Uncertainty Quantification in Question-Answering Models

Alex Kashi Harrison Termotto Håkon Grini

Abstract

Question Answering (QA) is a rich field within Natural Language Processing (NLP) comprised of many different corpora, domains, and model architectures. Certain downstream QA tasks are of critical importance, such as those in the medical domain that may directly affect human lives. However, work to understand the inherent uncertainty quantification qualities of models within QA is scarce in modern research. This work aims to explore the uncertainty quantification of BERT-based Span-Extractive QA models, one of the most commonly used and foundational models in QA. We compare the uncertainty characteristics of MC-Dropout, Deep Ensembles, and SNGP applied to ALBERT and DistilBERT base models via Entropy, Probability Variance, and Bayesian Active Learning by Disagreement. We find that ALBERT's capacity for being uncertain over a variety of answers lends itself not only to being a better perfoming model in this task, but also to creating high performing deep ensembles. Additionally, to our knowledge, this work is the first to apply SNGP to both DistilBERT and ALBERT for span-extractive question answering. All models are evaluated on SQuAD v2.0.

1. Introduction

Question answering is a long-studied task in NLP that contains variations in model architectures, question domains, and even types of questions, from multiple choice to long-form responses (Wang, 2022). Some models rely solely on a language modeling approach, such as BERT, and seek to identify a span of text containing an answer. Others combine this language model with a knowledge graph (KG) or database, and seek to use the language model to query this KG for an answer. More recent models include a graph neural network (GNN) which combines with language models to learn the ”correct” semantic graph structure for knowledge representation (Devlin et al., 2018; Yasunaga et al., 2021; Zhang et al., 2022).

Similarly, the downstream answering tasks span a wide array of topics, from common sense questions (CommonSenseQA), to questions that require some science or domain knowledge (OpenBookQA, MedQuAD), to reading comprehension questions (SQuAD) and more (Rajpurkar et al., 2016; 2018; Talmor et al., 2018; Mihaylov et al., 2018; Ben Abacha & Demner-Fushman, 2019). Each of these downstream tasks has a myriad of works proposing methods to achieve the best performance.

In this work we analyze the uncertainties inherent in BERT-based question-answering models on SQuAD v2.0, which is a span-extractive question answering dataset. Span extractive question answering, a task where a model is given a piece of text as a context and a question about this context (Rajpurkar et al., 2016). The goal is to identify the span of text in the context that best answers the question. This task is often used to evaluate the ability of NLP models to understand and reason about text.

The SQuAD v2.0 dataset consists of over 100,000 questions and their corresponding answers, extracted from a set of Wikipedia articles. The questions are open-ended, meaning that the model must use its understanding of the text to determine the answer, rather than simply selecting from a set of pre-defined answers (Rajpurkar et al., 2018). The SQuAD v2.0 dataset is an extension of the original SQuAD dataset (Rajpurkar et al., 2016). It includes more complex questions and also questions with no answer in the provided context (Rajpurkar et al., 2018).

Uncertainty quantification is an important consideration for many downstream tasks with real-life implications, such as question-answering in the medical domain that hopes to aid doctors in patient care (Lakshminarayanan et al., 2016). Yet, few existing works on span extractive QA, and QA in general, look to analyze the uncertainties inherent in these models and to determine uncertainty quantification properties characteristic of each model, or of question-answering models in general. While SQuAD v2.0 poses questions that may have no answer to them, i.e. the model needs to be able to identify when no answer in the context exists, we do not find specific discussions of this from an uncertainty perspective in many of the models that use SQuAD v2.0 as a benchmark (Rajpurkar et al., 2018).

We explore if BERT-based QA models have an accurate understanding of how likely they may be to get a question correct within the context of span-extractive question answering. Specifically, we use DistilBERT and ALBERT as base architectures, both of which are BERT variants, but are significantly smaller easier to train with our available resources (Lan et al., 2019; Sanh et al., 2019).

We utilize ensemble based modifications in order to quantify and compare uncertainties across both the techniques and the model classes. In particular we utilize Deep Ensembles, which are often touted as the state of the art method for uncertainty quantification, MC Dropout which is a common way to build ”pseudo-ensembles” from a single model, and Spectral Normalized Gaussian Process (SNGP) which seeks to build ”good” notations of uncertainty directly into the model via distance-preserving transformations and a Gaussian Process classification layer (Liu et al., 2020; Lakshminarayanan et al., 2016; Gal & Ghahramani, 2016; Nado et al., 2021; Penrod et al., 2022).

While there exist BERT-SNGP implementations and benchmarks (Nado et al., 2021), we have yet to see BERT-SNGP used in QA or any ALBERT-SNGP implementations. To the best of our knowledge, our project is the first work to do so.

Our work shows that an ability to be uncertain over a wide range of possible answers is beneficial both from a single model perspective and from the perspective of creating deep ensembles as it increases the ensemble diversity. We see this behavior in ALBERT but severely lacking in DistilBERT, causing it to be overconfident. Additionally, we see no benefit in SNGP for this task over deep ensembles, despite its specific uncertainty focused design.

2. Related Work

Contemporary QA Models are based on transformer architectures like BERT which do exceptionally well on language-based questions (Lan et al., 2019; Devlin et al., 2018). However transformers are not the only viable architecture. Researchers at DeepMind implemented a CNN+LSTM-based architecture for answering questions related to abstract reasoning commonly found on IQ tests (Santoro et al., 2018).

Another approach to QA modeling that has done well is Graph Neural Networks (GNNs). The main idea of such models is to generate a knowledge graph that structurally represents the model's knowledge about different entities and their relationships. This differs from traditional large language models that encode knowledge implicitly. A benefit of GNNs is that they are suitable for structured reasoning and for prediction explaining. Examples of GNNs applied to QA include GreaseLM and QA-GNN (Yasunaga et al., 2021; Zhang et al., 2022).

There exist few examples of uncertainty quantification in QA Models. One approach performed by Lixin Su et al. Investigated a dual model architecture for Web QA comprised of a qualify and decision model. The qualifying model produces a confidence score, and a decision model for candidate generation (Su et al., 2019).

Generally, uncertainty quantification is typically performed in one of three ways: 1. creating a prior distribution over model parameters to estimate the uncertainty in a Bayesian manner (Gal & Ghahramani, 2016). 2. Using ensembles based on the idea that if multiple uncorrelated sources are agreeing on an outcome that should attest to its accuracy. 3. Model uncertainty as an additional problem, using a separate estimator (Hendrycks & Gimpel, 2016).

Since our task is language-based we will focus on models based on bidirectional encoder representations from Transformers or BERT (Devlin et al., 2018). When BERT was published it achieved state-of-the-art performance on QA data sets such as SQuAD and SWAG. The ongoing trend in the state-of-the-art language models is the bigger the better. However, due to the resource constraints on our project, it is more practical to use an architecture that is more size/training efficient like ALBERT (Lan et al., 2019). ALBERT aims to increase the efficiency of BERT while decreasing overall size by utilizing a self-supervised loss and focusing on inter-sentence coherence.

Previous work has made an attempt to make BERT Uncertainty-aware using Spectral-normalized Neural Gaussian Processes (SNGP), this is achieved by adding a weight normalization step during training and replacing the output layer with a Gaussian process (Liu et al., 2020). The spectral normalization keeps the learned representations ”distance aware,” while the Gaussian Process layer provides better uncertainties over these distance aware representations. This method outperformed other single-model methods, and remained competitive with ensemble methods for uncertainty awareness.

3. Methodology

3.1. Models

We combine multiple common question-answering models with common uncertainty quantification techniques in the literature.

3.2. Base Language Models

DistilBERT: DistilBERT is a faster, and smaller version of BERT. Researchers used various distillation techniques like student-teacher, and pruning. Even though DistilBERT, has 40% fewer parameters and runs 60% faster than BERT, it still achieves a performance of up to 95% of the original BERT (Sanh et al., 2019).

ALBERT: ALBERT iterates on BERT, by utilizing self-supervised learning, which forces the model to better handle inter-sentence coherence. At release in 2020, the model achieved state-of-the-art performance in GLUE, RACE, and SQuAD while having fewer parameters than BERT. (Lan et al., 2019)

3.3. Model Modifications

Monte Carlo Dropout: MC dropout is a method used to generate multiple outputs for a given input, x, via running T stochastic forward passes enabled by keeping dropout on in when passing the input through the trained model. MC dropout is a commonly used approach for creating an "ensemble" of predictions in uncertainty quantification literature as it is a computationally easy way to gather a distribution over predictions without having to train many models individually, even if it is not a great approximation to Bayesian uncertainties (Folgoc et al., 2021; Gal & Ghahramani, 2016).

Deep Ensembles: In deep ensembles, one trains many different versions of the same model with different random seeds and potentially some other randomness in the data space. Then each input, x, is passed through each model, and the outputs from each model are averaged together to produce a single prediction. (Ganaie et al., 2021)

SNGP: Spectral-normalized Neural Gaussian Process introduced spectral normalization of each layer's weight matrices during training to make the hidden space transformation distance preserving. To then take advantage of this property, they replace the classification output layer with a Gaussian Process layer which then yields better uncertainty properties than a typical deep neural network (Liu et al., 2020). For BERT-based models, the spectral-normalization is applied only to the classification head. The output layer is still replaced with a Gaussian Process layer. While only constituting a small part of the network, this modification still shows useful improvements in uncertainty quantification (Liu et al., 2020), while easily fitting any BERT-based model.

3.4. Metrics

For quantification uncertainty over our model's predictions, we use a metric that was recently used in a paper investigating uncertainty quantification in transformer models (Shelmanov et al., 2021), *probability variance*, as well as *entropy* over the average class softmax probabilities for all models/elements in an ensemble.

Probability Variance: Probability variance is a way to measure disagreement between the individual models in an ensemble (or stochastic forward passes) and the average of the model outputs. Averaged over classes, it is defined as

$$\frac{1}{T}\sum_{c=1}^{C}\sum_{t=1}^{T}\left(p_t(y=c|x,\theta_t)-\bar{p}_T(y=c|x)\right)^2$$

where T is the number of models in an ensemble or stochastic forward passes, and θ_t represents the model parameters relating to member t of the ensemble (Shelmanov et al., 2021)

Entropy: Entropy over the averaged class probabilities acts as a gauge of uncertainty amongst which class is indeed the correct one. As opposed to probability variance, this measure captures the *epistemic* uncertainty, i.e. uncertainty inherent in a lack of data or model capacity. We define entropy over the average class softmax probabilities as

$$H(x)=-\sum_{c=1}^{C}\bar{p}_c log(\bar{p}_c).$$

Bayesian Active Learning by Disagreement: BALD, proposed by Houlsby et al. 2011 and used for uncertainty estimation in transformers by Shelmanov et al. 2021. BALD is defined as

$$H(x)+\frac{1}{T}\sum_{c=1}^{C}\sum_{t=1}^{T}p_t(y=c|x)log(p_t(y=c|x))$$

where $H(x)$ is entropy as defined above.

To evaluate the model as a whole, we calculated the average result of each metric across the entire validation set.

4. Experiments

All experiments are performed with Google Colab. As this is a resource-constrained environment, we organized our experiments around these limitations. We needed to limit ourselves to models with pretrained weights available freely for download and models with relatively short fine-tuning and inference times. This computational limitation is a direct reason for choosing ALBERT and DistilBERT, two lighter BERT variants with shorter training times. Weights for DistilBERT and ALBERT were pulled from HuggingFace (hug, a;b).

We fine-tune all models for span-extractive question answering on SQuAD v2.0. Due to computational requirements, we fine-tune each model for only three epochs with a batch size of 16. This takes approximately 6 hours for a single ALBERT model and 3.5 hours for a single DistilBERT model. We pull the suggested hyperparameters from the literature as these training times make it prohibitive to perform a thorough hyperparameter tuning effort on Google Colab. As a baseline, we fine-tuned pretrained uncased DistilBERT and ALBERT models for question answering on SQuAD v2.0. We use a learning rate of 2e-5, a weight decay of 0.01, a batch size of 16 and 3 epochs. These hyperparameters were kept constant amongst models unless otherwise noted.

We limit the sampling to 10 forwards passes and samples per data point for MC Dropout and SNGP, respectively. For MC Dropout, we use a dropout rate of 0.1. Our deep ensembles consist of three models fine-tuned with different random seeds. While a greater number of stochastic forward passes/samples would yield better approximations to the true distributions, and the typical number of ensembles in a deep ensemble appears to be 5-10 in literature (Liu et al., 2020), we felt these choices were feasible to accomplish with our constrained resources while still providing insight into model behavior.

For SNGP we use a GP kernel scale of 1.0, 2048 inducing points, a layer norm epsilon value of 1e-12, 1 power iteration, a spectral norm bound of 0.95, a GP covariance momentum term of 0.999 and a GP covariance ridge penalty of 1e-3.

5. Results

The results of all experiments and the standard SQuAD v2.0 metrics are reported in Table 1. We compute the variance of the total score, probability variance, and entropy of the start and end spans as detailed in the methodology section above. These results are reported in Table 2.

Baseline DistilBERT achieved an F1 score of 67.55 and an exact match (EM) score of 64.02. Baseline ALBERT achieved an F1 score of 81.83 and an EM of 78.72.

ALBERT fares better in task performance, and is more confident about its results, having approximately a 2x lower entropy albeit tending to have a higher probability variance.

There is a surprising drop in performance for DistilBERT SNGP when compared to plain DistilBERT. This potentially suggests the hyperparameters used were not ideal and more hyperparameter tuning is necessary.

When the models are ensembled, ALBERT gains 0.93 points to its F1 score, while DistilBERT gains 1.33 points reported in Table 1. ALBERT's probability variance and BALD scores are slightly larger, than DistilBERT's, however ALBERT's entropy is lower than DistilBERT's, as displayed in Table 2. Ultimately the ALBERT deep ensemble is the best performing out of all experiments.

6. Discussion

6.1. SNGP

Given the limitation in computational resources we were unable to perform much hyperparameter tuning for SNGP-based methods. While we pulled the parameters used in the official implementation we cannot confidently say that our results with SNGP are the best they could be, and certain hyperparameters, such as the spectral norm bound and number of epochs can have a significant effect on SNGP performance (Liu et al., 2020; Nado et al., 2021). Indeed, the large difference in performance between DistilBERT and DistilBERT-SNGP, compared to the very similar performance between ALBERT and ALBERT-SNGP, suggests that the model did not train properly. We see no clear advantage in uncertainty quantification over deep ensembles as evidenced by the results on the dataset and the uncertainty metrics in Table 2.

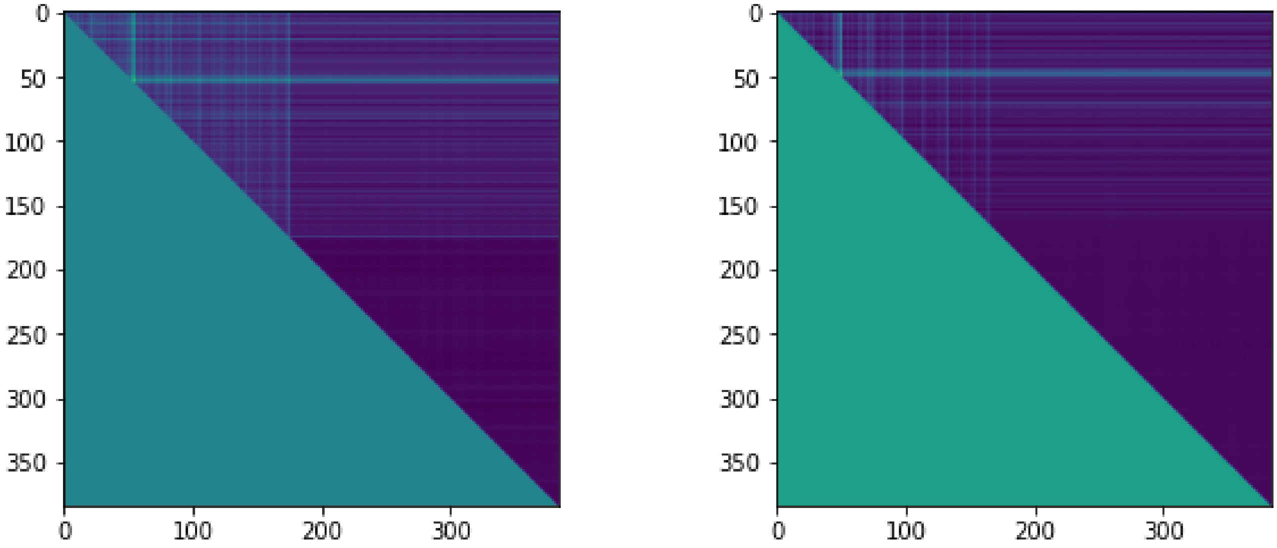
6.2. MC Dropout

When looking at the different uncertainty quantification metrics for MC dropout for ALBERT and DistilBERT it is clear that the DistilBERT model has a significantly higher entropy and BALD score, indicating that it is much less certain in its answers on a per-model basis. However, its probability variance across the models is slightly lower. This is in line with the results seen for the other methods as well. There could be several reasons for why DistilBERT would be more uncertain than ALBERT when evaluated with MC Dropout. One potential reason could be that DistilBERT has many more dropout layers, which naturally could mean that the model is more impacted by the stochastic nature of MC dropout, where the dropout layers are active in the forward passes and sets some of the outputs of those layers to 0 with a given probability of 0.1. When more values are set to 0 within the model the model will likely be more confused, leading to a more widespread distribution of the probability density and higher entropy.

It could also just be that the entropy is higher for DistilBERT because it is generally performing worse on the dataset with significantly lower exact and F1 scores. When the model cannot find the correct answer it is reasonable that it would be more uncertain, which could explain the significantly higher entropy score.

6.3. Deep Ensembles

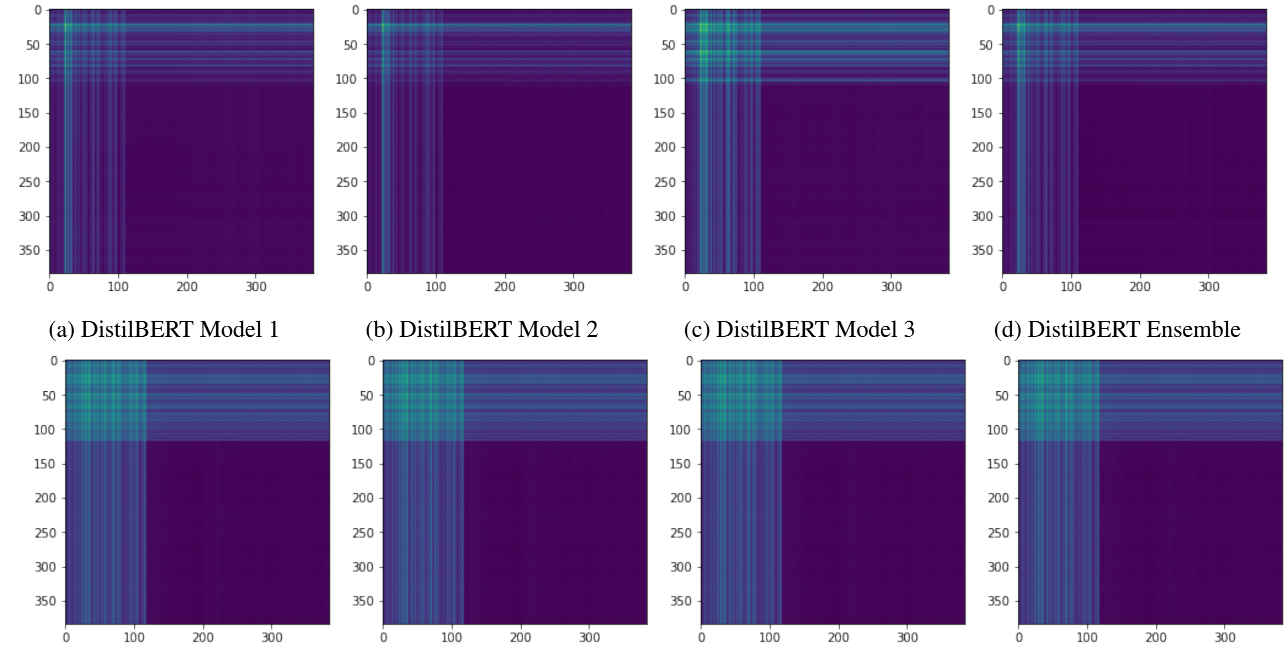
When ensembled, DistilBERT achieved a lower probability variance and BALD score when compared to ALBERT, even though it was less accurate. This result exemplifies the need for explainable AI. DistilBERT is more confident in its wrong answers, perhaps because the model is not powerful enough to capture the nuance of the context. This could be a



(a) ALBERT

(b) Distilbert

Figure 1. Logits matrix of the sum of start and end logits for each start and end position for ALBERT and DistilBERT for the context provided in appendix A (self-normalized)



(a) DistilBERT Model 1

(b) DistilBERT Model 2

(c) DistilBERT Model 3

(d) DistilBERT Ensemble

(e) ALBERT Model 1

(f) ALBERT Model 2

(g) ALBERT Model 3

(h) ALBERT Ensemble

Figure 2. Logits matrix of the sum of start and end logits for each start and end position for ALBERT and DistilBERT for the context provided in appendix B (self-normalized), ALBERT correctly identified the answer in all models, while DistilBERT did not identify the right answer in any model

Model	exact	F1	HasAns_exact	HasAns_F1	NoAns_exact	NoAns_f1	best_exact	best_f1
DistilBERT	64.01	67.55	67.20	74.27	60.84	60.84	64.01	67.55
DistilBERT SNGP	47.44	51.29	33.86	41.56	60.99	60.99	50.32	51.72
DistilBERT SNGP Sampled	30.90	38.56	19.33	34.67	42.44	42.44	50.11	50.11
DistilBERT MC Dropout	52.52	55.48	31.93	37.87	73.05	73.05	52.62	55.56
DistilBERT Ensemble (3)	65.38	68.88	69.08	76.09	61.70	61.70	65.38	68.88
ALBERT	78.71	81.82	75.16	81.38	82.25	82.25	78.72	81.83
ALBERT SNGP	78.10	81.43	74.00	80.67	82.19	82.19	78.10	81.43
ALBERT SNGP Sampled	70.81	77.59	61.67	75.27	79.92	79.92	70.81	77.59
ALBERT MC Dropout	77.80	80.67	72.59	78.33	82.99	82.99	77.81	80.67
ALBERT Ensemble (3)	**79.75**	**82.75**	**76.20**	**82.19**	**83.30**	**83.30**	**79.76**	**82.75**

Table 1. Key metrics of ALBERT and DistilBERT after fine-tuning on SQuAD v2.0 averaged across the validation set.

Model	Probability Var	BALD	Entropy
DistilBERT SNGP Sampled	1.43e-3	1.69	2.35
DistilBERT MC Dropout	7.32e-5	0.11	1.79
DistilBERT Ensemble (3)	6.19e-5	0.056	0.76
ALBERT SNGP Sampled	4.91e-4	0.402	0.96
ALBERT MC Dropout	9.62e-5	0.089	0.82
ALBERT Ensemble (3)	7.26e-5	0.066	0.61

Table 2. Key metrics of ALBERT and DistilBERT after fine-tuning on SQuAD v2.0 for 3 epochs

dangerous phenomenon when applied in a real-life scenario. The metrics seem to suggest that the ALBERT ensemble is more capable of ”good” uncertainty quantification than DistilBERT. While we hypothesize that more DistilBERT models could be used to ameliorate some of the disparity between the two models as we were limited to only three here due to computational constraints, Figure 2 seems to suggest that this difference in uncertainty awareness between DistilBERT and ALBERT ensembles is inherent in a single instance of the model class as well. This is a useful finding for practitioners looking to utilize BERT-based models in span-extractive QA.

Figure 2 clearly illustrates one example of the overconfidence typical of DistilBERT. Each model was proposed the question found in appendix B. DistilBERT was highly confident in its prediction, only activating a few start and end points. Figure 2(c) demonstrates that some initializations are heterogeneous, increasing the uncertainty of the ensemble. ALBERT's activations demonstrate significantly more uncertainty in the top left quadrant, translating to a model that outputs more nuanced predictions. Ultimately, all ALBERT models made the correct prediction. In this example, even though there was only one correct answer, ”Catholicism”, several related phrases, appeared near the correct answer, like ”(Christianity)”, therefore a more uncertain model seems intuitive.

Surprisingly, ALBERT's entropy was lower than DistilBERT. we hypothesize this is because the errors DistilBERT makes are more randomly distributed while, ALBERT makes more nuanced predictions around select candidates.

7. Conclusion

It appears that the inherent uncertainty properties between ALBERT and DistilBERT do differ noticeably. We see that ALBERT is able to identify potential answers in more places in the context, leading to less overconfident answers than DistilBERT. Additionally we see this allows deep ensembles of ALBERT to have more diversity as compared to DistilBERT, as is evidenced by the probability variances. When ensembled ALBERT has slightly higher uncertainty due to an ensemble diversity compared to DistilBERT as evidenced by the higher probability variance. This is seen in table 2. When examining the results in figure 2 it is clear that ALBERT outputs a more nuanced prediction. amongst each member in the ensemble.

While SNGP is designed specifically for beneficial uncertainty quantification properties, we don't see any significant advantage in using it over deep ensembles. Additionally, the added computational complexity and increased requirement for hyperparameter tuning make it a less desirable technique from a practical standpoint.

In applications of high-risk ALBERT, or a similar model that may be able to provide many reasonable candidate answers, would be more suitable. In this way it may be able to provide these candidate answers to a human when it is too uncertain and the human may make the final call. Ultimately no model is completely infallible, and therefore safety controls and oversights are needed in any downstream application of critical importance.

8. Impact Statement

NLP-based models are rapidly improving, we are rapidly reaching a point where more and more trust will be put in these models. With the explosion in use of OpenAI's new model ChatGPT, it is clear that these models will have a profound impact on our society. However, these models are not infallible, and it is important that the models are able to convey when they are confident about something and when they are more uncertain. If not, it is possible that mistakes are not caught by the users who will just assume that the model knows what it is talking about. This can have grave consequences in domains such as health-care, power plant operations, or autonomous driving to name a few.

The importance of uncertainty quantification in these contexts naturally means that the scrutiny put on uncertainty quantification methods should be severe. These methods and metrics should not only be able to indicate when the model is uncertain about its answer, but also be able to show when the model is very confident to avoid needlessly confuse or worry the user.

Another potential issue with uncertainty quantification in a question answering context is that it can give the user a sense of misplaced trust in the models. The uncertainty quantification only quantifies how confident or uncertain the models are, not if they in fact are right or wrong, and just because a model is confident that does not mean it is right.

Our experiments are a step in the right direction for exploring these uncertainty quantification properties inherent in many of the models used today, but more work in this area is needed. We urge all practitioners to always be mindful of their models failure modes, and to be particularly aware of models ability to underestimate their own "understanding."

9. Appendix

A. Example 1 Input and Output:

Example context: The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

Example Question: In what country is Normandy located?

True Output(s): France

B. Example 2 Input and Output:

Example context: The descendants of Rollo's Vikings and their Frankish wives would replace the Norse religion and Old Norse language with Catholicism (Christianity) and the Gallo-Romance language of the local people, blending their maternal Frankish heritage with Old Norse traditions and customs to synthesize a unique "Norman" culture in the north of France. The Norman language was forged by the adoption of the indigenous langue d'oïl branch of Romance by a Norse-speaking ruling class, and it developed into the regional language that survives today.

Example Question: What was the Norman religion?

True Output(s): Catholicism

References

Albert for question answering. https://huggingface.co/docs/transformers/model_doc/albert#transformers.AlbertForQuestionAnswering, a. Accessed: 2022-10-24.

Distilbert for question answering. https://huggingface.co/docs/transformers/v4.25.1/en/model_doc/distilbert#transformers.DistilBertForQuestionAnswering, b. Accessed: 2022-12-15.

Ben Abacha, A. and Demner-Fushman, D. A question-entailment approach to question answering. *BMC Bioinform.*, 20(1):511:1–511:23, 2019. URL https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-3119-4.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Folgoc, L. L., Baltatzis, V., Desai, S., Devaraj, A., Ellis, S., Manzanera, O. E. M., Nair, A., Qiu, H., Schnabel, J. A., and Glocker, B. Is MC dropout bayesian? *CoRR*, abs/2110.04286, 2021. URL https://arxiv.org/abs/2110.04286.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., and Suganthan, P. N. Ensemble deep learning: A review. 2021. doi: 10.48550/ARXIV.2104.02395. URL https://arxiv.org/abs/2104.02395.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. Bayesian active learning for classification and preference learning, 2011. URL https://arxiv.org/abs/1112.5745.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles, 2016. URL https://arxiv.org/abs/1612.01474.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942, 2019. URL http://arxiv.org/abs/1909.11942.

Liu, J. Z., Lin, Z., Padhy, S., Tran, D., Bedrax-Weiss, T., and Lakshminarayanan, B. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *CoRR*, abs/2006.10108, 2020. URL https://arxiv.org/abs/2006.10108.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

Nado, Z., Band, N., Collier, M., Djolonga, J., Dusenberry, M., Farquhar, S., Filos, A., Havasi, M., Jenatton, R., Jerfel, G., Liu, J., Mariet, Z., Nixon, J., Padhy, S., Ren, J., Rudner, T., Wen, Y., Wenzel, F., Murphy, K., Sculley, D., Lakshminarayanan, B., Snoek, J., Gal, Y., and Tran, D. Uncertainty Baselines: Benchmarks for uncertainty & robustness in deep learning. *arXiv preprint arXiv:2106.04015*, 2021.

Penrod, M., Termotto, H., Reddy, V., Yao, J., Doshi-Velez, F., and Pan, W. Success of uncertainty-aware deep models depends on data manifold geometry. 2022. doi: 10.48550/ARXIV.2208.01705. URL https://arxiv.org/abs/2208.01705.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016. URL http://arxiv.org/abs/1606.05250.

Rajpurkar, P., Jia, R., and Liang, P. Know what you don't know: Unanswerable questions for squad. *CoRR*, abs/1806.03822, 2018. URL http://arxiv.org/abs/1806.03822.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2019. URL https://arxiv.org/abs/1910.01108.

Santoro, A., Hill, F., Barrett, D. G. T., Morcos, A. S., and Lillicrap, T. P. Measuring abstract reasoning in neural networks. *ArXiv*, abs/1807.04225, 2018.

Shelmanov, A., Tsymbalov, E., Puzyrev, D., Fedyanin, K., Panchenko, A., and Panov, M. How certain is your Transformer? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1833–1840, Online, April 2021. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2021.eacl-main.157.

Su, L., Guo, J., Fan, Y., Lan, Y., and Cheng, X. Controlling risk of web question answering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 115–124, 2019.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *CoRR*, abs/1811.00937, 2018. URL http://arxiv.org/abs/1811.00937.

Wang, Z. Modern question answering datasets and benchmarks: A survey, 2022. URL https://arxiv.org/abs/2206.15030.

Yasunaga, M., Ren, H., Bosselut, A., Liang, P., and Leskovec, J. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. *arXiv preprint arXiv:2104.06378*, 2021.

Zhang, X., Bosselut, A., Yasunaga, M., Ren, H., Liang, P., Manning, C. D., and Leskovec, J. Greaselm: Graph reasoning enhanced language models for question answering. *arXiv preprint arXiv:2201.08860*, 2022.