus on multi-hop questions. 34% of the overconfidence errors are due to valid alternate answers or span mismatches, as shown in Figure 3.6a. On 65% of the underconfidence errors, the correct answer was the only span in the passage that could plausibly answer the question, as shown in Figure 3.6b, suggesting that the model arrived at



Passage: *[DOC] [TLE] last days of Downton Abbey draw closer – telegraph [PAR] Downton Abbey [PAR] … Julian Fellowes, the creator of Downton Abbey, gives his clearest hints yet that the show will end after one more series. It had been suggested that Fellowes could hand over the main Downton Abbey writing responsibilities to others, to allow him to concentrate on us show. [PAR] But his latest comments, in the Wall Street Journal, have made clear he does not see that as an option. [PAR] Fellowes writes the scripts for Downton.*

Question: *Who writes scripts for TV series Downton Abbey?*

Answer: *Julian Fellowes*

Prediction: *Fellowes*

(a) Overconfidence error.

Passage: *[DOC] [TLE] History of the Michelin guide - Business Insider History of the Michelin guide - Business Insider [PAR] Print [PAR] A Michelin star is one of the greatest honors a restaurant can receive. [PAR] Gordon Ramsay, the British celebrity chef known for the passionate and mean way he tears apart subpar food, actually cried when his New York restaurant The London lost its prestigious two Michelin stars last year, he told the Daily Mail. [PAR] When your restaurant is awarded a Michelin star, it is a sign that you've succeeded at the highest level as a chef. Two stars and your restaurant is excellent… … … The French entrepreneurs had started the tire company 11 years earlier…*

Question: *What is name of both tire company and restaurant guide*

Answer: *Michelin*

Prediction: *Michelin*

(b) Underconfidence error.

Passage: *[DOC] [TLE] Maine Coon cats - kittens, champions, polydactyls, home Maine Coon cats - kittens, champions, polydactyls, home-raised | pets4you.com [PAR] add me [PAR] Maine Coon cats: the “gentle giants” [PAR] the Maine Coon is the second most popular breed of domestic cat in America, (the Persian is first). It is one of the oldest breeds in North America, and is the official state cat of Maine…*

Question: *Maine Coon Munchkin Oriental Shorthair Persian Ragamuffin Russian Blue Siamese Siberian Snowshoe Sphynx Tonkinese and Manx are all breeds of what*

Answer: *domestic cat*

Prediction: *domestic cat*

(c) Underconfidence error.

Figure 3.5: Examples of calibrator errors where q_{known} = NewsQA and q_{unk} = TriviaQA. The overconfidence error in (a) is due to span mismatch — the calibrator was not incorrect to assign high confidence to this example. The underconfidence error in (b) is a case where the model must associate “the tire company” with “Michelin” many words prior, without an explicit connection. The underconfidence error in (c) is a case with insufficient evidence in the passage — note that only two of the breeds mentioned in the conjunction question appear in the passage.

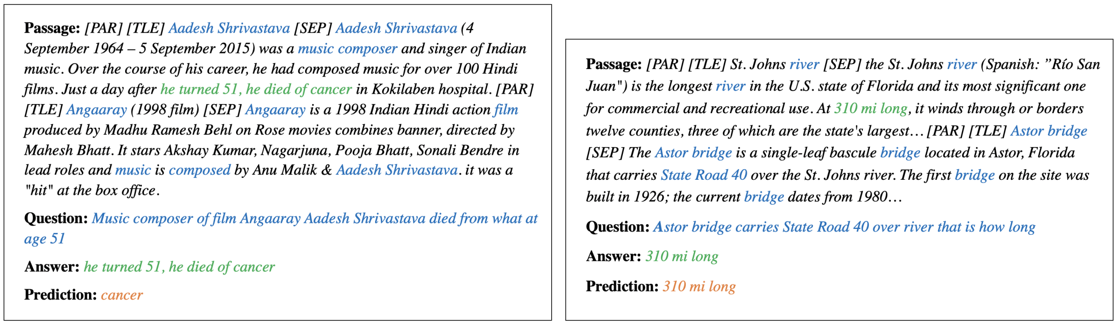
the answer due to artifacts in HotpotQA that facilitate guesswork (Chen and Durrett, 2019; Min et al., 2019). In these situations, the calibrator's lack of confidence is therefore justifiable.

3.5.7 Relationship with Unanswerable Questions

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

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

Selective prediction methods do not identify unanswerable questions. For both MaxProb and our calibrator, we pick a threshold $\gamma' \in \mathbb{R}$ and predict that a question is unanswerable if the



Passage: *[PAR] [TLE] Aadesh Shrivastava [SEP] Aadesh Shrivastava (4 September 1964 – 5 September 2015) was a music composer and singer of Indian music. Over the course of his career, he had composed music for over 100 Hindi films. Just a day after he turned 51, he died of cancer in Kokilaben hospital. [PAR] [TLE] Angaaray (1998 film) [SEP] Angaaray is a 1998 Indian Hindi action film produced by Madhu Ramesh Behl on Rose movies combines banner, directed by Mahesh Bhatt. It stars Akshay Kumar, Nagarjuna, Pooja Bhatt, Sonali Bendre in lead roles and music is composed by Anu Malik & Aadesh Shrivastava. it was a "hit" at the box office.*

Question: *Music composer of film Angaaray Aadesh Shrivastava died from what at age 51*

Answer: *he turned 51, he died of cancer*

Prediction: *cancer*

(a) Overconfidence error.

Passage: *[PAR] [TLE] St. Johns river [SEP] the St. Johns river (Spanish: ”Río San Juan") is the longest river in the U.S. state of Florida and its most significant one for commercial and recreational use. At 310 mi long, it winds through or borders twelve counties, three of which are the state's largest… [PAR] [TLE] Astor bridge [SEP] The Astor bridge is a single-leaf bascule bridge located in Astor, Florida that carries State Road 40 over the St. Johns river. The first bridge on the site was built in 1926; the current bridge dates from 1980…*

Question: *Astor bridge carries State Road 40 over river that is how long*

Answer: *310 mi long*

Prediction: *310 mi long*

(b) Underconfidence error.

Figure 3.6: Examples of calibrator errors where q_{known} = NewsQA and q_{unk} = HotpotQA. The overconfidence error in (a) is due to span mismatch — the calibrator was not incorrect to assign high confidence to this example. The underconfidence error in (b) is a case where the question asks for a length, and the passage contains only one length, making it possible that the model used type-matching to answer the question, rather than multi-hop reasoning through the bridge entity, "St. John's river".

confidence $c < \gamma'$. We choose γ' to maximize SQuAD 2.0 EM score. Both methods perform poorly: the calibrator (averaged over five choices of q_{known}) achieves 54.0 EM, while MaxProb achieves 53.1 EM.[7] These results only weakly outperform the majority baseline of 48.9 EM.

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

3.5.8 Changing ratio of in-domain to OOD

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

Figure 3.7 shows the difference in AUC between the trained calibrator and MaxProb. At both ends of the graph, the difference is close to 0, showing that MaxProb performs well in homogeneous settings. However, when the two data sources are mixed, the calibrator outperforms MaxProb significantly. This further supports our claim that MaxProb performs poorly in mixed settings.

3.5.9 Allowing access to q_{unk}

We note that our findings do not hold in the alternate setting where we have access to samples from q_{unk} (instead of q_{known}). Training the QA model with this OOD data and using MaxProb

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

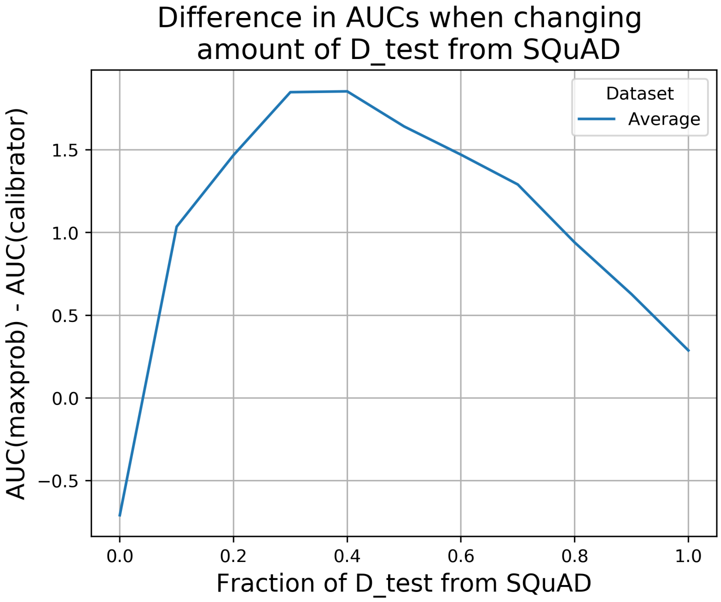


Figure 3.7: Difference in AUC between calibrator and MaxProb, as a function of how much of D_{test} comes from p_{source} (i.e., SQuAD) instead of q_{unk}, averaged over 5 OOD datasets. The calibrator outperforms MaxProb most when D_{test} is a mixture of p_{source} and q_{unk}.

achieves average AUC of 16.35, whereas training a calibrator achieves 17.87; unsurprisingly, training on examples similar to the test data is helpful. We do not focus on this setting, as our goal is to build selective QA models for unknown distributions.

Chapter 4

Conclusion

4.1 Summary

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

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

Our work provides a framework to study how models can recognize when they are well-equipped to provide an answer in both familiar and unfamiliar situations.

4.2 Future Directions

The work in this thesis is primarily based on Kamath et al. (2020). There are several interesting directions in which the ideas of our work can be extended.

Applying to other tasks. The most straightforward extension would be to apply our approach to other tasks. Our framework applies to any model that returns a probability; we could determine whether our findings hold for other tasks where OOD generalization is very important, and abstaining is highly preferable to returning erroneous outputs — such as other high-stakes NLP tasks like drawing inferences from medical reports, or even tasks in non-NLP fields, such as medical image interpretation.

Using probes to predict when models can generalize. In this work, we take only the top 5 probabilities from the final softmax layer of the BERT model to help the calibrator guess whether or not the model can safely generalize to a new example, in addition to a small number of features based on the data itself. In the future, we plan to determine whether there is additional signal that we can gain from the internal representations of the model. Prior work (Tenney et al., 2019; Hewitt and Manning, 2019) has shown that linguistic information such as POS tags, parse trees, coreference chains, etc. can be extracted from BERT representations using probes. We plan to determine whether there is a correlation between the per-example performance of the model representations on these "intermediate tasks" as measured by these probes, and corresponding per-example performance on the "end task" for which BERT was finetuned, e.g. Natural Language Inference. Primarily, we will investigate whether we can reliably predict when our model can safely generalize to an OOD example.

Supervision for the probes will come from a package such as Stanza (Qi et al., 2020). As the Stanza models are trained on a more diverse dataset than our model, we believe they will generalize well to OOD data, giving us a good approximation of probing task error on OOD data. We will determine whether this will enable us to estimate OOD error on the end task without using labeled OOD data.

Leveraging work from other fields to predict when models can generalize. The above approach uses probes to find signal about safe generalization in the model representations. However, there are other ways to draw inferences about new inputs from model representations that have been explored in other fields, and from which we may draw insights. Papernot and McDaniel (2018) compare model representations at test time to those at training time to identify outliers. There is also a significant body of work in the domain of safe exploration for reinforcement learning models (Dalal et al., 2018; Lipton et al., 2016; Richter and Roy, 2017; Achiam and Amodei, 2019; Kahn et al., 2017; Fu et al., 2017; Lee et al., 2019) to identify if a transition leads to an "unsafe" state. It would be an interesting challenge to leverage these techniques to summarize the extremely high-dimensional representation space such that it retains the signal desired, in a way that would capture the model's ability to generalize to new domains, and not only perform outlier detection.

Improving model performance In this work, we do not change the underlying model; we instead attempt to predict when it will err, and abstain accordingly. Going one step further, we could consider making model updates at test time in a way that would make the model more likely to get OOD examples correct, while not sacrificing in-domain performance. Self-training (Chapelle et al., 2006) has been shown to improve OOD accuracy in gradual domain shift (Kumar et al., 2020), and unlabeled OOD data has been leveraged using self-supervision to improve generalization (Gururangan et al., 2020). An additional challenge that may improve performance could be to (perhaps softly) identify domains without requiring labels, which could be used to enable features to have varying weights across domains: for example, this might enable a sentiment classifier to weigh "funny" positively for movie reviews and negatively for restaurant reviews.

4.3 Final thoughts

NLP systems in production inevitably face domain shift: even if inputs are restricted to a single domain, domains change over time (Kramer, 1988). Systems in deployment must therefore be prepared to handle a mixture of familiar and unfamiliar inputs. However, systems trained on finite data cannot generalize to all OOD inputs (Geiger et al., 2019). Thus, recognizing when to abstain is valuable. This makes selective prediction under domain shift vital, particularly in business- and safety-critical applications, where abstaining is highly preferable to producing an incorrect output. Our work provides a framework to study how models can more judiciously abstain in these challenging environments.

As discussed above, there are several ways to extend our work, with the overarching goal of achieving safe generalization. In closing, we emphasize the need to evaluate NLP systems in the practical setting of a mixture of in-domain and OOD inputs, to ensure the challenges we tackle in research via benchmark datasets translate to improved natural language understanding in the wild.

Appendix A

Appendix for Chapter 3

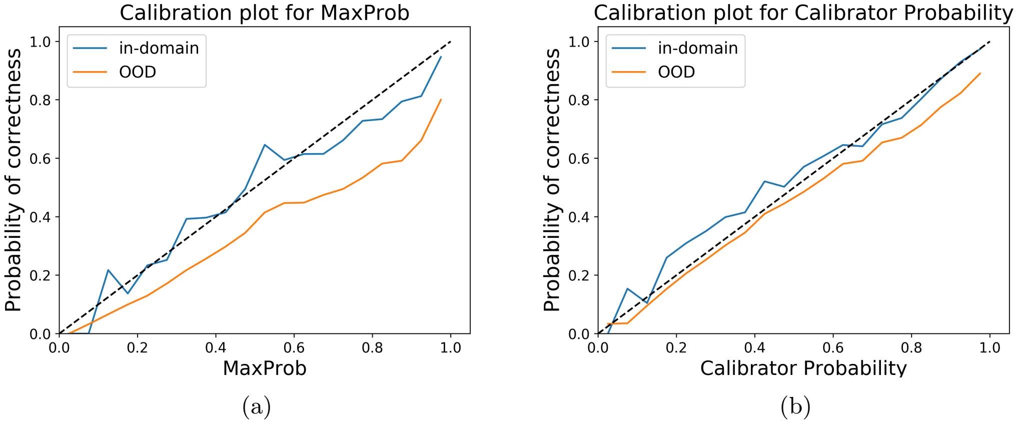
A.1 Dataset Sources

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

As mentioned in Section 3.5.1, only "hard" examples were selected from HotpotQA, as defined by Yang et al. (2018). This was done in order to focus on multi-hop questions. Additionally, this was done to prevent a train/test mismatch, as the validation data for this dataset consists of only "hard" questions. Without this filtering, the accuracy of the model trained on SQuAD 1.1 on the HotpotQA training data is 59.99 and accuracy on the HotpotQA validation data is 44.80, a significant mismatch.

A.2 Outlier Detection for Selective Prediction

In this section, we study whether outlier detection can be used to perform selective prediction. We train an outlier detector to detect whether or not a given input came from the in-domain dataset (i.e., SQuAD) or is out-of-domain, and use its probability of an example being in-domain for selective prediction. The outlier detection model, training data (a mixture of p_{source} and q_{known}), and features are the same as those of the calibrator. We find that this method does poorly, achieving an AUC of 24.23, Coverage at 80% Accuracy of 37.91%, and Coverage at 90% Accuracy of 14.26%. This shows that, as discussed in Section 3.2 and Section 3.5.2, this approach is unable to correctly identify the OOD examples that the QA model would get correct.



(a)

(b)

Figure A.1: When considering only one answer option as correct, MaxProb is well-calibrated in-domain, but is still overconfident out-of-domain (Figure A.1a). Meanwhile, the calibrator is almost perfectly calibrated on both in-domain and out-of-domain examples (Figure A.1b).

A.3 Underconfidence of MaxProb on SQuAD

As noted in Section 3.5.3, MaxProb is underconfident on SQuAD examples due to the additional correct answer options given at test time but not at train time. When the test time evaluation is restricted to allow only one correct answer, we find that MaxProb is well-calibrated on SQuAD examples (Figure A.1a). The calibration of the calibrator improves as well (Figure A.1b). However, we do not retain this restriction for the experiments, as it diverges from standard practice on SQuAD, and EM over multiple spans is a better evaluation metric since there are often multiple answer spans that are equally correct. Evidence of this can also be seen in Section 3.5.6, where a significant portion of the “errors” were span mismatches, which would be largely resolved if other datasets had multiple spans like SQuAD does.

A.4 Accuracy and Coverage per Domain

Table 3.1 in Section 3.5.2 shows the coverage of MaxProb and the calibrator over the mixed dataset D_{test} while maintaining 80% accuracy and 90% accuracy. In Table A.1, we report the fraction of these answered questions that are in-domain or OOD. We also show the accuracy of the QA model on each portion.

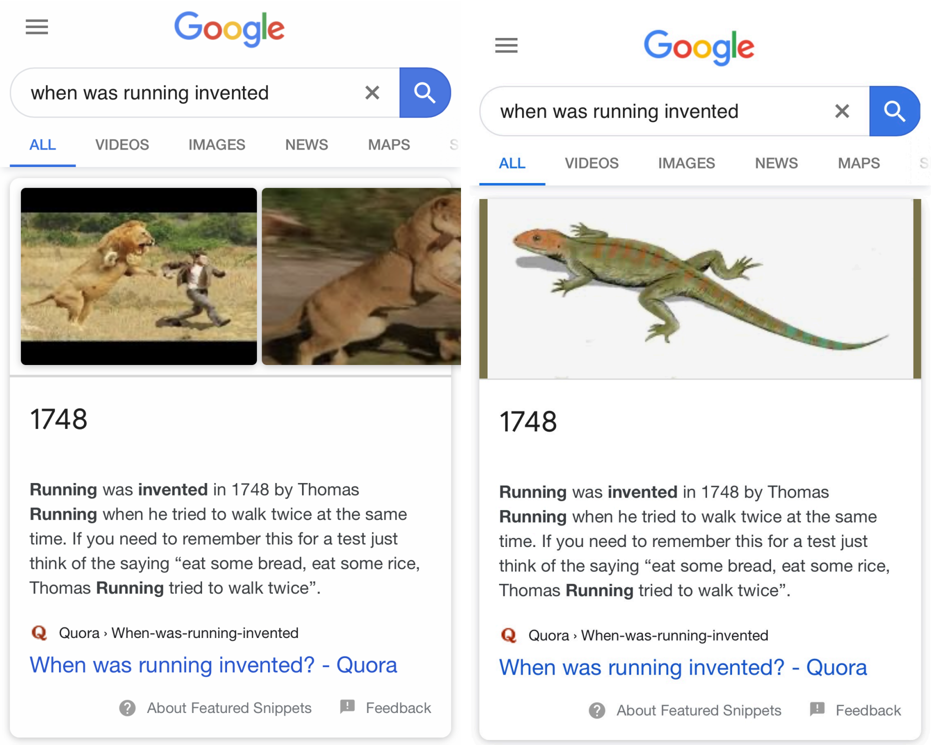
Our analysis in Section 3.5.3 indicated that MaxProb was overconfident on OOD examples, which we expect would make it answer too many OOD questions and too few in-domain questions. Indeed, at 80% accuracy, 62% of the examples MaxProb answers are in-domain, compared to 68% for the calibrator. This demonstrates that the calibrator improves over MaxProb by answering more in-domain questions, which it can do because it is less overconfident on the OOD questions.

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

Table A.1: Per-domain accuracy and coverage values of MaxProb and the calibrator (p_{source} and q_{known}) at 80% and 90% Accuracy on D_{test}.

Appendix B

Real-world examples calling for abstention



when was running invented

1748

Running was **invented** in 1748 by Thomas **Running** when he tried to walk twice at the same time. If you need to remember this for a test just think of the saying “eat some bread, eat some rice, Thomas **Running** tried to walk twice”.

Quora › When-was-running-invented

When was running invented? - Quora

(a)

when was running invented

1748

Running was **invented** in 1748 by Thomas **Running** when he tried to walk twice at the same time. If you need to remember this for a test just think of the saying “eat some bread, eat some rice, Thomas **Running** tried to walk twice”.

Quora › When-was-running-invented

When was running invented? - Quora

(b)

Figure B.1: Sometimes, it really is better to abstain. Google results from (a) 5/10/2020 and (b) 5/18/2020.

Bibliography

J. Achiam and D. Amodei. 2019. Benchmarking safe exploration in deep reinforcement learning. *arXiv*.

B. Avanzi, G. C. Taylor, P. A. Vu, and B. Wong. 2020. A multivariate evolutionary generalised linear model framework with adaptive estimation for claims reserving. *arXiv*.

J. Blitzer, M. Dredze, and F. Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Association for Computational Linguistics (ACL)*.

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

A. Bobu, E. Tzeng, J. Hoffman, and T. Darrell. 2018. Adapting to continuously shifting domains. In *International Conference on Learning Representations Workshop (ICLR)*.

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

J. Brocker. 2009. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society*, 135(643):1512–1519.

O. Chapelle, A. Zien, and B. Scholkopf. 2006. *Semi-Supervised Learning*. MIT Press.

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

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

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

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

E. Choi, H. He, M. Iyyer, M. Yatskar, W. Yih, Y. Choi, P. Liang, and L. Zettlemoyer. 2018. Quac: Question answering in context. In *Empirical Methods in Natural Language Processing (EMNLP)*.

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

G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa. 2018. Safe exploration in continuous action spaces. *arXiv*.

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

M. H. DeGroot and S. E. Fienberg. 1983. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 32:12–22.

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

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

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

R. El-Yaniv and D. Pidan. 2011. Selective prediction of financial trends with hidden markov models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

L. Fei-Fei, R. Fergus, and P. Perona. 2006. One-shot learning of object categories. In *IEEE Trans. Pattern Anal. Mach. Intell.*

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

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

M. Fink. 2004. Object classification from a single example utilizing class relevance metrics. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

J. Fu, J. D. Co-Reyes, and S. Levine. 2017. Ex2: Exploration with exemplar models for deep reinforcement learning. *arXiv*.

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

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

Y. Geifman and R. El-Yaniv. 2019. Selectivenet: A deep neural network with an integrated reject option. In *International Conference on Machine Learning (ICML)*.

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

T. Gneiting and A. E. Raftery. 2005. Weather forecasting with ensemble methods. *Science*, 310.

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

S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Association for Computational Linguistics (ACL)*.

B. Hanczar and E. R. Dougherty. 2008. Classification with reject option in gene expression data. *Bioinformatics*.

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

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

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

J. Hewitt and C. D. Manning. 2019. A structural probe for finding syntax in word representations. In *Association for Computational Linguistics (ACL)*.

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

J. Jiang and C. Zhai. 2007. Instance weighting for domain adaptation in NLP. In *Association for Computational Linguistics (ACL)*.

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

H. Jiayuan, S. A. J., G. Arthur, B. K. M., and S. Bernhard. 2006. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

T. Jurczyk, M. Zhai, and J. D. Choi. 2016. Selqa: A new benchmark for selection-based question answering. *arXiv*.

G. Kahn, A. Villaflor, V. Pong, P. Abbeel, and S. Levine. 2017. Uncertainty-aware reinforcement learning for collision avoidance. *arXiv*.

L. Kaiser, O. Nachum, A. Roy, and S. Bengio. 2017. Learning to remember rare events. In *International Conference on Learning Representations (ICLR)*.

A. Kamath, R. Jia, and P. Liang. 2020. Selective question answering under domain shift. In *Association for Computational Linguistics (ACL)*.

J. Khan, J. S. Wei, M. Ringnér, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer. 2001. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*.

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

A. H. Kramer. 1988. Learning despite distribution shift. In *Connectionist Models Summer School*.

A. Kumar, T. Ma, and P. Liang. 2020. Understanding self-training for gradual domain adaptation. *arXiv preprint arXiv:2002.11361*.

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

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

L. Lee, B. Eysenbach, E. Parisotto, E. Xing, S. Levine, and R. Salakhutdinov. 2019. Efficient exploration via state marginal matching. *arXiv*.

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

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

Z. C. Lipton, K. Azizzadenesheli, A. Kumar, L. Li, J. Gao, and L. Deng. 2016. Combating reinforcement learning's Sisyphean curse with intrinsic fear. *arXiv*.

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

W. Markus, B. Alex, and P. Ingmar. 2018. Incremental adversarial domain adaptation for continually changing environments. In *International Conference on Robotics and Automation (ICRA)*.

G. Michael, E. Dennis, K. B. Mara, B. Peter, and M. Dorit. 2018. Gradual domain adaptation for segmenting whole slide images showing pathological variability. In *Image and Signal Processing*.

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

A. H. Murphy. 1973. A new vector partition of the probability score. *Journal of Applied Meteorology*, 12(4):595–600.

A. H. Murphy and R. L. Winkler. 1977. Reliability of subjective probability forecasts of precipitation and temperature. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 26:41–47.

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

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

N. Papernot and P. McDaniel. 2018. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*.

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

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

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

A. Peñas, 'Alvaro Rodrigo, and F. Verdejo. 2007 2007. *Overview of the Answer Validation Exercise 2007*. Overview of the Answer Validation Exercise.

D. A. Philip. 1982. The well-calibrated Bayesian. *Journal of the American Statistical Association (JASA)*, 77(379):605–610.

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

P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv*.

J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. 2009. *Dataset shift in machine learning*. The MIT Press.

P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng. 2017. CheXnet: Radiologist-level pneumonia detection on chest X-rays with deep learning. *arXiv*.

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

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

S. Reddy, D. Chen, and C. D. Manning. 2018. Coqa: A conversational question answering challenge. *arXiv preprint arXiv:1808.07042*.

C. Richter and N. Roy. 2017. Safe visual navigation via deep learning and novelty detection. In *Robotics: Science and Systems*.

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

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

H. Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244.

R. Shu, H. H. Bui, H. Narui, and S. Ermon. 2018. A DIRT-T approach to unsupervised domain adaptation. In *International Conference on Learning Representations (ICLR)*.

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

L. Su, J. Guo, Y. Fan, Y. Lan, and X. Cheng. 2019. Controlling risk of web question answering. In *ACM Special Interest Group on Information Retreival (SIGIR)*.

M. Sugiyama, M. Krauledat, and K. Muller. 2007. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research (JMLR)*, 8:985–1005.

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

I. Tenney, D. Das, and E. Pavlick. 2019. BERT rediscovers the classical NLP pipeline. *arXiv*.

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

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

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

B. Uyumazturk, A. Kiani, P. Rajpurkar, A. Wang, R. L. Ball, R. Gao, Y. Yu, E. Jones, C. P. Langlotz, B. Martin, G. J. Berry, M. G. Ozawa, F. K. Hazard, R. A. Brown, S. B. Chen, M. Wood, L. S. Allard, L. Ylagan, A. Y. Ng, and J. Shen. 2019. Deep learning for the digital pathologic diagnosis of cholangiocarcinoma and hepatocellular carcinoma: Evaluating the impact of a web-based diagnostic assistant. *arXiv*.

L. Valiant. 1984. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

K. R. Varshney. 2011. A risk bound for ensemble classification with a reject option. In *2011 IEEE Statistical Signal Processing Workshop (SSP)*.

O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. 2016. Matching networks for one shot learning. *arXiv*.

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

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

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

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

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

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

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

Y. Zhang and A. A. Lee. 2019. Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning. *CoRR*, 0.

H. Zhao, R. T. des Combes, K. Zhang, and G. J. Gordon. 2019. On learning invariant representation for domain adaptation. In *International Conference on Machine Learning (ICML)*.