Selective Prediction in Question Answering

Adithya Samavedhi asamavedhi@ucsd.edu
Avni Kothari akothari@ucsd.edu
Hari Vamsi Yadavalli hyadavalli@ucsd.edu
Naigam Shah n7shah@ucsd.edu
Samanvitha Sateesha ssateesha@ucsd.edu

1 What you proposed vs. what you accomplished

- ~~Combine small subsets from datasets to diversify the out of domain (OOD) dataset~~: DONE
- ~~Build a distance based baseline to abstain from questions~~: DONE
- ~~Build the MaxProb baseline which abstains from questions based on the softmax probability~~: DONE
- ~~Build and train a Random Forest calibrator to detect OOD questions~~: DONE.
- ~~Implementd multi-known out of domain with a two known domains apart from the source, but both of those with nearly half as much samples. We performed two experiments with varying sample lengths and inferred their significance.~~: DONE.
- Determine if the BERT based calibrator performs better than the Random Forest calibrator: **We compared the results with BERT embeddings and without BERT embeddings.**
- Implement a Variational AutoEncoder (VAE) to generate a latent vector representation for the given passage which is then passed to the calibrator: **We did not find a pretrained VAE model on our datasets. We did not have enough resources to train and fine tune VAE model. Therefore, could not complete this.**

2 Introduction

The advancement of Natural Language Processing (NLP) and Large Language Models (LLMs) have garnered researchers' attention and touched numerous aspects of one's day to day lives. The latest models have made it possible to accomplish tasks like Text Classification, Question Answering (QA), Dialogue Generation etc. One of the coveted applications of these models is to act as a dialogue system and answer questions. Question Answer models now have the capabilities to address the diverse set of queries thrown at it by the users. QA models are of two types – Abstractive and Generative. Abstractive QA models extract the answer to a question given some context. They assume the context has the answer to the question within it. Generative models have the capabilities to generate a response based on information learned during training. However, in real world scenarios, the QA models often come across questions outside the data it was trained on. QA models are not finely calibrated to abstain from certain questions and end up showing overconfidence. This brings the attention of researchers to make QA models more reliable and robust. Selective Prediction is the process of choosing whether to answer a question or refrain from doing so.

In this project, we aim to expose the overconfidence of QA models and attempt to train models to refrain from answering questions out of their domain knowledge accurately. We first show that softmax probabilities of a model trained on *CrossEntropyLoss* are not a good estimate of confidence. We develop a setting to train our model with some out-of-domain data to calibrate its confidence scores. We evaluate the performance of our model against multiple combinations of QA datasets to show generalisability.

2.1 How is Selective prediction different from Out-of-domain detection?

Selective prediction although seems similar to out-of-domain detection, is a very different concept.

Selective prediction refers to a technique where a machine learning model selectively decides to make a prediction only when it is confident of answering correctly. Selective prediction is done to avoid making incorrect predictions when the model is unsure. In real world scenarios, it is often very costly to make mistakes and selective prediction can be used in such cases where it is better to abstain from making a prediction rather than making a wrong prediction.

Out-of-domain prediction on the other hand refers to the identification of text which is outside the domain knowledge on which the machine learning model was trained on. For example, a model trained to answer questions on wildlife, a question about movies would be considered out-of-domain. Selective prediction differs from out of domain in the sense that machine learning models, QA models in specific are capable of answering questions which are out of their domain. If a model were to be conservative and abstain from answering out of domain questions, it would not be used to its full potential. Instead, selective prediction requires models to answer questions which it maybe confident on. This could mean the model answers some out of domain questions which seem easy and maybe also abstain from answering difficult or ambiguous in-domain questions in order to achieve a more accurate performance and coverage.

3 Related work

QA models have witnessed a significant boom in their research in recent times. The earliest QA models were developed using Information Retrieval (IR) approaches to process and search for relevant answers from a large corpus of text. (Voorhees, 2001) proposed the TREC QA dataset, which comprised of articles from multiple sources including news articles, web pages among others.

Over the years, machine learning found its way into QA. Deep QA developed by IBM Watson won the Jeopardy game against human participants (Ferrucci et al., 2010). The model generated answers to natural language questions using a combination of rule-based heuristics and machine learning techniques. This marked the beginning of the era of advanced language models. Complex language models began to evolve increasing the accuracy and the effectiveness of the QA models. (Seo et al., 2018) developed Bi-Directional Attention Flow (BiDAF) to understand the semantic relationship between the question and the passage and use it to generate answers . (Devlin et al., 2018) also developed BERT, which is a pre-trained transformer-based model which can be used for downstream tasks such as QA .

Researchers began to focus on developing machines with the objective of identifying out-of-domain data in order to make them more reliable and safer to use in real-world situations. One of the first approaches to detect out-of-domain text was proposed by (Otero, 2009), who proposed leveraging the differences of the probability distribution of words to identify out-of-domain text. The task of out-of-domain detection slowly transformed into selective prediction.

Selective Prediction has recently gained traction with the advancement of transformer based models (Vaswani et al., 2017). (Jain and Shenoy, 2022) proposed selective classification as a robust meta learning approach. (Geifman and El-Yaniv, 2017) introduced a technique to trade-off coverage as a solution to selectively predict in deep neural network classifiers. Finally, selective prediction has also found it way into QA models, (Kamath et al., 2020) proposed using a calibrator for selective prediction in QA models under a domain shift.

4 Dataset

We used question answering datasets from Hugging Face which are SQuAD (Rajpurkar et al., 2016), HotpotQA (Yang et al., 2018), NewsQA (Trischler et al., 2017), TriviaQA (Joshi et al., 2017), and Natural Questions (Kwiatkowski et al., 2019). Each sample in these datasets contain a question and passage. These are all extractive question datasets, so all of questions have answers. We chose these datasets because of the wide range of questions and passages they cover. Sample questions and passages can be seen in Table 1.

We consider SQuAD (Rajpurkar et al., 2016) to be in domain and other datasets to be out of domain. We train our model on a combination of both in domain (known) and out of domain questions (unknown). For our experiments, we rotate each out of domain dataset in the train dataset and determine how the model performs in detecting out of domain questions for all the datasets in the test dataset.

We train our model on 1600 known questions and 1600 unknown questions. Our validation set

Dataset	Sample Question	Sample Passage
SQuAD (Rajpurkar et al., 2016)	”In what year was the Grotto of Our Lady of Lourdes at Notre Dame constructed?”	”Because of its Catholic identity, a number of religious buildings stand on campus. The Old College building has become one of...”.
TriviaQA (Joshi et al., 2017)	”In 2002, Chechen terrorists took more than 700 people as hostages in Moscow, what type of building were they in?”	”Russian troops invaded to oust Dudayev in December 1994, setting off 13-month war that killed up...”
HotpotQA (Yang et al., 2018)	”Which film director is older, Charles Martin Smith or Yakov Protazanov?”	”Man from the Restaurant is a 1927 Soviet drama film directed by Yakov Protazanov based on the story by Ivan Shmelyov...”
NewsQA (Trischler et al., 2017)	”What did Contador take from Andy Schleck?”	”Defending champion Alberto Contador has issued an apology...”
Natural Questions (Kwiatkowski et al., 2019)	”who wrote the music to the music man”	”Robert Meredith Willson (May 18 , 1902 – June 15 , 1984) was an American flautist , composer , musical arranger , bandleader and playwright...”

Table 1: A question and context from each dataset. SQuAD is our in domain dataset and the others are out of domain datasets. It can be seen from this table the questions and contexts cover a wide range of topics allowing the model to see many different types of questions and learn a pattern of which questions it should abstain from.

contains 400 known questions and 400 unknown questions and our test set contains 4000 known questions and 1600 unknown questions.

Table 2 refers to some of the properties of each dataset.

4.1 Data preprocessing

We first combine subsets of each dataset and parse the sample to only contain the question and context. We then tokenize the text by combining the question and the context. The tokenizer we use is a pretrained BERT based model from Hugging Face. When tokenizing the text we truncate the length to 512 tokens. The advantage with truncation is this helps with computational efficiency especially since we are using limited resources. The disadvantage with this approach is that we lose information.

Since we are only using samples of each dataset the questions and passages can greatly vary. Not only on topic but also on length. The model may not be able to learn a pattern with such limited data. We are hoping the calibrator can help with determining in and out of domain questions.

5 Baselines

5.0.1 Distance-based Baseline

Our initial baseline is going to convert the questions in the training data and the question the user inputted into word embeddings. We will then compute the cosine similarity between all inputs in the training data and the user input. If the user input is ϵ away from all the points in the training data then the model will abstain from answering the question. We will have to try different values of ϵ once we have built the model to see if one performs better than the other.

$$cos(\theta) = \frac{X * b^T}{||X||^2||b||^2} \quad (1)$$

Here, X represents a matrix of all word embeddings in the training dataset and b represents a vector of the word embeddings for the user input. If the cosine similarity is 1 then the vectors are exactly the same, and if the cosine similarity is -1 then the vectors are opposite. Initially we might try values $-1 \leq \epsilon \leq .2$

Dataset	#Instances	Avg Passage Length (words)	Avg Question Length (words)
SQuAD	87599	120	10
TriviaQA	15368	222	13
HotpotQA	90447	971	18
NewsQA	74160	496	7
Natural Questions	104071	152	9

Table 2: Data and Numerical Analysis of Datasets

5.0.2 MaxProb Baseline

The Baseline approach is inspired and adapted from the authors of the selective prediction (Kamath et al., 2020). The authors propose to use MaxProb as a baseline to estimate model's confidence. MaxProb directly uses the probability assigned by the base model to the predicted outcome as an estimate of confidence. MaxProb can be a effective choice for the baseline because of it's wide applications and it's effectiveness in distinguishing the samples the model predicts correctly from those it fails to predict correctly from the in-domain data. The equation for computing MaxProb is

$$c_{MaxProb} = f(\hat{y}|x) = \max_{y_0 \in Y(x)} f(y_0|x)$$

Here, $\hat{y}$ denotes the predicted outcome against the baseline model.

6 Approaches

We ran numerous experiments for the given dataset and compared and analysed our results for different approaches. The following are a set of approaches we implemented.

6.1 Random Forests calibrator

(Kamath et al., 2020) propose to use an auxiliary model, calibrator for OOD detection. The authors proposed training a calibrator in addition to the QA model. The calibrator is trained on a combination of in-domain and out-of-domain questions. The calibrator is tuned to detect out of domain samples. The softmax scores of the calibrator can be used as an estimate to determine whether the QA model shall abstain or answer the question. In this approach, we plan to tune a Random Forests calibrator based on selected characteristic features of the question. Features we considered in this approach are: the combined length of question and context, length of predicted answer and top 5 softmax probabilities from the QA model output that denote the start index of the answer and the top 5 softmax probabilities from the QA model output that denote the end index of the answer. So, totally we have considered 12 features. We trained the calibrator such that it assigned lower scores to out-of-domain samples it wasn't confident to answer and help us identify the questions the model should abstain from. The calibrator will assign higher confidence scores to those questions that are in it's domain knowledge and capable of answering. In this way, it will assist in the decision making process.

6.2 BERT embeddings as a feature vector

BERT architecture encompasses CLS embeddings that capture the representations of the input sequence. We take the CLS embeddings from the last hidden state and use it as a feature vector for the RF model. The input sequence is both context and question. As we are generating the embeddings for context as well as question, we do not consider the length of context, question in this scenario. The feature vector consists of 768 embeddings as well as the 10 softmax probabilities as mentioned in 6.1. We totally have 778 features in this appproach and train our RF calibrator on these faetures.

We observed that for a few cases, the model performed better for feature vector without BERT embeddings as compared to the feature vector with BERT embeddings. We had not expected this as BERT embeddings provided the context of the entire sequence. Therefore, we theorized that this might be due to the reason that the BERT embeddings were of high dimenions and contained more information than the model could capture. Therefore, we wanted to experiment with a few dimensionality reduction techniques such as Principal Component Analysis (PCA). We reduced the dimensionality of BERT embeddings from 768 to 50. We observed that the results from this experiment were more consistent compared to previous one without dimensionality reduction where we observed more anamolies and a strange dip in a few results. A more detailed results can be found in

6.3 Multi known OOD Approach

The current approach aims to address the limitation of using a small sample size from a single known Out-of-Distribution (OOD) dataset. Instead, we incorporate multiple known OOD datasets, utilizing small samples from each of them. Through various permutations and combinations of these known domains, we explore different combinations to enhance model performance. Additionally, we adjust the limit on the global selection of small samples from all known OOD datasets to be at most 20%, effectively including the entire Source Dataset. However, the combined size of the multi-OOD datasets will not exceed 20% of the actual Source Dataset.

Let's denote the original domain as p_{source}, and the known OOD as $q_{known_1}, q_{known_2}, \cdots, q_{known_n}$, representing n known OOD samples. We also have an unknown OOD, $q_{unknown}$. We curate the data in the format shown below:

$$p_{source} + q_{known_1} + q_{known_2} | q_{unknown}$$

The $p_{source} + q_{known_1} + q_{known_2}$ resemble our training data. Here, p_{source} is fixed to SQuAD, and $q_{known_1}, q_{known_2}, q_{unknown}$ taken in all possible combinations of NewsQA, TriviaQA, HotpotQA and NaturalQuestions. Our testing data is a mix of p_{source}, comprising 50% of $q_{unknown}$ and 50% of the test data from p_{source}.

Intuition behind these approaches is that, we are forcing the model to attempt to learn from few examples from different known OODs, thus there is more for the model to learn from such a setting. We perform two sets of experiments to perform a comparative study between choosing different sample sizes for the source and known OODs.

For our setting one, we took 1600 samples from SQuAD as our p_{source}, 800 samples from our chosen q_{known_1}, and 800 samples from our chosen q_{known_2}, and we test on the 4000 samples from p_{source}, and $q_{unknown}$, this is our setting one. For our setting two, we took 800 samples from SQuAD as our p_{source}, 800 samples from our chosen q_{known_1}, and 800 samples from our chosen q_{known_2}, and we test on the 4000 samples from p_{source}, and $q_{unknown}$.

The results for the setting one are in Table 3, and the results for the setting two are in the appendix section. Based on our comparative study, we notice that when the Multi-OOD BERT+RF model is exposed with half the size of the first known domain samples of that of the BERT+RF model, and the other half from the second domain, we note that the model is more robust and resilient in terms of dealing with out of domain data.

We observed something very interesting here, like when using the BERT+RF model with p_{source} = SQuAD, and q_{known} = NewsQA and $q_{unknown}$ = HotpotQA, we get the Cov@80%Acc as 0.525 and for p_{source} = SQuAD, and q_{known} = NewsQA and $q_{unknown}$ = "natural", we get the Cov@80%Acc as 0.215. Similarly, when p_{source} = SQuAD, and q_{known} = TriviaQA and $q_{unknown}$ = HotpotQA, we get the Cov@80%Acc as 0.5153 and for p_{source} = SQuAD, and q_{known} = TriviaQA and $q_{unknown}$ = NaturalQuestions, we get the Cov@80%Acc as 0.176. However, with the Multi-OOD BERT+RF model with p_{source} = SQuAD, q_{known_1} = TriviaQA, q_{known_2} = NewsQA and $q_{unknown}$ = HotpotQA, we get the Cov@80%Acc as 0.526, also p_{source} = SQuAD, q_{known_1} = TriviaQA, q_{known_2} = NewsQA and $q_{unknown}$ = NaturalQuestions, we get the Cov@80%Acc as 0.2283.

This proves that our initial intuition about the model trying to learn better when given different distributions of data even when the sample size of the each distribution are 50% less but from two different distribution. We can see its significance specifically when the model is faced with complete OOD data which it has never seen before which is better than that of the BERT-RF model.

7 Error analysis

The following sections consist of in-depth analysis of how the project unfolded. We show the comparison of various models using the coverage@80% accuracy in the Table

7.1 Why Selective Prediction is needed?

A QA model as is, will be unable to differentiate between an *easy* question and a *difficult* question. The need for ML models to understand their capabilities is of utmost importance and we would like ML models to only respond when confident. However, Maximum Softmax Probability, referring to the highest softmax probability of the output vector is intuitively a good estimate of the confidence

Model	Train Data \Test Data	news	trivia	hotpot	natural
Distance-Baseline	squad	0.578	0.375	0.373	0.021
MaxProb - Baseline	squad	0.657	0.007	0.203	0.359
RF	squad + news	0.665	0.463	0.515	0.525
	squad + trivia	0.632	0.526	0.522	0.588
	squad + hotpot	0.590	0.456	0.540	0.669
	squad + natural	0.650	0.370	0.463	0.593
BERT+RF	squad + news	0.594	0.494	0.526	0.215
	squad + trivia	0.552	0.518	0.515	0.176
	squad + hotpot	0.604	0.467	0.527	0.585
	squad + natural	0.590	0.394	0.520	0.551
BERT(PCA)+RF	squad + news	0.645	0.489	0.519	0.541
	squad + trivia	0.537	0.507	0.497	0.403
	squad + hotpot	0.571	0.464	0.526	0.476
	squad + natural	0.607	0.378	0.516	0.561
Multi-OOD BERT+RF	squad + news + trivia	0.517	0.509	0.526	0.228
	squad + news + hotpot	0.561	0.497	0.523	0.302
	squad + news + natural	0.551	0.443	0.527	0.540

Table 3: Cov@80%Acc Comparison of all approaches

and one would expect that *easy* questions should have a higher confidence score than *difficult* ones. Empirically, it is shown that Maxprob is not a realistic estimate of confidence and ML models often suffer from overconfidence wherein they respond to a question outside of their capabilities but with high confidence. QA models as is, are quite overconfident on all questions and are unable to differentiate between what they can answer and what they cant.

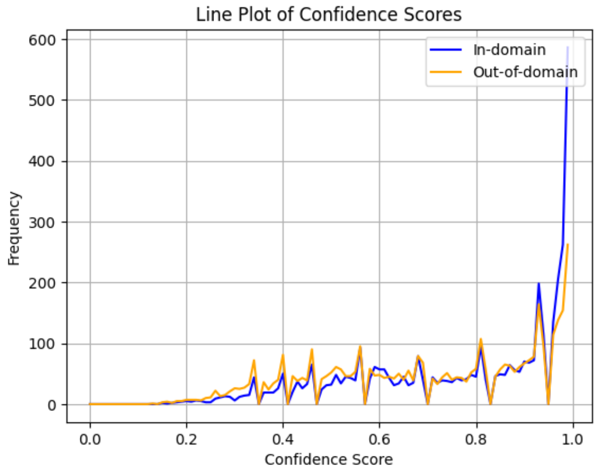

Figure 1: MaxProb Confidence Scores

The Figure 1 shows how the MaxProb model is equally confident for in-domain and out-of-domain samples, in this setting the indomain dataset is SQUAD and $q_{unknown}$ is *trivia*. Hence the model ends up answers to almost all questions in a confident manner. We want to improve this, such that the model answers with lower confidence on questions it is unaware of and selectively answers questions. The next section talks about how having a callibrator helps make our QA models into selective predictors.

7.2 How does a callibrator help QA models selectively predict?

In the previous section, we noticed how QA models as is are completely unaware of their capabilities and respond in a confident manner to all questions (correct or wrong). We want to avoid this case. In this section we introduce a callibrator in the form of a *Random Forests* model. The Random Forests model is trained on the following features: *top 5 start softmax probabilities, top 5 end softmax probabilities, passage length, answer length*. We also experiment with variants of the RF: *BERT+RF, BERT(PCA) +RF*. These two variants use a similar architecture as the RF callibrator, but instead use BERT embeddings of the Question and Context. The BERT(PCA)+RF method uses BERT embeddings reduced in dimensionality by PCA and fed into the RF callibrator.

The Table 4 refers to the accuracy of the QA model when it answers all questions. Evidently, we see that the accuracy is not very great when the model answers all questions, Hence requiring the need for callibrator.

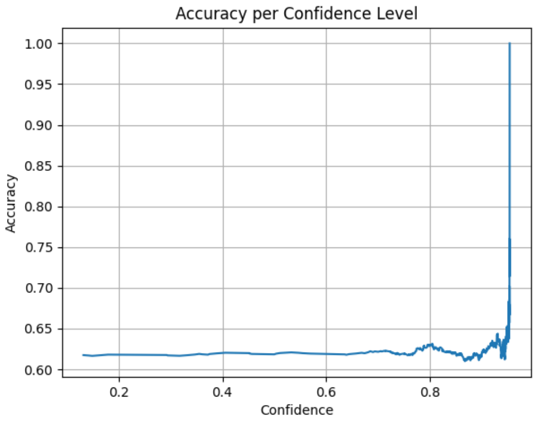

Figure 2: Accuracy vs Confidence

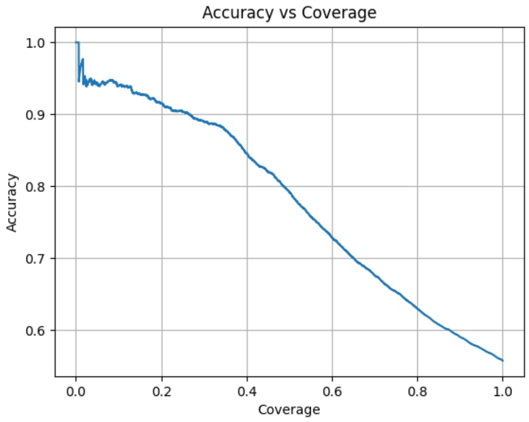

Figure 3: Accuracy vs Coverage

We take a setting where p_{source} is *SQuAD*, q_{known} is *newsqa* and $q_{unknown}$ is *trivia*. The figures 2 and 3 showcase how with a callibrator the coverage and the confidence scores vary with accuracy. In the Accuracy vs Confidence graph, we see that the accuracy remains at around 60% while the confidence is low, that is, the accuracy when answering non confident questions is 60%. As the confidence increases, we expect the model to answer questions which it is able to answer, hence the increased axcuracy. The accuracy vs coverage showcases the same, as the coverage inreases the accuracy decreases since the model choses to answer the easy questions first.

7.3 Case Study:

In this case study we focus on the dataset (p_src, q_known, q_unknown) combination of (SQuAD, News, Trivia). We first evaluate the performance of different models in selective prediction using the coverage at 80% accuracy metric. The results provided valuable insights into the effectiveness of each approach. The MaxProb Baseline achieved a relatively low coverage of 0.007, while the Distance Baseline showed significant improvement with a coverage of 0.375. Utilizing a calibrator RF further enhanced the results, increasing the coverage to 0.463. Introducing 768 BERT embeddings along with the RF calibrator yielded even better performance, reaching a coverage of 0.494. Surprisingly, when applying the top-50 PCA components of BERT embeddings with the RF calibrator, the coverage slightly decreased to 0.489.

As can be seen in table 4, without using a calibrator the model achieves an accuracy of 55.79%, but using BERT+RF approach, we restrain our model from answering some of the questions it's not confident at answering. As shown in figure 3, as we increasingly restrain the model, the accuracy of the QA model increases.

During our experiments with different calibrators RF, Bert+RF, and Bert(PCA)+RF, we made an interesting observation when considering the dataset combination of (squad, news, trivia). The Bert+RF model demonstrated the highest coverage at 80% accuracy initially, but its performance plateaued beyond this threshold, showing limited improvement with reduced coverage. In contrast, both the RF and Bert(PCA)+RF calibrators exhibited consistent improvement even after surpassing the 80% accuracy mark. Notably, the Bert(PCA)+RF model displayed an upward parabolic trend after reaching the 80% accuracy point, indicating its ability to assign higher confidence scores to a greater number of data points that the model can accurately answer compared to the RF calibrator. These trends are displayed in the Figure 4. The observed trends can be attributed to the nature of the RF calibrator and the dimensionality of the Bert embeddings. With a 768-dimensional Bert embeddings vector, the RF calibrator may struggle to capture and retain comprehensive information about the data distribution, potentially requiring additional training. However, by employing a reduced Bert embeddings vector obtained through the top-50 PCA components, the calibrator model is able to effectively extract the essential data information, leading to improved performance.

p_src\q_unknown	news	trivia	hotpot	natural
squad	0.642	0.557875	0.451	0.635625

Table 4: QA model Accuracy without Calibrator

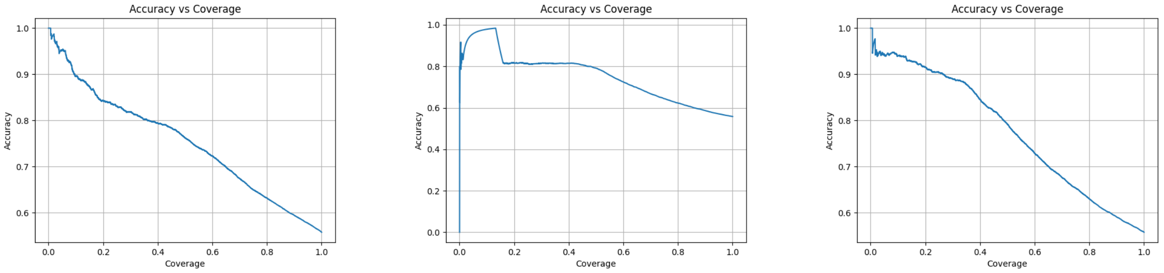

Figure 4: Accuracy vs. Coverage plots for proposed techniques: (left to right) RF calibrator, Bert embeddings with RF calibrator and top-50 PCA components of Bert embeddings with RF calibrator.

8 Contributions of group members

Our team consists of five members: Adithya Samavedhi (PID: A59019778), Naigam Shah (PID: A59019699), Samanvitha Sateesha (PID: A59020658), Avni Kothari (PID: A59008482) and Hari Vamsi Yadavalli (PID: A59018058).

Adithya Samavedhi: Adithya has contributed by implementing the approach where we consider BERT embeddings as a feature to the RF calibrator. He has also worked on the maxprobs approach completely. Along with this he has worked on generating different plots like coverage vs accuracy among others. Along with this he has worked on Introduction, Related Work, Baselines and Error Analysis sections of the report.

Naigam Shah: Naigam has contributed to baseline model with RF calibrator and has curated the datasets. He has worked on generating results for the different combinations of dataset to get coverage at 80% accuracy from the RF calibrator where feature set does not include BERT embeddings. He has also worked on getting the accuracy of the model experiments with 100% coverage. Naigam also generated results for the PCA with BERT embeddings. Along with this he has worked on Baselines and Error Analysis sections of the report.

Samanvitha Sateesha: Samanvitha has worked on getting the results for the different combinations of dataset to get coverage at 80% accuracy from the RF calibrator where feature set includes BERT embeddings. She has also worked on curation of feature sets with and without BERT embeddings along with Naigam and Adithya respectively. She conducted additional experiments to analyse feature importance according to RF calibrator. She worked on the Introduction, Related Work, Approaches and Conclusion sections of the report.

Avni Kothari: Avni worked on implementing the distance based baseline. She also worked on generating the plots for the distance baseline and the experimental results table to get the coverage at 80% accuracy. She worked on a few sections of the report and wrote the What you proposed vs. what you accomplished and the datasets section of the report.

Hari Vamsi Yadavalli: Hari worked on getting the results for different combinations of datasets to get coverage at 80% accuracy for the Multi-OOD BERT+RF model. He performed a comparative analysis for two sets of distributions as described in the 6.3 section. He experimented VAE model in two different ways as described in the appendix section and their short-comings. He has worked on sections What you proposed vs. what you accomplished and Approaches, and Appendix of the report.

9 Conclusion

We learnt about the importance of selective prediction and how using a calibrator affects the accuracy of a QA model. We learnt the significance of training a language model to abstain from the out-of-domain data. We were surprised by a few results where we expected significant improvement in one approach compared to other but that was not the case. However, we got a lot of results that were consistent throughout and followed the trend as expected. As a future scope of this project, we want to conduct experiments on larger datasets, and train our own models. We also want to work on Variational Auto Encoder and observe how that will influence the results.

References

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J., et al. (2010). Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79.

Geifman, Y. and El-Yaniv, R. (2017). Selective classification for deep neural networks. *Advances in neural information processing systems*, 30.

Jain, N. and Shenoy, P. (2022). Selective classification using a robust meta-learning approach.

Joshi, M., Choi, E., Weld, D., and Zettlemoyer, L. (2017). triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv e-prints*, page arXiv:1705.03551.

Kamath, A., Jia, R., and Liang, P. (2020). Selective question answering under domain shift. *arXiv preprint arXiv:2006.09462*.

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

Otero, J. (2009). Question generation and anomaly detection in texts. In *Handbook of metacognition in education*, pages 59–71. Routledge.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, page arXiv:1606.05250.

Seo, M., Kembhavi, A., Farhadi, A., and Hajishirzi, H. (2018). Bidirectional attention flow for machine comprehension.

Trieu, H.-L., Miwa, M., and Ananiadou, S. (2021). BioVAE: a pre-trained latent variable language model for biomedical text mining. *Bioinformatics*. btab702.

Trischler, A., Wang, T., Yuan, X., Harris, J., Sordoni, A., Bachman, P., and Suleman, K. (2017). Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Voorhees, E. (2001). Overview of the trec 2001 question answering track report. In *Proceedings of the 10th Text Retrieval Conference (TREC10), 2001*.

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

Appendix

A Multi-OOD Bert+RF

Table 5 indicates the results obtained for the second setting of Multi-OOD approach implemented using bert embeddings alongwith RF as our calibrator.

Table 5: Multi-OOD BERT+RF (for setting two)

Train Data	Test Data	Cov@80%Acc
squad, hotpot, natural	trivia	0.41875
squad, hotpot, natural	news	0.60125
squad, hotpot, trivia	natural	0.256875
squad, hotpot, trivia	news	0.574625
squad, hotpot, news	natural	0.25825
squad, hotpot, news	trivia	0.49375
squad, natural, hotpot	trivia	0.40775
squad, natural, hotpot	news	0.59975
squad, natural, trivia	hotpot	0.5245
squad, natural, trivia	news	0.570625
squad, natural, news	hotpot	0.526875
squad, natural, news	trivia	0.439375
squad, trivia, hotpot	natural	0.24
squad, trivia, hotpot	news	0.57375
squad, trivia, natural	hotpot	0.52125
squad, trivia, natural	news	0.57375
squad, trivia, news	hotpot	0.52625
squad, trivia, news	natural	0.262

B Approaches not-implemented

B.1 VAE-based OOD detection

In this approach, we plan on using a Variational AutoEncoder (VAE) to generate a latent vector representation for the given passage. This vector, along with other features like question length and top softmax probabilities, is fed into the calibrator. The VAE enables the calibrator to capture semantic and contextual information, enhancing it's ability to detect out-of-domain samples and difficult questions. By combining the latent vector with other features, we expect the approach to improve decision-making, distinguishing between question types and domains. This results in more accurate predictions and informed abstentions, enhancing the model's performance in handling challenging questions.

We tried two different approaches - pretrained VAE model, and custom built VAE model. We were surprised to see that there were no pretrained models for VAE on huggingface, so we referred to (Trieu et al., 2021) to setup their BioVAE. Though not an optimal choice for our application, we can try to proceed with it. However, we ran out of storage on google drive while trying to setup the BioVAE. So as a last resort tried to implement our own VAE model, but that did not yield anywhere, as we faced many debugging situations, and eventually had to leave it for future work weighing the time, resources into consideration.

B.2 Active Learning Approach

With active learning we can use the softmax probabilities to determine which questions the model is unsure of answering. Once it finds a question it is unsure of answering it can generate clarification questions for the user. After it has asked a few clarifying questions, the model can answer. The data generated through this process can then be supplied back to the model as training data to improve it.