Controlling Risk of Web Question Answering

Lixin Su, Jiafeng Guo, Yixing Fan, Yanyan Lan, and Xueqi Cheng
CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China
University of Chinese Academy of Sciences, Beijing, China
{sulixin,guojiafeng,fanyixing,lanyanyan,cxq}@ict.ac.cn

ABSTRACT

Web question answering (QA) has become an dispensable component in modern search systems, which can significantly improve users' search experience by providing a direct answer to users' information need. This could be achieved by applying machine reading comprehension (MRC) models over the retrieved passages to extract answers with respect to the search query. With the development of deep learning techniques, state-of-the-art MRC performances have been achieved by recent deep methods. However, existing studies on MRC seldom address the predictive uncertainty issue, i.e., how likely the prediction of an MRC model is wrong, leading to uncontrollable risks in real-world Web QA applications. In this work, we first conduct an in-depth investigation over the risk of Web QA. We then introduce a novel risk control framework, which consists of a qualify model for uncertainty estimation using the probe idea, and a decision model for selectively output. For evaluation, we introduce risk-related metrics, rather than the traditional EM and F1 in MRC, for the evaluation of risk-aware Web QA. The empirical results over both the real-world Web QA dataset and the academic MRC benchmark collection demonstrate the effectiveness of our approach.

CCS CONCEPTS

• **Information systems → Question answering**.

KEYWORDS

risk control, predictive uncertainty, Web QA

ACM Reference Format:
Lixin Su, Jiafeng Guo, Yixing Fan, Yanyan Lan, and Xueqi Cheng. 2019. Controlling Risk of Web Question Answering. In *42nd Int'l ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR'19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331184.3331261

1 INTRODUCTION

It has been a long-term expectation that search systems can not only connect people to documents, but also connect people directly to information. One step towards this is to provide direct answers (e.g., facts, definitions or a short piece of text) to users' search queries. In this way, we can significantly improve users' search experience by saving their efforts on clicking and reading the result pages. This would be especially valuable in mobile search scenarios where browsing is quite difficult due to the limited screen size.

在现代搜索引擎中,在检索模块之上加一个QA模块.

In modern search systems, the above target could be achieved by deploying a Web question answering (QA) component [4, 12] upon the retrieval module: Given a search query and a set of top-ranked passages, the Web QA component determines whether the candidate set contains any direct answer and extracts the answer as the output if it exists. Without loss of generality, the Web QA component could be implemented by applying machine reading comprehension (MRC) models [6, 33, 43, 44] over the retrieved passages. In recent years, with the development of deep learning techniques, state-of-the-art MRC performances have been achieved by a variety of deep MRC models [7, 20].

Web QA 的工作思想

不可忽视的

However, directly applying existing MRC techniques for Web QA brings non-negligible risks in practice. Facing with open-domain long-tail queries and noisy search results, even the most advanced deep MRC models are prone to produce unreliable answers (i.e., overconfident incorrect answers [8]). These unreliable answers may hurt users' search experience significantly, which we will discuss later in section 3. Therefore, for practical applications, it is expected that people can be aware of MRC models' confidence or uncertainty[1] [13] on the predicted answers to enable risk control in Web QA. This calls for the investigation on the predictive uncertainty issue of MRC, a core research problem we want to tackle in this work. Unfortunately, most previous research on MRC has been focused on improving the model effectiveness [37, 42, 46, 47]. There has been little work addressing the predictive uncertainty issue of MRC. Note that there have been some studies tackling the unanswerable question (i.e., null answer) problem in MRC [20, 23, 34], which is different from the predictive uncertainty issue we discuss here.

WebQA 在实际应用中的风险性.

In recent years, there has been increasing interest on the predictive uncertainty issue of deep learning models in the machine learning (ML) field. Existing work on predictive uncertainty can mainly be divided into three categories. The first class stems from Bayesian Neural Networks [4], which aims to estimate the predictive uncertainty by introducing a prior distribution over the model parameters [14, 29]. However, these methods are usually computationally expensive and the effectiveness largely depends on the correctness of the prior assumption. The second class [16, 25] borrows the ensemble idea, which tries to estimate the predictive uncertainty by model average based on a bag of models learned with different initialization or from different epochs. The third

深度学习模型的预测不确定性问题,

目前的研究主要分3类

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331261

[1] We use confidence and (negative) uncertainty interchangeably in this paper.

class [2, 18, 30] takes the predictive uncertainty estimation as an additional task, where some heuristic strategies have been employed to learn another prediction model. However, most of these studies have been conducted in the Computer Vision (CV) field, with only a few in natural language processing (NLP), including semantic parsing and machine translation [3, 10]. So far as we know, there has been little work tackling the predictive uncertainty issue of MRC.

In this paper, we propose to control the risk of Web QA by modeling the predictive uncertainty of deep MRC models[2]. We first take an in-depth analysis over the risk in the Web QA scenario and show its speciality as compared with the risks in other applications. Based on this analysis, we then introduce a novel risk control framework for Web QA with the *selective classification* idea [15]. Specifically, our framework consists of two major components, namely a *qualify* model and a *decision* model. The qualify model is used for predictive uncertainty estimation, which produces a confidence score for the prediction of an MRC model. Inspired by the probe idea introduced by Alain and Bengio [1], we design a novel and general *PROBE-CNN* model to act as the qualify model based on the unified abstraction of deep MRC models. The decision model aims to learn a rejection region over the confidence score for selective output. By rejecting those low-confidence predictions of an MRC model, we can well control the risk of Web QA. Note that our framework is a post-processing framework which means it could be applied over almost any existing state-of-the-art deep MRC model.

否定区域

For evaluation, traditional widely used metrics such as EM and F1 only focus on the effectiveness of an MRC model. In this work, we introduce risk-related metrics following the idea in [15, 16], including *coverage*, *risk* and *AURC* for risk-aware Web QA evaluation. We conduct extensive experiments on two large scale publicly available datasets. One is a real-world WebQA dataset and the other is a widely adopted academic MRC benchmark collection [35]. Two representative deep MRC models, i.e., BIDAF [37] and BERT [7] are employed, and several state-of-the-art uncertainty estimation methods have been compared within our risk control framework. The experimental results show that our approach can outperform all the baseline methods in terms of all the evaluation metrics. Besides, we also provide detailed analysis to gain better understanding on our probe-based qualify model.

The main contributions of this paper include:

- We introduce the risk control problem of Web QA which calls for addressing the predictive uncertainty issue of MRC models. So far as we know, this is the first work tackling such uncertainty issue of MRC models in Web search.
- We propose a risk control framework for Web QA with a novel and general qualify model designed based on the probe idea and the unified abstraction of deep MRC models.
- We introduce risk-aware evaluation metrics for Web QA and conduct extensive experiments to demonstrate the effectiveness of our approach.

2 RELATED WORK

In this section, we will briefly survey three related topics to our work, including Web QA, MRC, and model uncertainty.

2.1 Web QA

Web QA is an important task across both NLP and information retrieval (IR) fields, which aims to answer users' questions using Web resources. Web QA is a type of open-domain QA in the sense that queries are usually from unconstrained categories and resources are typically unstructured Web documents. Without loss of generality, Web QA could be performed by a two-step process, i.e., relevant document retrieval and answer extraction. In this work, we focus on the answer extraction part.

Early techniques and systems on Web QA have been largely driven by the TREC QA track [41]. Most of these studies were based on shallow linguistic processing and complicated rules. For example, AskMSR [4] was a search-engine based QA system that relies on data redundancy to find short answers. Moldovan [31] proposed window-based word scoring technique to rank potential answer pieces for Web QA.

Recently, researchers have released several large scale datasets, such as SearchQA[11], TriviaQA[22] and Quasar [9], to accelerate the study and application of deep learning techniques for Web QA. Based on these datasets, Wang et al. [43] proposed to apply a deep ranker-reader model to extract answers from the top passages using reinforcement learning. Some other works [27, 44] tried to employ deep MRC models to extract answers by either de-noising the data or re-ranking multiple candidate answers. In this work, we focus on controling the risk of Web QA, instead of proposing another Web QA model.

2.2 Machine Reading Comprehension

The MRC task is to predict the exact answer from a context passage for a given question [5]. According to the answer form, the MRC task can be further divided into four categories, namely cloze-style MRC [19], multi-choice MRC [24, 36], span-prediction based MRC [34, 35] and free-form MRC [32]. Here, we focus on the span-prediction based MRC, which is more practical and prevalent, and also highly related to Web QA [6].

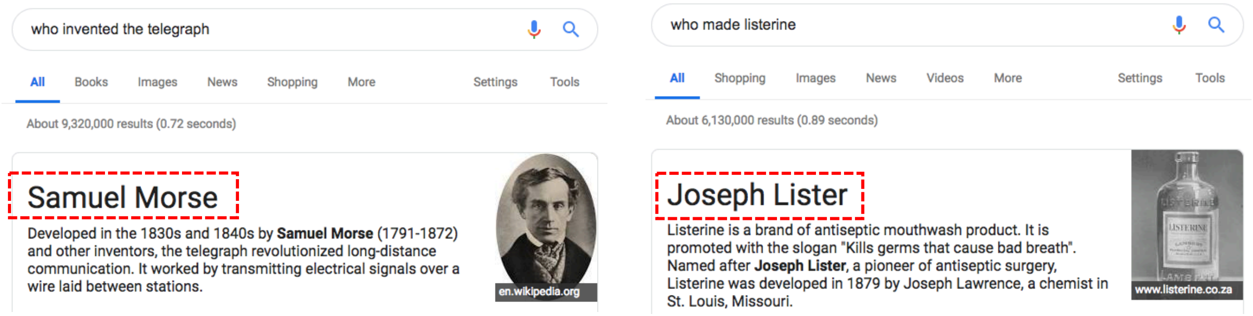
MRC的四种类型

In recent years, a large number of models have been proposed to address the span prediction task in MRC. In general, these models typically consist of two components, namely the reading component and the answering component. The reading component is used to capture the interactions between the question and the passage, and to collect the evidences from the passage. Researchers have introduced different types of architectures for the reading component, such as matching-LSTM [42], gated attention and self-attention mechanisms [44], cross-layer interaction and hierarchical interaction [21, 39, 45]. The answering component aims to extract the exact answer span from the passage based on the signals from the reading component. Most works [42] utilized pointer network [40] to predict the answer span in this component.

MRC Model的一般结构: 阅读模块、回答模块

Early studies on MRC mainly focused on the questions whose answer definitely exists in the context passage. Recently, more attentions have been paid to the MRC task with unanswerable questions [34]. A common solution is to add a null position to the

[2] We focused on deep MRC models since they are most popular and advanced techniques for Web QA.



who invented the telegraph

About 9,320,000 results (0.72 seconds)

Samuel Morse

Developed in the 1830s and 1840s by **Samuel Morse** (1791-1872) and other inventors, the telegraph revolutionized long-distance communication. It worked by transmitting electrical signals over a wire laid between stations.

who made listerine

About 6,130,000 results (0.89 seconds)

Joseph Lister

Listerine is a brand of antiseptic mouthwash product. It is promoted with the slogan "Kills germs that cause bad breath". Named after **Joseph Lister**, a pioneer of antiseptic surgery, Listerine was developed in 1879 by Joseph Lawrence, a chemist in St. Louis, Missouri.

(a) A correct answer for the query "*who invented the telegraph*".

(b) A wrong answer for the query "*who made listerine*".

Figure 1: Direct answers from the Web QA component in modern search engines.

passage [23, 38]. Some work also introduced an additional model to detect those unanswerable questions [20].

Unlike most previous work which mainly focused on the effectiveness of the MRC model, our work focus on addressing its predictive uncertainty.

2.3 Predictive Uncertainty in Deep Models

目前不确定性评估的三种方法:

Modeling predictive uncertainty [13] is critical for robust or safe AI. Recently, the uncertainty prediction problem has attracted a lot of interests, especially in the classification task. Without loss of generality, existing methods can be categorized into three classes. The first class is based on the Bayesian Neural Network, which introduces a prior distribution over the model parameters [14, 28] to estimate the uncertainty. Their performance depends on the concrete form of approximation made due to computational constraints. Monte-Carlo Dropout is a kind of approximation using an ensemble of multiple stochastic forward passes and computing the spread of the results. However, the high computational cost still makes it intractable for complex deep neural networks. The second class uses the ensemble idea to estimate the predictive uncertainty, with the assumption that multiple predictions on the same instance can provide information about the uncertainty. For example, Geifman et al. [16] proposed to ensemble model information from different training epochs for the uncertainty estimation. The third class takes uncertainty estimation as an additional prediction problem, and employs some heuristic function for estimation. In [18], the simple *max* function was applied to the model's last output to produce the uncertainty score. More complex functions such as local density based functions [30] have also been utilized for estimation.

Most previous work on predictive uncertainty estimation has been proposed in the ML community and mainly applied on CV tasks. To the best of our knowledge, this is the first work to address the predictive uncertainty issue of MRC for the Web QA task.

3 ANALYSIS ON THE RISK OF WEB QA

Web QA has become a key feature in modern search engines. As shown in Figure 1(a), given a search query "who invented the telegraph", beyond the traditional *ten blue links*, a direct answer "Samuel Morse" is provided through Web QA. In this way, users' information need could be satisfied without browsing any returned Web page. However, Web QA might also face unexpected risks in practice. For example, as shown in Figure 1(b), the returned direct answer for the query "who made listerine" is incorrect since its true developer is "Joseph Lawrence" as shown in the corresponding passage. Such incorrect answers might mislead users, and even become harmful if this happens for search queries seeking for medical or legal suggestions. In the following, we try to study two research questions to gain a better understanding of the risk in the Web QA scenario.

Table 1: Categorization of the output in Web QA.

	Model Prediction		
	Direct Answer(D)		Null Answer(N)
Answerable(A)	AD^+	AD^-	AN
Unanswerable(U)	UD		UN

正确输出的情况

真正的风险 of WebQA

Firstly, what are the true risk of Web QA? As shown in Table 1, the input queries[3] in Web QA could be classified into two categories, i.e., answerable (A) and unanswerable (U). Answerable queries refer to the queries whose answer exists in the top returned results, while unanswerable queries are the opposite. Meanwhile, considering the model prediction, a QA model may output a direct answer (D), either correct (+) or not (-), or a null answer (N) for a given query. Therefore, the output of Web QA must fall into one of the five folds, where AD^+/AD^- denotes that the model predicts an correct/incorrect answer for an answerable query respectively, AN/UN denotes that the model predicts a null answer for an answerable/unanswerable query respectively, and UD denotes that the model predicts a direct answer for an unanswerable query. As we can see, there is obviously no risk in the AD^+ and UN folds. The true risks of Web QA only come from the AD^- and UD folds, where the model predictions are incorrect with respect to the ground truth. It is noteworthy that there is no risk in the AN fold either, even though the model's prediction is wrong. This is because users' search experience may not be influenced when the model provides a null answer. Such a unique risk structure makes Web QA different from those CV or NLP tasks (which do not have null answer predictions) where risks have been tackled before.

Secondly, what causes the risk of Web QA? In the Web QA scenario, when we rely on MRC techniques for answer prediction, the risk happens when the MRC model cannot provide reliable

[3] Note that in practice there would be a triggering component (i.e., an intent classifier) which determines whether to trigger the Web QA component, i.e., whether a query is likely to have a short answer. However, this is out of the scope of this paper and here we refer to input queries as those have passed such triggering component.

Table 2: Basic notations used in this paper.

Meaning	Notation
Query space	$\mathcal{Q}$
Passage space	$\mathcal{P}$
Answer space	$\mathcal{A}$
Query representation matrix in layer t	$Q^{(t)}$
Passage representation matrix in layer t	$P^{(t)}$
Start position vector of the answer span	$\vec{s}$
End position vector of the answer span	$\vec{e}$

predictions but people are not aware of it. There are two major reasons related to this problem. On one hand, although MRC models can achieve human-like performance on some close-world dataset [35], it will encounter significant challenges in Web QA due to its open-domain nature. Facing with long-tail queries and noisy search results, even the most advanced deep MRC models cannot generalize well over many previously unseen QA patterns. On the other hand, recent studies have revealed that deep models are poor at uncertainty qualification [17] and tend to produce overconfident predictions [8]. In other words, although deep MRC models could provide a probability on its answer, that probability cannot well reflect its confidence on the prediction. In summary, deep MRC models are prone to produce overconfident but incorrect answers, leading to the risk of Web QA.

Based on the above analysis, we can find that Web QA is a special problem with its unique risk structure. The risk of Web QA comes from people's unawareness of the model uncertainty on its predictions. Therefore, in the following, we try to control the risk of Web QA by modeling the predictive uncertainty of deep MRC models.

4 RISK CONTROL FRAMEWORK

In this section, we describe the risk control framework for Web QA. Note that we do not aim to propose a specific MRC model that can qualify its uncertainty better than existing methods, but rather design a general framework that could work for a large variety of MRC models. In this way, our framework could be easily integrated with the practical Web QA system and do not restrict any future upgrade of the applied MRC models. Towards this purpose, we adopt the third class modeling methodology on predictive uncertainty, which takes the uncertainty estimation as an additional prediction problem, and make the framework a post-processing one so that it could be applied over any existing state-of-the-art deep MRC models. A key difference of our framework from previous uncertainty estimation work is that we do not simply employ some heuristic estimation functions, but rather introduce a general probe-based uncertainty prediction model specifically designed for modern deep MRC models.

本文提出的架构不针对任何一个MRC model，而是通用的，因此是采用第3类不确定性评估方法，把该问题看成一个额外的预测问题

Some basic notations frequently used in this paper are listed in Table 2. Overall, our framework takes an MRC model as the input, and learns two new models, namely the qualify model and the decision model, to control the risk. The formal definition of each model is as follows.

The MRC model f is a function which predicts an answer a based on a passage p with respect to a search query q, denoted as $f : (\mathcal{Q}, \mathcal{P}) \rightarrow \mathcal{A}$. Note here we only consider the MRC model f which takes a single passage for direct answer prediction for simplicity (which is also a typical case in practical Web QA where only the top relevant passage is used due to the efficiency concern). However, our framework is not limited to that, but could well adapt to answer extraction from multiple retrieved passages.

The Qualify model g is core in our framework, which aims to estimate the predictive uncertainty of the MRC model. Specifically, given an MRC model f and a specific query-passage instance (q,p), the qualify model g outputs a confidence score for f's prediction, denoted as $g(q,p|f) \in [0,1]$. Here, the confidence score represents the likelihood that f's prediction is correct. The detailed implementation of the qualify model g will be described in Section 5.

The Decision model h is used to make the final decision whether we shall abandon f's prediction by defining a rejection region over the confidence score,

$$h(g) \triangleq \begin{cases} 1 & \text{if } g(q,p|f) \geq \theta \\ 0 & \text{if } g(q,p|f) < \theta, \end{cases} \tag{1}$$

where $\theta \in [0,1]$ denotes the confidence threshold parameter for decision. When $h(g) = 1$, we choose to trust/output f's prediction, otherwise not.

Learning of the framework could be derived as follows. Given an MRC model f, we can write down the *risk* as 是一个给定的 loss function

$$R(f|P) = E_{Pr(q,p,a)}[\ell(f(q,p),a)], \tag{2}$$

where $\ell : \mathcal{A} \times \mathcal{A} \rightarrow \mathbb{R}^+$ is a given loss function, and $Pr(q,p,a)$ is an unknown data distribution over $\mathcal{Q} \times \mathcal{P} \times \mathcal{A}$. Given a labeling set $S_m = \{(q_i,p_i,a_i)\}_{i=1}^m \in (\mathcal{Q} \times \mathcal{P} \times \mathcal{A})$, the *empirical risk* is as follows:

$$\hat{r}(f,h,g|S_m) \triangleq \frac{\sum_{i=1}^m \ell(f(q_i,p_i),a_i)}{m}. \tag{3}$$

Based on the analysis in Section 3, the loss function in Web QA could be defined as follows

$$\ell(f(q,p),a) \triangleq \begin{cases} 0 & f(q,p) \in AD^+ \cup AN \cup UN \\ 1 & f(q,p) \in AD^- \cup UD. \end{cases} \tag{4}$$

可以忽略

In traditional uncertainty estimation, an *optimal* qualify model g (for f) should reflect true loss monotonicity in the sense that the confidence score should be higher for the instance with lower loss. However, due to the special risk structure in Web QA, this is not always the case. Specifically, although there is no loss in the AN set (i.e., null answer for answerable query) as shown in Table 1 , we do not expect a high confidence in this area since the predictions are actually wrong. Moreover, we could omit the estimation for instances in the UN (i.e., null answer for unanswerable query) and AN set, since there would be no gain even if we predict a confidence score for those instances. Therefore, we only need to require the monotonicity between instance $f(q_i,p_i) \in AD^+$ and $f(q_j,p_j) \in AD^- \cup UD$,

单调性

$$g(q_i,p_i|f) \geq g(q_j,p_j|f). \tag{5}$$

Based on this target, it turns out we could optimize the following pairwise objective function in order to learn the qualify model g,

$$loss = max(0, 1 - g(q_i,p_i|f) + g(q_j,p_j|f)), \tag{6}$$

where $f(q_i,p_i) \in AD^+$ and $f(q_j,p_j) \in AD^- \cup UD$.

With the learned qualify model g, we can use the decision model h to selectively output f's prediction and obtain the following *empirical selective risk*

$$\hat{r}(f,h,g|S_m) \triangleq \frac{\sum_{i=1}^m \ell(f(q_i,p_i),a_i)h(g)}{\sum_{i=1}^m h(g)}. \in [0,1] \tag{7}$$

Note here $\hat{r}(f, h, g|S_m) \in [0, 1]$. Finally, to decide the confidence threshold θ in h in practice, we only need to setup a desired risk level and find the θ that satisfies the risk level over S_m (which is not necessarily the training set).

5 QUALITY MODEL IMPLEMENTATION

In this section, we focus on describing the core component in our risk control framework, i.e., the qualify model. As mentioned above, our goal is to construct a general risk control framework that could be applied over a large variety of deep MRC models. This in turn requires that the qualify model could adapt to different MRC models. In this paper, we mainly focus on deep MRC models designed for answer span prediction, which are prevalent and state-of-the-art techniques in Web QA. We will leave other MRC models, e.g., traditional models or generative models, for the future work.

Our main design for a general qualify model is inspired by the probe idea [1], which attempts to use linear-classifier probes to understand neural network models. In this work, we try to probe deep MRC models to estimate predictive uncertainty in a unified way. Specifically, we first try to abstract modern deep MRC models into a unified view. Based on this unified view, we show how to insert probes into MRC models to extract useful signals for uncertainty estimation. Finally, based on the learned probes, we show how to build an estimation model to produce the confidence score via supervised learning.

5.1 A Unified View for Deep MRC Models

Many studies on MRC formulate the QA task as an answer span prediction problem. Given a (question) query[4] and a passage, an MRC model extract an answer (i.e., a segment of text in the passage) if and only if the passage contains an answer. When the passage does not contain any answer, the model returns *null*.

Formally, the passage and the query are described as a sequence of word tokens, denoted as $p = \{w_i^p\}_{i=1}^{l_p}$ and $q = \{w_j^q\}_{j=1}^{l_q}$ respectively, where l_p is the passage length and l_q is the query length. Then the extracted answer could be denoted as $a = \{w_i^p\}_{i=s}^{e}$, where s and e denotes the start and end position of the answer span respectively. If there is no answer, s and e would point to a null position. In this way, the learning objective of MRC becomes to learn a model f to maximize the log-likelihood

$$\log Pr(a|q, p) = \log Pr(s, e|q, p). \tag{8}$$

Then in prediction,

$$\hat{s}, \hat{e} = \text{argmax}_{s,e \in \mathcal{L}(p)} Pr(\hat{s}, \hat{e}|q, p), \tag{9}$$

where $\mathcal{L}(p)$ denotes all the possible start-end position pairs in p.

With the development of deep learning techniques, most work on MRC implements f by deep neural networks and continuously refreshes the state-of-the-art performance. Different complicated architectures have been designed for deep MRC models, typically including components like embedding, sequential encoding, convolution, self-attention, co-attention and so on. However, to design a qualify model that can be applied on these different deep MRC models, we need a unified view over these existing models.

In this work, we find that although existing deep MRC models on answer span prediction have quite different architectures, they all can be abstracted as the following process shown in Figure 2:

1. Convert the text sequence q and p into their initial embedding representations $Q^{(0)} \in R^{l_q \times h_d^{(0)}}$ and $P^{(0)} \in R^{l_p \times h_d^{(0)}}$, where $h_d^{(0)}$ denotes the embedding dimension in the first layer;
2. Distill the passage representation with a series of steps $P^{(1)}, P^{(1)}, \ldots, P^{(T)}$ through a variety of complicated interactions (e.g., self-attention, inter-attention and so on);
3. Compute the start and end position vector $\vec{s}$ and $\vec{e}$ based on the final distilled representation $P^{(T)}$ and decode the optimal span $[s, e]$ according to Equation (9).

MRC model
的抽象
统一形式

Based on the above unified view, we find that deep MRC models in essence are about distilling useful representations of the passage for distinguishing the start and end position of the answer span. Such a common process inspires us to look into the intermediate distilled representations to extract useful signals for predictive uncertainty estimation, leading to the following probe-based method.

5.2 Probing MRC Models

The original probe idea proposed by Alain and Bengio [1] was to monitor the features at every layer of a model and measure how suitable they are for classification. In this work, we borrow the idea to investigate each distilled passage representation in a deep MRC model and measure how likely they are for distinguishing the start and end position of the answer span.

Specifically, given the t-th layer passage representation $P^{(t)}$, for each word position, we apply a linear layer over its latent representation to obtain a start and end score respectively, and then normalize the scores from all the positions through softmax to obtain the final start and end vector respectively,

第t层的表达

$$\vec{s}^{(t)} = softmax(P^{(t)}\mathbf{v}_s^{(t)}), \tag{10}$$

$$\vec{e}^{(t)} = softmax(P^{(t)}\mathbf{v}_e^{(t)}), \tag{11}$$

第t层的 probe parameters.

where $\mathbf{v}_s^{(t)}$ and $\mathbf{v}_e^{(t)}$ are probe parameters for the t-th layer. Note here we use a linear probe due to its convexity property [1]. We can avoid the issue of local minima since training a linear probe using softmax cross-entropy is a convex problem. Each value in $\vec{s}^{(t)}$ and $\vec{e}^{(t)}$ actually denotes the possibility of a position to be the start and end position of the answer span, respectively. Therefore, the probe parameters can be learned so as to minimize the cross-entropy loss which is usually used for the last layer of the MRC model,

$$loss_{probe}^{(t)} = -log(\vec{s}_s^{t}) - log(\vec{e}_e^{(t)}). \tag{12}$$

Note that the probes do not affect the learning and prediction of the deep MRC model. They only measure the level of discrimination of the passage representation at a given layer.

With the learned probes, for each query-passage instance and a given MRC model, we can get a series of signals $\{(\vec{s}^{(t)}, \vec{e}^{(t)})\}_{t=1}^{T}$. The signals are essential for the following uncertainty estimation.

5.3 Predictive Uncertainty Estimation

For a given MRC model and a query-passage instance, the predictive uncertainty estimation takes the probed signals above as the input to produce a confidence score within the range [0, 1]. Here we

[4] To avoid confusion on the terminology, we view a question as a query so that we can use the term query for consistency.

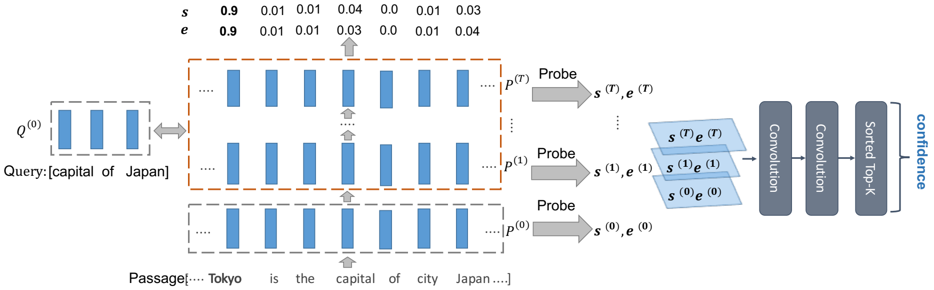


Figure 2: The architecture of the deep qualify model.

build a deep model, namely PROBE-CNN, to achieve this purpose as shown in Figure 2. Specifically, we concatenate the start and end vectors in each layer into a feature matrix. We stack these feature matrices layer by layer from bottom to top. We then apply two convolution layers and a sorted top-k layer over the stacked representations. Finally, we feed the top-k signals into a fully connected layer to produce the final confidence score.

The underlying design idea of the above PROBE-CNN model is as follows. We view the probed signals from each layer of the MRC model as a "X-ray" picture of the distilled passage representation in that layer. Each X-ray picture indicates how likely that particular distilled passage representation is for distinguishing the start and end position of the answer span. With these X-ray pictures in hand, we try to contrast 对比 and summarize them to diagnose 确定 the predictive uncertainty, which is achieved by the convolution process. We plug 插入 in a sorted top-k layer since we consider the uncertainty more related to the signal distribution rather than position patterns. In our expectation, if the probed signals are unstable between layers or ??? distributed uniformly, there will be large uncertainty in prediction. We will verify this in our experiments.

The deep qualify model is learned on a validation set for generalization. Specifically, [we first train a MRC model together with the probes on the training set. We then apply the learned MRC model and the probes on the validation set to obtain the prediction labels shown in Table 1 as well as the probed signals. Finally, we train the qualify model based on these labels and probed signals according to the objective function as shown in Equation (6).]

6 EXPERIMENTS

In this section, we conduct experiments to demonstrate the effectiveness of our risk control framework. We first introduce the the experimental settings, including datasets, basic MRC models, baseline methods and evaluation metrics. We then present the major comparison results on the benchmark collections. Furthermore, we also provide detailed analysis over the core component in our framework, i.e., the deep qualify model.

6.1 Datasets

We conduct experiments on two publicly available datasets. The detailed statistics of these two datasets are shown in Table 3.

SogouRC [5] is a large scale Web QA dataset released by the Chinese commercial search engine Sogou. It includes 30, 000 queries selected from search logs that can be satisfied by short answers, and the corresponding top ranked passages from the search result. After pre-processing, we obtain 299, 429 query-passage pairs, among which around 40% are answerable instances, and the rest are unanswerable instances.

Table 3: Dataset statistics. # denotes number, $|\text{len}_p|$ denotes average query length, $|\text{len}_q|$ denotes the average passage length, and null% denotes the proportion of unanswerable instances.

	#train	#dev	#test	$\|\text{len}_p\|$	$\|\text{len}_q\|$	null%
SogouRC	200,000	49,566	49,863	80.47	9.66	60%
SQuAD 1.0	72,599	15,000	10,570	116.98	10.2	0
SQuAD 2.0	115,319	15,000	11,873	127.90	10.02	35%

SQuAD [34] is a large scale MRC datasets with two versions. SQuAD 1.0 only contains answerable questions. SQuAD 2.0 combines the SQuAD 1.0 dataset with unanswerable questions. Although SQuADs are not typical Web QA datasets from search engines, here we use them for experiments due to the following reasons: 1) SQuAD 2.0 contains both answerable and unanswerable queries which are very similar to the Web QA scenario; 2) SQuADs have been widely adopted in MRC related research and our work addresses the predictive uncertainty of MRC.

6.2 Basic MRC models

As described in the section 4, our risk control framework can be applied to a variety of existing deep MRC models. Here we take two representative deep MRC models BIDAF [37] and BERT [7] as the basic MRC model in our framework.

BIDAF is a RNN-based MRC model. It first maps each word in the passage and query to the embedding space by combining character-level and word-level embeddings. LSTM layers with attention are then applied to collect contextual information, update the passage representation with respect to the query representation, and distill the representation of the passage. Finally, two vectors indicating the distribution of the start and the end index are derived from the final passage representation. Note that BIDAF model was originally designed only for answerable queries. To handle unanswerable queries in Web QA, we use the updated BIDAF model [26] that can handle null answers [6].

[5] http://huodong.sogou.com/cips-sogou_qa/

[6] We use the implementation: https://bit.ly/2rDHBgY

Table 4: Main results of two MRC models under our framework. The number in the parentheses denotes the relative improvement of the model against PROBA.

		SogouRC			SQuAD		
		AURC	ROC	AP	AURC	ROC	AP
BIDAF	PROBA	21.89	73.44	61.62	32.68	65.65	58.43
	AES	21.75(0.6%)	73.51(0.1%)	60.53(1.8%)	32.47(0.6%)	66.46(1.2%)	59.13(1.2%)
	ENS	21.18(3.2%)	74.71(1.7%)	62.53(1.5%)	31.35(4.1%)	67.02(2.1%)	60.89(4.2%)
	PROBE-CNN	**19.53**(10.8%)	**76.39**(4.0%)	**63.8**(3.5%)	**31.09**(4.9%)	**67.41**(2.7%)	**60.98**(4.4%)
	oracle	7.9	100.0	100.0	11.01	100.0	100.0
BERT	PROBA	15.28	74.21	52.19	27.02	64.32	51.83
	AES	14.95(2.2%)	74.36(0.2%)	52.35(0.3%)	26.4(2.3%)	66.19(2.9%)	52.25(0.8%)
	ENS	14.58(4.6%)	74.94(1.0%)	53.3(2.1%)	23.99(11.2%)	68.58(6.6%)	55.22(6.5%)
	PROBE-CNN	**14.16**(7.3%)	**75.46**(1.7%)	**53.8**(3.1%)	**23.41**(13.4%)	**69.55**(8.1%)	**56.47**(9.0%)
	oracle	5.23	100.0	100.0	7.92	100.0	100.0

BERT [7] is a universal language representation model and can be used for many NLP tasks. The main structure of BERT is a multi-layer transformer encoder. Specifically, for MRC, it first maps each word in the query and the passage its word embedding, position embedding and segment embedding. It then interacts the query and the passage to distill the passage representation. Based on the final-layer representation of the passage, the start and end vectors can be derived by a single linear layer to locate the answer in the passage. For recognizing unanswerable queries, BERT places a special token in the passage to indicate the null answer.

6.3 Baselines

Since the core component in our risk control framework is the qualify model, here we consider several existing predictive uncertainty estimation methods as our major baselines, including manually designed functions and ensemble-based methods.

PROBA [18] simply employs a heuristic *max* function on the normalized output probability of a model as the uncertainty estimation. Specifically, for deep MRC models, PROBA takes the max probability of spans in the passage

$$max(softmax(\vec{s}) \otimes softmax(\vec{e}))$$

as the confidence score. This is the most intuitive method for estimating the confidence in MRC.

集成

AES [16] makes use of the ensemble idea which averages model predictions from different epochs to estimate the predictive uncertainty. To adapt this method to deep MRC models, we train basic MRC models and save the model snapshot in each learning epoch. Note that it takes 20 epochs for BIDAF and 3 epochs for BERT to reach convergence. We then average the predicted start and end vectors from different model snapshots, and compute max probability as PROBA to obtain the final confidence score.

ENS [25] trains multiple models from different initialization to estimate the predictive uncertainty of deep models on the image classification task. To adapt this model to the Web QA scenario, we train deep MRC models multiple times with different initialization. We then average the predicted start and end vectors, and compute the max span probability as the confidence score. We tried different initialization numbers and find that with 3 randomly initialized models we can achieve good effectiveness-efficiency trade-off. Further increase on the initialization number brings little gain with much larger computational cost.

6.4 Evaluation Metrics

For evaluation, we introduce some widely adopted metrics in previous work on predictive uncertainty estimation [16, 25] for risk-aware Web QA evaluation.

Firstly, the performance of the risk control framework could be qualified using *risk* and *coverage*. The coverage is the probability mass of the non-rejected region of the confidence score,

Coverage:

$$\hat{\phi}(f|S_m) \triangleq \frac{1}{m}\sum_{i=1}^{m} h(g). \tag{13}$$

越高越好

The risk is defined as the empirical loss in the non-rejected region

risk:

$$\hat{r}(f,h,g|S_m) \triangleq \frac{\frac{1}{m}\sum_{i=1}^{m} \ell(f(q_i,p_i),a_i)h(g)}{\hat{\phi}(f|S_m)}, = \frac{\sum_{i=1}^{m} l(f(q_i,p_i),a_i)h(g)}{\sum_{i=1}^{m} h(g)}$$

越低越好

which is the same as the empirical selective risk defined in Equation (7). These two measures can be empirically evaluated over any finite labeled set S_m (not necessarily the training set) in a straightforward manner whenever we choose a specific threshold parameter θ in the decision model. Moreover, these two metrics are actually trade-off to each other in the sense that if we increase θ, we will often observe worse (i.e., lower) coverage with better (i.e., lower) risk. Therefore, we can draw the *risk-coverage curve* (RC-curve) to show the trade-off relation.

Now we introduce the metric *area under the (empirical) RC curve* (AURC) as an overall performance measure for the risk control framework that is free of the specific selection of the threshold parameter θ.

怎么计算 ???

$$AURC(f,g|V_n) = \frac{1}{n}\sum_{h\in\mathcal{H}} \hat{r}(f,h,g|S_m). \tag{14}$$

AURC ↓, 风险控制框越好.

The smaller the AURC, the better the risk control framework is. Note that we can not only use the AURC to evaluate different risk control framework based on the same basic MRC model, but also compare them across different MRC models since AURC also reflects the effectiveness of the MRC model itself (when coverage equals 1).

Besides, we can also view the risk control framework as a filter which filters out the incorrect predictions of an MRC model. In this

[7] We use the implementation: https://bit.ly/2B4TjGi

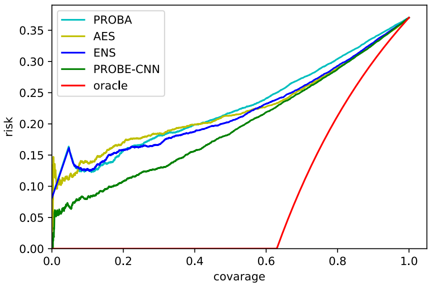


Figure 3: Risk-coverage curve of BIDAF on SogouRC.

way, the risk control framework can be view as a binary classifier and we can measure its performance via *the area under the ROC curve* (ROC) and *the average precision* (AP) metrics. The difference of these two metrics from the AURC is that they are independent of the MRC model. They only measure the performance of the qualify model. Opposite to the AURC, the larger ROC or AP, the better the qualify model is.

6.5 Main Comparison

We present the main comparison results of the risk control framework using different qualify models in Table 4. In the table, we also show the *oracle* result, which refers to a perfect risk control framework that can correctly identify instances in AD$^-$ and UD from those in AD$^+$. In this way, the ROC and AP of the oracle is always 1, while the AURC of the oracle can reflect the inherent ability of the MRC model as well as the dataset characteristic.

From the results, we have the following observations: (1) BERT can always obtain lower AURC score than BIDAF on the same dataset in the oracle mode, which indicates higher effectiveness of BERT over BIDAF on answer extraction. (2) Using the same basic MRC model, SogouRC always obtains lower AURC score than SQuAD 2.0 in the oracle mode, which indicates the SogouRC dataset is easier than the SQuAD 2.0. This is somehow counter-intuitive to us since the SogouRC dataset is an open domain Web QA dataset collected from the real-world search engine. However, after careful investigation, we find that SQuAD 2.0 is actually more difficult than SogouRC. The reason is that manually designed unanswerable queries in SQuAD 2.0 are much more difficult to identify (since the queries are often highly related to the passages) than those real-world unanswerable queries (which are unanswerable often due to the lack of related passages). (3) The ensemble-based methods, including AES and ENS, obtain better results than the manually designed PROBA method in terms of all the evaluation metrics. The results show that by using the average of multiple model variants (either from different epochs or from different initializations), we can obtain better predictive uncertainty estimation. However, ensemble-based methods usually require large computational cost which may not well fit the online requirement. (4) Our proposed PROBE-CNN model outperformd all baseline methods consistently in terms of all the evaluation metrics. For example, on the SogouRC dataset, the relative improvement of our method over the best-performing baseline ENS is about 7.7% in terms of AURC. It is worth to note that our method is orthogonal to the ensemble-based method, in the sense that we can further include ensemble-based idea into our framework to improve the risk control performance. We will leave this as our future work.

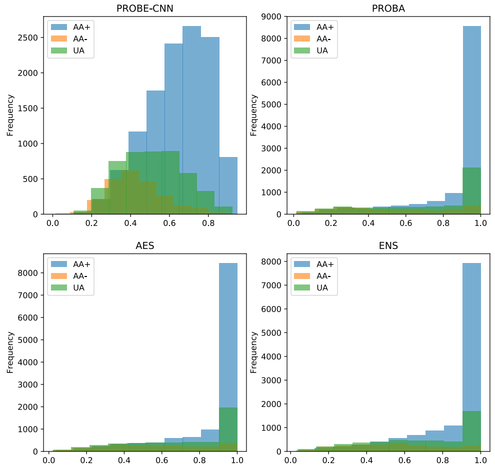


Figure 4: Confidence histogram of BIDAF on SogouRC.

6.6 Detailed Comparison

Beyond these overall performance comparison, we further take some detailed analysis. Here we draw the RC-curve of the four methods as well as the oracle on the SogouRC dataset using the BIDAF as the basic MRC model, as shown in Figure 3. We can see the ideal RC-curve is the red line from the oracle. When we increase the coverage (by lowering the confidence threshold θ), an ideal risk control framework should first keep the risk at 0 upto certain coverage, and then increase steadily until the coverage reaches 1. Such curve means that the risk control framework can correctly rank the instances based on the predicted confidence score so that there is a rejection region that no risk would happen within it. Among the four methods, we can observe that the RC-curve of our PROBE-CNN method is always under all the other curves and also the most close to the oracle curve, showing that our method can consistently predict better confidence score for all the instances.

We also depicted the histogram of predicted confidence scores from the four methods over the three set of instances, i.e., AD$^+$, AD$^-$ and UD, to obtain better understanding. The results are shown in Figure 4. Ideally, we expect a good risk control framework could distinguish these instances clearly and give instances in the AD$^+$ set higher scores than those in the AD$^-$ and UD set. From the results, we can observe that the three baseline methods cannot distinguish them very well as both AD$^+$ and UD have their peak value on the right. Our method can better distinguish these instances in terms of the score distribution, although UD set is still the most difficult one to deal with. This is probably due to that there might be no clear/stable pattern in the UD instances that could be captured by our probes.

Since the UD instances are the most difficult to handle in the risk control framework, some people might wonder whether our method gain the improvement by just better recognizing those UD instances from the rest. Here we conduct a further analysis by comparing all the four risk control methods only on the answerable queries. We use the answerable part in SoguoRC (Named SogouRC

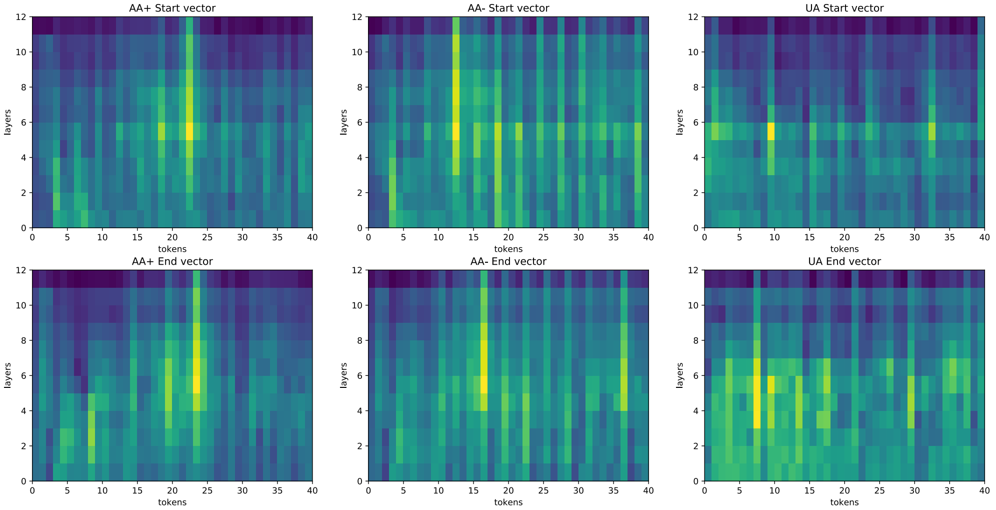


Figure 5: Heatmap of the probed signals of three sampled cases from the AD$^+$, AD$^-$ and UD set using BERT on SogouRC .

Table 5: Comparison results on the answerable datasets.

		SogouRC Answerable			SQuAD 1.0		
		AURC	ROC	AP	AURC	ROC	AP
BIDAF	PROBA	7.38	83.06	55.15	8.37	80.29	52.42
	AES	7.20	83.36	55.92	7.96	81.5	54.10
	ENS	7.06	84.2	56.18	7.90	81.63	54.40
	Ours	6.9	84.71	56.85	7.8	81.72	53.88
BERT	PROBA	3.22	85.49	47.57	5.24	78.22	37.74
	AES	3.03	86.34	47.56	5.10	79.28	44.38
	ENS	2.78	87.50	47.99	4.20	83.9	46.90
	Ours	2.66	88.03	48.93	3.63	85.53	49.26

Answerable) and SQuAD 1.0 dataset for experiments. The results are shown in Table 5. From the results, we can see that our method can also consistently outperform all the baseline methods in terms of all the evaluation metrics. The results indicate that our PROBE-CNN can well control the risk via distinguishing AD$^+$ from AD$^-$.

6.7 Analysis on the Probe

We further conduct experiments to analyze our probe-based qualify model. We first investigate the effect of layer numbers to our PROBE-CNN method. In our original design, we probe each intermediate layer of the MRC model to obtain signals for predictive uncertainty estimation. Here we test the performance if we only probe the last layer $P^{(T)}$ of the MRC model, namely PROBE-CNN$_{P(T)}$. The results are shown in Table 6. We can see that there is a clear performance drop if we only probe the last layer. The relative decrease of PROBE-CNN$_{PT}$ over the vanilla PROBE-CNN is about 3% in terms of AURC on the SogouRC dataset. The results demonstrate that the last layer does contain some useful signals, e.g., the distribution of the final s and e scores, but the intermediate layers can bring much richer information for uncertainty estimation.

Furthermore, We plot the outputs of the probes to gain some intuitive understanding on what the probes have learned. We sample three cases from the AD$^+$, AD$^-$ and UD set using the BERT model respectively and show their probed start and end vector signals in Figure 5. Note the vertical axis denotes the layer number and the horizontal axis denotes the word index in the passage, and the brighter the signal, the larger value it is. From the figure we can see that the correct instance (from the AD$^+$ set) shows a distinguishable position consistently from bottom to top. While for the incorrect instances (from the AD$^-$ and UD set), there are often multiple indistinguishable positions with strong signals and the patterns are varying from layer to layer. These probed patterns cope well with our intuition and bring good interpretability to our probe-based qualify model.

Table 6: Comparison results over the probe number.

		AURC	ROC	AP
SogouRC	PROBE-CNN$_{P(T)}$	14.60	75.12	53.1
	PROBE-CNN	14.16(3.0%)	75.46	53.8
SQuAD	PROBE-CNN$_{P(T)}$	24.39	68.52	55.30
	PROBE-CNN	23.41(4.0%)	69.55	56.47

7 CONCLUSIONS

In this paper, we introduced the risk control problem of Web QA by modeling the predictive uncertainty of deep MRC models. We conducted an in-depth analysis of the risk of Web QA. Based on the analysis, we proposed a risk control framework with a novel and general deep qualify model designed based on the abstraction of modern MRC models. We conducted extensive experiments on publicly available benchmark datesets using risk-aware evaluation metrics. The empirical results demonstrate the effectiveness and show the good interpretability of our proposed method. For the future work, we may apply our proposed risk control framework to other IR tasks, e.g., query suggestion.

ACKNOWLEDGMENTS

This work was funded by the National Natural Science Foundation of China (NSFC) under Grants No. 61425016, 61722211, 61773362, and 61872338, the Youth Innovation Promotion Association CAS under Grants No. 20144310, and 2016102, the National Key R&D

Program of China under Grants No. 2016QY02D0405, and the Foundation and Frontier Research Key Program of Chongqing Science and Technology Commission (No. cstc2017jcyjBX0059).

REFERENCES

[1] Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644* (2016).

[2] Yuval Bahat and Gregory Shakhnarovich. 2018. Confidence from Invariance to Image Transformations. *arXiv preprint arXiv:1804.00657* (2018).

[3] John Blatz, Erin Fitzgerald, George F Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. (2004), 315.

[4] Eric Brill, Susan Dumais, and Michele Banko. 2002. An analysis of the AskMSR question-answering system. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 257–264.

[5] Danqi Chen. 2018. *Neural Reading Comprehension and Beyond*. Ph.D. Dissertation. Stanford University.

[6] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1870–1879.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[8] Akshay Raj Dhamija, Manuel Günther, and Terrance Boult. 2018. Reducing Network Agnostophobia. In *Advances in Neural Information Processing Systems*. 9175–9186.

[9] Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. 2017. Quasar: Datasets for Question Answering by Search and Reading. *arXiv preprint arXiv:1707.03904* (2017).

[10] Li Dong, Chris Quirk, and Mirella Lapata. 2018. Confidence Modeling for Neural Semantic Parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. 743–753.

[11] Matthew Dunn, Levent Sagun, Mike Higgins, Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179* (2017).

[12] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building Watson: An overview of the DeepQA project. *AI magazine* 31, 3 (2010), 59–79.

[13] Yarin Gal. 2016. Uncertainty in deep learning. *University of Cambridge* (2016).

[14] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*. 1050–1059.

[15] Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. In *Advances in neural information processing systems*. 4878–4887.

[16] Yonatan Geifman, Guy Uziel, and Ran El-Yaniv. 2019. Bias-Reduced Uncertainty Estimation for Deep Neural Classifiers. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SJfb5jCqKm

[17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On Calibration of Modern Neural Networks. *international conference on machine learning* (2017), 1321–1330.

[18] Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136* (2016).

[19] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*. 1693–1701.

[20] Minghao Hu, Yuxing Peng, Zhen Huang, Nan Yang, Ming Zhou, et al. 2018. Read+ verify: Machine reading comprehension with unanswerable questions. *arXiv preprint arXiv:1808.05759* (2018).

[21] Minghao Hu, Yuxing Peng, and Xipeng Qiu. 2017. Mnemonic reader for machine comprehension. *arXiv preprint arXiv:1705.02798* (2017).

[22] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1601–1611.

[23] Souvik Kundu and Hwee Tou Ng. 2018. A Nil-Aware Answer Extraction Framework for Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4243–4252.

[24] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683* (2017).

[25] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*. 6402–6413.

[26] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke S Zettlemoyer. 2017. Zero-Shot Relation Extraction via Reading Comprehension. *conference on computational natural language learning* (2017), 333–342.

[27] Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. 2018. Denoising distantly supervised open-domain question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1736–1745.

[28] David JC MacKay. 1992. A practical Bayesian framework for backpropagation networks. *Neural computation* 4, 3 (1992), 448–472.

[29] Andrey Malinin and Mark Gales. 2018. Predictive Uncertainty Estimation via Prior Networks. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 7047–7058. http://papers.nips.cc/paper/7936-predictive-uncertainty-estimation-via-prior-networks.pdf

[30] Amit Mandelbaum and Daphna Weinshall. 2017. Distance-based Confidence Score for Neural Network Classifiers. *arXiv preprint arXiv:1709.09844* (2017).

[31] Dan Moldovan, Sanda Harabagiu, Marius Pasca, Rada Mihalcea, Roxana Girju, Richard Goodrum, and Vasile Rus. 2000. The structure and performance of an open-domain question answering system. 563–570.

[32] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* (2016).

[33] Kyosuke Nishida, Itsumi Saito, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2018. Retrieve-and-read: Multi-task learning of information retrieval and reading comprehension. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 647–656.

[34] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*. 784–789.

[35] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2383–2392.

[36] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 193–203.

[37] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603* (2016).

[38] Chuanqi Tan, Furu Wei, Qingyu Zhou, Nan Yang, Weifeng Lv, and Ming Zhou. 2018. I Know There Is No Answer: Modeling Answer Validation for Machine Reading Comprehension. In *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 85–97.

[39] Yi Tay, Anh Tuan Luu, Siu Cheung Hui, and Jian Su. 2018. Densely Connected Attention Propagation for Reading Comprehension. In *Advances in Neural Information Processing Systems*. 4911–4922.

[40] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer Networks. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 2692–2700.

[41] Ellen M Voorhees and Dawn M Tice. 1999. The TREC-8 Question Answering Track Evaluation.. In *TREC*, Vol. 1999. 82.

[42] Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905* (2016).

[43] Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerald Tesauro, Bowen Zhou, and Jing Jiang. 2017. R3: Reinforced Reader-Ranker for Open-Domain Question Answering. *arXiv preprint arXiv:1709.00023* (2017).

[44] Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2018. Evidence Aggregation for Answer Re-Ranking in Open-Domain Question Answering. In *International Conference on Learning Representations*.

[45] Wei Wang, Ming Yan, and Chen Wu. 2018. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. *arXiv preprint arXiv:1811.11934* (2018).

[46] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 189–198.

[47] Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604* (2016).