24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Combination of Active Learning and Self-Paced Learning for Deep Answer Selection with Bayesian Neural Network

Qunbo Wang[1] and **Wenjun Wu**[2] and **Yuxing Qi**[3] and **Zhimin Xin**[4]

Abstract. Answer Selection is an important subtask of Question Answering tasks. For this learning-to-rank problem, deep learning methods have outperformed traditional methods. To train a high-quality deep answer selection model, it often requires large amounts of labeled data, which is a costly and noise-prone process. Active learning and semi-supervised learning are usually applied in the modelling training procedure to achieve optimal accuracy with fewer labeled training samples. However, traditional active learning methods rely on good uncertainty estimates that are hard to obtain with standard neural networks. And the performance of semi-supervised learning methods are always affected adversely by the quality of the pseudo-labeled data. In this work, we propose a new framework integrating active learning and self-paced learning in training deep answer selection models. This framework proposes an uncertainty quantification method based on Bayesian neural network, which can guide active learning and self-paced learning in the same iterative process of model training. Experiments were conducted on two kinds of deep answer selection models with real-world datasets including YahooCQA and SemiEvalCQA. The results reveal that the proposed method can significantly reduce the labeled samples for model training.

1 INTRODUCTION

Answer Selection is an important subtask of Question Answering, whose goal is to retrieve the best answers to a given question from a set of candidate answers [19]. Especially, on the community question answering site such as Stack Overflow, Yahoo! Answers, it is important to facilitate users to locate the best answers to their questions because the quality of answers varies largely due to the diverse attitude and ability of different community members. Therefore, the high-quality answers are the important resources to generate useful question-answer pairs, which are of great value for information retrieval [35]. Various efforts have been dedicated to developing automated tools to rank and identify candidate answers [32, 25, 37].

Obviously, Answer Selection is a learning-to-rank problem by nature, in which deep learning methods have outperformed traditional methods recently [32, 37]. But these deep learning models are difficult to be adopted in practical scenarios because to train these models requires large amounts of labeled data. For the community QA sites that can produce large volumes of data every day [38], it is very appealing to train deep learning models with few human annotations, given their massive amounts of unlabeled data. Currently, the goal of active learning (AL) and semi-supervised learning are to enable a machine learning model to achieve greater accuracy with fewer labeled training samples, which are the two categories of learning methods with the opposite criteria [23].

The main idea of active learning [6, 31] is to progressively select and annotate most informative unlabeled samples to boost the model. Recent research efforts on deep active learning are designed for image classification [10, 30, 29] and named entity recognition [34, 33], etc. None of the works have tackled with the problem of learning-to-rank, which is often considered as more difficult tasks than image and text classification. On the other hand, the semi-supervised learning takes a different approach by exploiting unlabeled samples to improve the model performance. The self-training is a commonly used semi-supervised learning technique for the answer selection task, in which the unlabeled data are exploited to find reliable samples that can be pseudo-labeled by the learning model. These selected new data can be further added to the training dataset in an iterative training process. The major problem of the simple self-training method is caused by the pre-maturity of the initial model because noisy pseudo-labeled samples may significantly reduce learning performance [20]. Recently, the self-paced learning (SPL) [18] is introduced to address this problem of self-training [5, 28, 39], which evolved from the curriculum learning [3] biologically inspired by the common human process of gradual learning. In self-training scenario, SPL gradually incorporates pseudo-labeled samples with high reliability for training to avoid the compromise of ambiguous labeling and poor local minima in model training [16].

In fact, both learning approaches rely upon good uncertainty quantification mechanisms. But it is hard to design a suitable uncertainty quantification mechanism in answer selection tasks that need considering the structure and dependence relationship in each sample. Therefore, for the deep answer selection model, this work firstly extends the expected loss optimization with Bayesian neural network to calculate the prediction uncertainty of each unlabeled sample, which can determine informative samples in AL and reliable samples in SPL. Then, this work proposes a framework based on the combination of active learning and self-paced learning to minimize annotation costs while maximizing the desired performance. Specifically, the learning process includes multiple rounds until the model achieves to the optimal performance. In each round, the active learning method can query the label for the low confidence samples those minimize the model uncertainty, while the self-paced learning method can exploit the remaining unlabeled data to make the model more solid. Both manually-labeled and pseudo-labeled samples are added to the training data to retrain the deep learning model. Experiments were conducted on two kinds of deep answer selection models with real-world datasets including YahooCQA and SemiEvalCQA. The results

[1] BEIHANG University, China, email: wangqunbo@nlsde.buaa.edu.cn
[2] BEIHANG University, China, email: wwj@nlsde.buaa.edu.cn
[3] BEIHANG University, China, email: qiyuxing@nlsde.buaa.edu.cn
[4] BEIHANG University, China, email: xinzhimin@nlsde.buaa.edu.cn

reveal that the proposed method can significantly reduce the labeled samples for model training. In summary, this paper makes the following contributions:

- We propose a new uncertainty quantification method for the deep learning model of learning-to-rank. This method extends the expected loss optimization with Bayesian neural network for the answer-selection tasks.
- We study how to integrate both active learning and self-paced learning in training a deep answer selection model, which can minimize the annotation labor to achieve the optimal performance of the model.
- We conduct a thorough evaluation on two deep answer selection models with real-world datasets to prove the effectiveness of the proposed method.

The remainder of this paper is structured as follows: In Section 2, we present previous related work. And we describe the deep answer selection model in Section 3. Then we introduce the proposed framework in Section 4. Using the YahooCQA and SemiEvalCQA, we make an evaluation of the proposed method in Section 5. In Section 6, we present our conclusions and directions for future work.

2 RELATED WORK

2.1 Active learning and semi-supervised learning

The goal of active learning is reducing required annotations for training machine learning models, which focus on actively selecting and annotating the most informative unlabeled samples to avoid unnecessary and redundant annotations [31]. Active learning starts with a small labeled dataset for training the initial model, and a large pool of unlabeled data. It needs to run multiple rounds of data labelling and model retraining. In each round, based on the trained model, an acquisition function is implemented to choose informative samples from unlabeled data to be annotated up, which are used to augment the labeled dataset [6].

In fact, designing active learning algorithm on ranking tasks poses more challenges because learning-to-rank is quite different from classification and regression tasks due to its unique query-document structure. And it has received significant amount of attention from the research community [4, 22, 24]. However, the previous research efforts about learning-to-rank are all based on the traditional machine learning models, such as Gradient Boosting Decision Tree(GBDT) [8], SVM [7], etc. These methods can not be directly applied to deep learning because they require training an ensemble of learners to estimate the uncertainty, which becomes computationally expensive for deep learning models. Recently, Gal and Ghahramani [9] show an equivalence between approximate Bayesian inference and dropout, which enables the application of Bayesian methods to obtain the prediction of uncertainties over neural networks. Based on this method, AL algorithms can effectively query informative unlabeled samples for deep learning models [17]. And many research efforts [10, 34, 33] attempt to develop an active learning framework based on Bayesian neural networks. But none of the works have tackled with the uncertainty estimate of deep learning models for learning-to-rank tasks.

On the other hand, active learning only emphasizes utilizing low-confidence samples to reduce labelling budget without considering the potential use of the other high-confidence samples. In fact, semi-supervised learning can explore the unsupervised information beyond labeled data to train robust models [27]. In this setting, self-training is a practical approach that is more suitable for answer selection tasks. The approach of self-training was first presented by Nigam et. al. [26] and it was shown that it can be interpreted as an instance of the Classification Expectation Maximization algorithm [2]. Recently, B Hyams et al. [14] has proposed a semi-supervised deep learning method for the image classification task, in which the samples with highest predicted class probability are assigned pseudo-labels for training. It employs dropout in Bayesian deep learning to obtain prediction distribution over the deep learning model, and substantially improves learning efficiency .

2.2 Combination of active learning and semi-supervised learning

It is nature to integrate active learning and semi-supervised learning in the same model training procedure. For example, for synthetic aperture radar image recognition, Gao et al. [11] presented an active semi-supervised learning approach that accepts some pseudo-labels with highest confidence in every iteration and asks an oracle for a chosen number of samples with lowest confidence. The major limitation of his method lies in the measurement of label confidence with classification probability, which is not suitable for deep learning models. With the advantages of uncertainty modeling with Bayesian neural network model over network activations and predictions [17], Rottmann et al. [23] proposed a deep Bayesian active semi-supervised learning approach for classification tasks paired with an active learning component and approximate Bayesian uncertainty. Note that such an approach is only designed for the simple classification tasks, thus cannot be applied in ranking tasks that we aim to address in this paper.

The above efforts adopted the pseudo-labelling method in their semi-supervised learning components have inherent problems. In its iterative training process, it only evaluates unlabeled samples and always retains the pseudo-labels generated previously. Moreover, most self-training frameworks manually set the threshold for choosing high confidence samples in their training process. But it is necessary to fine-tune such a threshold in different cases, which is not a trivial task for researchers. And given the premature model in the initial phase of semi-supervised learning, improper threshold values may allow many noisy pseudo-labeled samples to be incorporated into the training data [20]. Therefore, unavoidable classification errors of the premature model under training may result in error amplification, which adversely affect the performance of self-training.

In order to tackle with the above limitation of the current deep active semi-supervised learning, we replace the simple pseudo-labelling by self-paced learning, which gradually incorporates the high reliability unlabeled samples for pseudo-labelling. In fact, the self-paced learning is a more general implementation for curriculum learning because the curriculum in the SPL is independent of model objectives in the specific problems [15]. Recently, some research projects successfully adopt self-paced learning for semi-supervised tasks and prove that it can avoid poor local minima and noisy pseudo-label samples [5, 28, 39]. For example, Chen et al. [5] leveraged the self-paced learning in semi-supervised object detection, and their experiments show that the self-paced learning paradigm can achieve the promising accuracy with a smaller amount of labeled training data. Sangineto et al. [28] proposed a training protocol based on the self-paced learning paradigm in a weakly-supervised scenario object detector. Lin et al. [21] combined active learning and self-paced learning for SVM based facial identification. Therefore, we expect that integration of active learning and self-paced learning should effectively achieve better model accuracy and robustness against noisy samples than previous frameworks. Note that previous research work

didn't integrate active learning and self-paced learning for training deep neural networks. In this work, for the answer selection task, we study the combination of self-paced learning and active learning to train a robust deep learning model.

3 DEEP LEARNING FOR ANSWER SELECTION

Given a dataset in the answer selection task, where each sample X_i includes a question sentence q_i and a list of answer sentences $\{a_{i1}, a_{i2}, ..., a_{iJ}\}$, the answer selection task is to find the best answer candidate for each question. Therefore, the goal is to train a deep learning model that can measure the matching degree of each question-answer pair, where the answer with the highest degree is the best candidate for the corresponding question. In addition, getting the training data could be approached as a multi-labelling task for the answer selection model. The answer candidates of question are labeled with their judgements $label_i = \{l_{i1}, l_{i2}, ..., l_{iJ}\}$, where $l_{ij} = 1$ if the answer j is the best answer to the question i, and otherwise $l_{ij} = 0$. Generally, we can simply use each tuple (q_i, a_{ij}, l_{ij}) to train the answer selection model respectively.

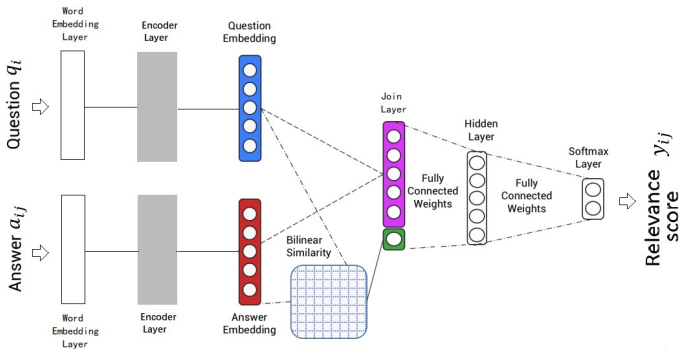


Figure 1. The neural network architecture for answer selection

In this work, we adopt two basic neural networks for the answer selection task from [32] and [36], which use CNN and LSTM as the encoder respectively. The architecture of two neural networks are shown in Figure 1. They both adopt two distributional sentence models based on the encoder to arrive at a question embedding e_{q_i} for the question q_i and an answer embedding $e_{a_{ij}}$ for the candidate answer a_{ij}, which are then used to learn the semantic similarity between them. At the join layer, all intermediate representations are concatenated into a single vector. The architecture includes an additional hidden layer right before the softmax layer to model interactions between the components of the intermediate representation. At last, the neural network gives a prediction y_{ij} ($y_{ij} \in [0, 1]$) of the relevance score between question q_i and answer a_{ij} , which means the probability that the candidate answer is correct for the question. Therefore, for all answers in the sample X_i, the model gives a prediction permutation $Y_i = \{y_{i1}, y_{i2}, ..., y_{iJ}\}$.

4 THE INTEGRATED FRAMEWORK WITH DEEP ACTIVE LEARNING AND SELF-PACED LEARNING

For answer selection task, this section describes a new framework that integrates active learning and self-paced learning to improve the performance of deep answer selection model. The workflow of the proposed framework is depicted in Figure 2. This framework runs the DASL algorithm described in Section 4.3, which incorporates uncertainty quantification, active learning and self-paced learning in the same workflow.

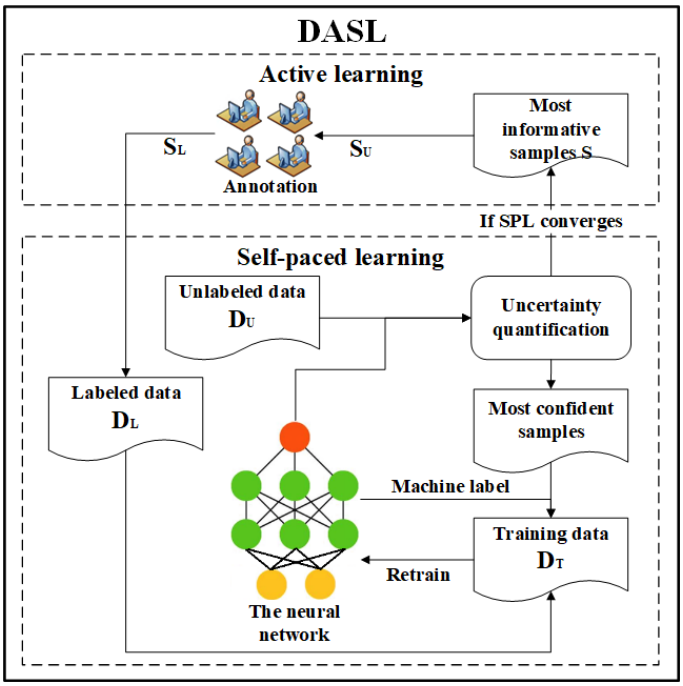


Figure 2. The workflow of the integrated framework

In the workflow of the framework, there are three datasets: the unlabelled dataset D_U, the labelled dataset D_L, and the training dataset D_T. D_U and D_L are used as the original input by the DASL algorithm. And the training dataset D_T is dynamically updated by the pseudo-labeled and manually-labeled samples from D_U and D_L with multiple rounds of the DASL algorithm. In each round, the uncertainty quantification method for answer selection tasks (as mentioned in Section 4.1) estimates the confidence of each unlabeled sample in D_U, including pseudo-labeled samples. Based on the self-paced learning paradigm, the most reliable samples are gradually labeled by the deep learning model and incorporated into D_T for retraining (as mentioned in Section 4.2). And when the model converges to the maximum performance by the self-paced learning on the validation data, the most uncertain unlabeled samples are annotated by the active learning method and incorporated into the labeled dataset D_L.

4.1 Uncertainty quantification

For the deep learning model of learning-to-rank, this section proposes an uncertainty measure method based on the expected loss optimization. Specifically, the first step is to estimate the expected error of a label in an unlabeled sample X, in which the posterior distribution $P(Y|X, D_T)$ is employed following Bayesian formalism and a loss function $L(label, Y)$ is used to estimate the error of the label in each prediction. According to Bayesian decision theory, the label

with the least expected loss should be adopted by sample X, and the expected loss is defined as follows:

$$EL(X) := \min_{label} \int_Y L(label, Y) P(Y|X, D_T) \mathrm{d}Y \tag{1}$$

Obviously, a sample with the highest expected loss will impart the greatest change to the current model, which is more uncertain for learning model. On the other hand, in answer selection tasks, given a sample X, the prediction Y is a permutation of the relevant scores between the question and all candidate answers, where a label can be viewed as a ranking permutation π. In order to calculate the loss function of a ranking permutation, we define Equation 2 and 3 by using discounted cumulative gain (DCG) to measure the quality of a rank and calculating the difference between the DCG of that permutation and the best permutation with the highest DCG.

$$L(\pi, Y) = \max_{\pi'} DCG(\pi', Y) - DCG(\pi, Y) \tag{2}$$

$$DCG(\pi, Y) = \sum_{j}^{J} \frac{2^{y_j} - 1}{\log_2 (1 + \pi(j))} \tag{3}$$

Where y_j is the relevance score of the jth answer given by the deep answer selection model. $\pi(j)$ is the position of the jth answer in the ordered permutation π. Correspondingly, the expected loss for a given sample X can be expressed in Equation 4.

$$EL(X) := \min_{\pi} \int_Y (\max_{\pi'} DCG(\pi', Y) - DCG(\pi, Y)) P(Y|X, D_T) \mathrm{d}Y \tag{4}$$

For the deep learning model, to infer the posterior distribution of the prediction, the Bayesian neural network is utilized in our method. It is used to describe a neural network as a probabilistic model $P(Y|X, \omega)$ with prior probability distributions placed over a set of model parameters $\omega = \{w_1, ...\}$:

$$\omega \sim P(\omega) \tag{5}$$

Further, given the training data $D_T = \{(X_1, label_1), ..., (X_n, label_n)\}$, Bayesian inference for neural networks can first calculate the posterior distribution of the weights $P(\omega|D_T)$, and then calculate the predictive distribution of a test data sample X.

$$P(Y|X, D_T) = \int_\omega P(Y|X, \omega) P(\omega|D_T) \mathrm{d}\omega \tag{6}$$

Because it is often infeasible to perform exact Bayesian inference in Equation 6, we must rely upon variance approximate inference technique. It often attempts to find a distribution $P^*_\theta(\omega)$ to minimize the Kullback-Leibler (KL) divergence with the true model posterior $P(\omega|D_T)$. Gal and Ghahramani [9] proved the equivalence between approximate Bayesian inference and dropout. Therefore, approximate Bayesian inference can be done according to Equation 7. It performs stochastic forward passes with dropout to sample from the approximate posterior $P^*_\theta(\omega)$, which is marginalized over the approximate posterior using Monte Carlo integration.

$$\begin{aligned} P(Y|X, D_T) &= \int_\omega P(Y|X, \omega) P(\omega|D_T) \mathrm{d}\omega \\ &\approx \int_\omega P(Y|X, \omega) P^*_\theta(\omega) \mathrm{d}\omega \\ &\approx \frac{1}{T} \sum_{t=1}^{T} P(Y|X, \hat{\omega}^t) \end{aligned} \tag{7}$$

Equation 7 formulates the predictive distribution given an input X, in which $P^*_\theta(\omega)$ represents the dropout distribution and $\hat{\omega}^t \sim P^*_\theta(\omega)$ that represents the model parameters over forward pass t with dropout [9]. Given an input in the answer selection task, the prediction is a permutation with relevance scores of candidate answers, which is determined over a forward pass. For convenience, we represent the predicted result Y over forward pass t as $Y^t = \{y^t_1, y^t_2, ..., y^t_J\}$. Therefore, substituting Equation 7 into Equation 4, the expected loss for a given X can be easily derived as Equation 8.

$$\begin{aligned} EL(X) &\approx \min_{\pi} \frac{1}{T} \sum_{t=1}^{T} (\max_{\pi'} DCG(\pi', Y^t) - DCG(\pi, Y^t)) \\ &= \frac{1}{T} \sum_{t=1}^{T} \max_{\pi'} DCG(\pi', Y^t) - \max_{\pi} \frac{1}{T} \sum_{t=1}^{T} DCG(\pi, Y^t) \end{aligned} \tag{8}$$

The first component of Equation 8 can be rewritten as Equation 9, where π^t is sorted by the prediction Y^t over forward pass t because this has the maximum DCG .

$$\begin{aligned} c1 : \frac{1}{T} \sum_{t=1}^{T} \max_{\pi'} DCG(\pi', Y^t) \\ = \frac{1}{T} \sum_{t=1}^{T} DCG(\pi^t, Y^t) \end{aligned} \tag{9}$$

The second component of Equation 8 means to find a permutation π^* with the maximum sum of DCG over all the forward passes, which can be sorted by the approximate posterior relevance score $\hat{y_j}$ of each answer for the sample X.

$$\begin{aligned} c2 : \max_{\pi} \frac{1}{T} \sum_{t=1}^{T} DCG(\pi, Y^t) \\ = \frac{1}{T} \sum_{t=1}^{T} DCG(\pi^*, Y^t) \end{aligned} \tag{10}$$

$$\hat{y_j} = \frac{1}{T} \sum_{t=1}^{T} y^t_j \tag{11}$$

In addition, for a sample X, the value of the acquisition function of active learning is calculated with the prediction permutation of relevance scores $Y = \{y_1, y_2, ..., y_J\}$. Therefore, to ensure the comparability of the value between different samples, it is necessary to normalize the relevance scores of the sample firstly.

$$Y = \{\frac{y_1}{\sum_{j=1}^{J} y_j}, \frac{y_2}{\sum_{j=1}^{J} y_j}, ..., \frac{y_J}{\sum_{j=1}^{J} y_j}\} \tag{12}$$

The details of the proposed method are described in Algorithm 1. And the sample is more uncertain for the deep answer selection model when its EL value is higher, otherwise is more confident.

4.2 Self-paced learning

Our framework adopts self-paced learning method to reduce noises in pseudo-labeled samples. It is a robust learning strategy for self-training, which involves multiple iterations for model training. In each iteration, all unlabeled samples and pseudo-labeled samples are

Algorithm 1 The uncertainty quantification algorithm

Input:
deep learning model M, unlabeled samples D_U
Output:
for each sample $X \in D_U$ **do**
  for forward pass with dropout $t = 1, ..., T$ **do**
    $y_j^t \leftarrow$ prediction of deep answer selection model M over forward pass t on the jth answer
    $Y^t = \{\frac{y_1^t}{\sum_{j=1}^J y_j^t}, \frac{y_2^t}{\sum_{j=1}^J y_j^t}, ..., \frac{y_J^t}{\sum_{j=1}^J y_j^t}\}$
    get π^t sorted by Y^t
  end for
  calculate $c1 = \frac{1}{T}\sum_{t=1}^T DCG(\pi^t, Y^t)$ following Eq. 9
  for $j = 1, ..., J$ **do**
    calculate $\hat{y_j} = \frac{1}{T}\sum_{t=1}^T y_j^t$
  end for
  get π^* sorted by $\{\hat{y_1}, \hat{y_2}, ..., \hat{y_J}\}$
  calculate $c2 = \frac{1}{T}\sum_{t=1}^T DCG(\pi^*, Y^t)$ following Eq. 10
  $EL(X) \approx c_1 - c_2$ following Eq. 8
end for
return the EL value of each sample

re-evaluated, and the most reliable samples are automatically pseudo-labeled according to the "pace" parameter, whose size is progressively increased in the subsequent iterations with the improvement of the model under training. In this iterative process, since the unlabeled samples are gradually included into the training set, it avoids incorrectly pseudo-labeled samples. Self-paced learning reformulates curriculum learning as an optimization problem by jointly modeling the curriculum and the task, which can be formulated as a general optimization problem in Equation 13:

$$\min_{w,v} \sum_{i=1}^{n} v_i L(w; X_i, Y_i) + f(v_i; \lambda) \tag{13}$$

where w represents the model parameters. $L(w; X_i, Y_i)$ denotes the cost between the objective label and the model prediction, which represents the reliability of the sample. For training data, a self-paced learning model places weighted loss terms $v = [v_1, v_2, ..., v_n]$ on all samples reflecting their importance. Meanwhile, a regularizer $f(v; \lambda)$ is imposed on sample weights, whose parameter λ controls the pace to determine which samples to be selected. Our framework adopts the negative l1-norm regularizer, which is general and applicable to various learning tasks with different loss functions [15]. On the other hand, to represent the reliability of each unlabeled sample in our case of answer selection, our framework adopts the uncertainty quantification method defined in Section 4.1. Based on the negative l1-norm regularizer and our uncertainty quantification method, the problem becomes convex and can be explicit expressed as follows in our setting.

$$v_i = \begin{cases} 1, & EL(X_i) < \lambda, \\ 0, & EL(X_i) \geq \lambda. \end{cases} \tag{14}$$

Obviously, highly reliable samples should be considered for adding into the training data in each pace. After the model is retrained with the updated training data, the model becomes more mature in the next pace. By sequentially optimizing the model with gradually increasing pace parameter λ, more samples can be automatically discovered in a pure self-paced way.

4.3 DASL algorithm

The detail of the DASL algorithm is shown in Algorithm 2. At the beginning, a small set of unlabeled data is selected randomly to be labeled as the initial labeled data. In each round, we use self-paced learning to exploit unlabeled dataset D_U firstly. Specifically, the model parameters are updated by retraining on the training dataset D_T in line 5, which includes the labeled data and the pseudo-labeled data. In line 6 and line 7, the highly reliable samples are selected through Algorithm 1, and assigned pseudo-labels with the model prediction. The self–paced learning is repeated until the model convergences to the maximum performance on the validation data, and the pace λ is updated in each iteration. Secondly, based on the trained model that achieves the best improvement with the support of self-paced learning, the acquisition function of active learning can choose the most informative samples from D_U to the predefined budget through Algorithm 1, which are annotated up and incorporated into the labeled dataset D_L. This process runs multiple rounds until it reaches the maximum round, in which pseudo-labeled samples and unlabeled samples are also re-evaluated in each round. In addition, in each round, we train from scratch to avoid possibly overfitting the data collected in earlier rounds based on observations by Hu et al. [13].

Algorithm 2 DASL algorithm

Input:
initial labeled data D_L, unlabeled data D_U, initial pace λ^0, parameter $\mu > 1$
Output:
1: **while** not reach maximum round R **do**
2: $\lambda = \lambda^0$, $v_i = 0$ for $\forall X_i \in D_U$
3: **while** not convergence **do**
4: $D_T \leftarrow D_L \cup \{(X_i, \hat{Y_i}) | v_i = 1, \forall X_i \in D_U\}$
5: update model parameters ω with D_T
6: update indicators v_i for $\forall X_i \in D_U$ following Eq. 14 and the uncertainty estimated by Algorithm 1
7: label samples $\{X_i | v_i = 1, \forall X_i \in D_U\}$ with the model prediction $\hat{Y_i}$
8: update pace $\lambda = \mu\lambda$
9: **end while**
10: query labels for most uncertainty samples S from D_U by Algorithm 1
11: $D_L \leftarrow D_L + S$
12: $D_U \leftarrow D_U - S$
13: **end while**
14: **return** model parameters ω

5 EXPERIMENTS

5.1 Experimental setup

5.1.1 Datasets

The proposed frame and DASL algorithm is evaluated in community-based question answering (CQA) datasets, including benchmark dataset YahooCQA and SemEvalCQA.

- **YahooCQA** is a large-scale and real world dataset for community-based question answering, which is collected from Yahoo Answers community. Since active learning in the DASL algorithm requires frequent retraining of the neural network model as long

as new labeled samples are acquired, it is very time-consuming to retrain the model given the large size of the Yahoo dataset. We have to select a part of all the data for experiments in order to reduce the training time. According to the method in [36], we selected question-answer pairs containing questions and best answers of length 5-50 tokens, and we constructed negative samples for each question by sampling four samples from the top 1000 hits obtained via Lucene 7 search.

- **SemEvalCQA** is a well-studied dataset for community-based question answering, which is collected from SemEval-2016 Task 3 Subtask A [1]. This is a real-world dataset obtained from Qatar Living Forums. In this dataset, the answers in each question 'thread' are marked as 'Good', 'Potentially Useful' or 'Bad'. For each question, we randomly selected a 'Good' answer as positive sample and four 'Bad' answers as negative samples.

5.1.2 Implementation details

To demonstrate that our DASL algorithm is more effective for answer selection tasks, we compare it with the random sampling and other deep active learning methods.

- **Random Sampling.** Using this acquisition function is equivalent to choosing a point uniformly at random from the unlabeled data pool.
- **Bayesian Active Learning by Disagreement (BALD).** Based on the dropout approximating method in Bayesian neural network, it boils down to choosing data points, in which each stochastic forward pass through the model would have the highest probability assigned to a different class [12, 10].
- **Core-Set.** It chooses the samples that best cover the dataset in the learned representation space [29].
- **DAL.** To analyze the efficiency of our proposed uncertainty quantification method, we implement a variant of our method DASL, which discards high-confidence sample pseudo-labelling via self-paced learning.
- **All Data.** We manually label all the training samples and use them to train the deep answer selection model.

To ensure the fairness of the experiment, all methods were assessed with two answer selection models described in Section 3, which adopt CNN and LSTM as the encoder respectively. Following the settings in the related work, we randomly selected 80% samples to form the training data, 10% samples to form the validation data and the rest were left as the testing data in our experiments. The initial labeled training data D_L was randomly determined, and the rest of the samples were used as the unlabeled training data D_U for incremental model learning process. In addition to tune the parameters of the adopted neural networks, the validation data was also used to find the optimal parameter value of the self-paced learning. All experiments were run for a fixed amount of acquisition steps, or equivalently, until a certain amount of training data was labeled. To reduce the influence of randomness, the experiment for each active learning method was repeated four times and the results were averaged, in which each forward consisted of an identical architecture but different random initialization. For Bayesian active learning, in addition to using dropout for training deep learning model, we also employed dropout to decide which data samples to acquire at training time including 100 forward passes. The dropout rate was 0.5 for neural network. Our code is at (https://github.com/BUAAw-ML/Active-Self-paced-Learning).

5.1.3 Evaluation metric

The answer selection task identifies the correct answers to the question by ranking all the candidate answers. Typically, the performance of an answer selection system is measured in Mean Reciprocal Rank(MRR) and Mean Average Precision (MAP), which are standard metrics in answer selection. In our setting, because each question has exactly one correct answer, the value of MRR and MAP are the same. Therefore, we only need to use MRR as our evaluation metric. Given a set of questions Q, MRR is calculated as Equation 15, $\pi(i)$ refers to the rank position of the correct candidate answer to the ith question.

$$MRR = \frac{1}{Q}\sum_{i=1}^{|Q|}\frac{1}{\pi(i)} \tag{15}$$

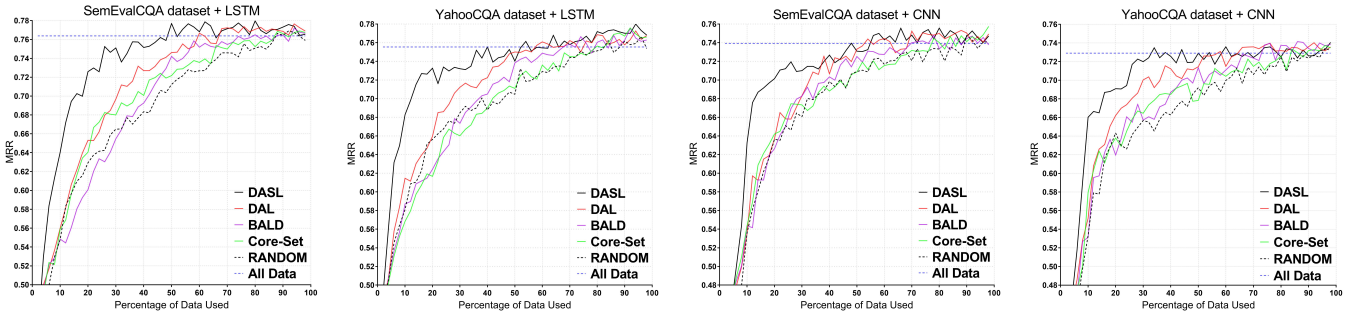
5.2 Comparison results and empirical analysis

The experiments in Figure 3 adopt the LSTM-based deep learning model and CNN-based deep learning model as the underlying model for the answer selection task respectively. They illustrate the MRR curve of different active learning methods on the YahooCQA and SemEvalCQA dataset. It is obviously that our proposed DASL method clearly outperforms other methods in any case. As shown in the Figure 3(a), random sampling outperforms BALD in early rounds of experiments for the LSTM-based deep learning model. This phenomenon may be caused by the sample-selection strategy of BALD, which is the most common method in deep active learning but is not very suitable for ranking problem. From all the experimental results, we find that our DASL outperforms the compared methods in a clear margin given the same percentage of annotated samples, especially when the percentage of annotated samples is low. We also find that DAL outperforms other active learning methods, which validate the effectiveness of our uncertainty quantification method for ranking problem. In addition, it is interesting that the performance of Core-Set is poor comparing other deep active learning methods. The reason may be that using the geometry intuition is difficult to select representative samples in the ranking task, which aims at measuring the matching degree of question-answer pair. Note that curves in Figure 3 are fluctuate due to the randomness of model training, but it dose not affect the comparison of different methods.

Then, as shown in Table 1, we compare the percentage of annotated data needed for different methods to achieve the performance of training with all data. Table 1(a) gives the result of experiments using the LSTM-based deep learning model as the underlying model. Specifically, DASL labels only 48% and 40% of the samples on the SemEvalCQA dataset and the YahooCQA dataset respectively. As a comparison, DAL requires 59% and 57% labeled training samples, BALD requires 79% and 68% labeled training samples, Core-Set requires 87% and 79% labeled training samples. On the other hand, our proposed DASL method also clearly outperforms BALD and Core-Set as shown in Table 1(b), which adopts the CNN-based deep learning model as the underlying model. In general, DASL achieves the performance of training with all data using much fewer acquisitions than other methods in any case. This justifies that our proposed DASL can effectively reduce the needed labor of labelling massive samples in training deep learning models for answer-selection tasks.

5.3 Importance of self-paced learning

To further analyze how self-paced learning contributes to performance in DASL, we implement variants of our method by simply



(a) Experiments for the LSTM-based deep learning model

(b) Experiments for the CNN-based deep learning model

Figure 3. Extensive experiment results of different methods, in which the vertical axes represent the performance and the horizontal axes represent the percentage of annotated data of the whole set. One can observe that our method DASL works consistently better than other methods

Table 1. Percentage of acquired data for different active learning methods to achieve the performance of training with all data

(a) Results for the LSTM-based deep learning model

Dataset	Core-Set	BALD	DAL	DASL
SemEvalCQA	87%	79%	59%	**48%**
YahooCQA	79%	68%	57%	**40%**

(b) Results for the CNN-based deep learning model

Dataset	Core-Set	BALD	DAL	DASL
SemEvalCQA	77%	70%	53%	**47%**
YahooCQA	84%	73%	57%	**33%**

using self-training to exploit unlabeled data. These implementations choose fixed thresholds to determine the percentage of most reliable samples to be pseudo-labeled in unlabeled dataset D_U, including the threshold 0.1(ST0.1), 0.3(ST0.3) and 0.5(ST0.5). Note that we also re-evaluate all unlabeled samples and pseudo-labeled samples in different rounds, otherwise the performance will be worse due to many retained noisy pseudo-labels generated by the pre-maturity of the initial model. In addition, from the experimental results in Figure 3, it is obvious that the prediction ability of the LSTM-based model is better than the CNN-based model, which means that the LSTM-based model is more likely to predict a right label for a sample. Therefore, we uses the LSTM-based deep learning model as the underlying model in this experiment.

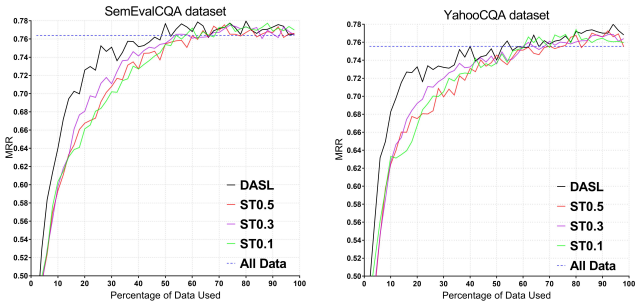


Figure 4. Performance with the increase of annotated samples between DASL and variants that simply use self-training to exploit unlabeled data

A comparison between DASL and other self-training based methods is given in Figure 4. By exploiting unlabeled data with the self-paced learning, DASL can attain higher accuracy early on, and converge to a higher accuracy overall. This demonstrates that self-paced learning has significantly improved the training efficiency. We also observe that the performance of ST0.3 is higher than both ST0.1 and ST0.5. This is mainly because the pseudo-labeled samples for training model is insufficient if the percentage is low, while the more noisy labels are incorporated if the percentage is high. In addition, to avoid costly computation due to the multiple training iterations in SPL, we set the parameter μ at a sufficiently large value. And in our experiments, the self–paced learning is repeated with around three iterations until the model achieves the maximum performance. On the other hand, due to the multiple training iterations in SPL of our method, we only implement the half of epochs than other methods for retraining the model. From the experimental results, the training time of our method only slightly increased.

6 CONCLUSION

we propose a new framework integrating active learning and self-paced learning in training deep answer selection models, which is a kind of learning-to-rank task. This framework proposes a new uncertainty quantification method based on Bayesian neural network to guide active learning and self-paced learning in the same iterative process of model training. Lastly, we conduct a thorough evaluation on two kinds of deep answer selection model with the real-world datasets including YahooCQA and SemiEvalCQA. The experiment results demonstrate the proposed method can achieve better performance than other deep active learning methods. Furthermore, this method can be easily extended to other learning-to-rank tasks such as information retrieval. In the future, we are going to make further exploration with our framework in the deep learning models of different learning-to-rank tasks.

ACKNOWLEDGEMENTS

This paper is supported by National Key R&D Program of China (No. 2018YFB1402800), National Natural Science Foundation of China (No. 61532004), State Key Laboratory of Software Development Environment (Funding No. SKLSDE-2017ZX-03)

REFERENCES

[1] Preslav Nakov Lluıs Marquez Alessandro Moschitti, Walid Magdy Hamdy Mubarak Abed Alhakim Freihat, James Glass, and Bilal Randeree, 'Semeval-2016 task 3: Community question answering', *Proceedings of SemEval*, 525–545, (2016).

[2] Massih-Reza Amini and Patrick Gallinari, 'Semi-supervised logistic regression', in *ECAI*, pp. 390–394, (2002).

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston, 'Curriculum learning', in *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48. ACM, (2009).

[4] Wenbin Cai and Ya Zhang, 'Variance maximization via noise injection for active sampling in learning to rank', in *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 1809–1813. ACM, (2012).

[5] Ying Chen, Yibo Lin, Tianyang Gai, Yajuan Su, Yayi Wei, and David Z Pan, 'Semi-supervised hotspot detection with self-paced multi-task learning', *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, (2019).

[6] David A Cohn, Zoubin Ghahramani, and Michael I Jordan, 'Active learning with statistical models', *Journal of artificial intelligence research*, **4**, 129–145, (1996).

[7] Pinar Donmez and Jaime G Carbonell, 'Optimizing estimated loss reduction for active sampling in rank learning', in *Proceedings of the 25th international conference on Machine learning*, pp. 248–255. ACM, (2008).

[8] Jerome H Friedman, 'Greedy function approximation: a gradient boosting machine', *Annals of statistics*, 1189–1232, (2001).

[9] Yarin Gal and Zoubin Ghahramani, 'Dropout as a bayesian approximation: Representing model uncertainty in deep learning', in *international conference on machine learning*, pp. 1050–1059, (2016).

[10] Yarin Gal, Riashat Islam, and Zoubin Ghahramani, 'Deep bayesian active learning with image data', in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1183–1192. JMLR. org, (2017).

[11] Fei Gao, Zhenyu Yue, Jun Wang, Jinping Sun, Erfu Yang, and Huiyu Zhou, 'A novel active semisupervised convolutional neural network algorithm for sar image recognition', *Computational intelligence and neuroscience*, **2017**, (2017).

[12] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel, 'Bayesian active learning for classification and preference learning', *arXiv preprint arXiv:1112.5745*, (2011).

[13] Peiyun Hu, Zachary C Lipton, Anima Anandkumar, and Deva Ramanan, 'Active learning with partial feedback', *arXiv preprint arXiv:1802.07427*, (2018).

[14] Gal Hyams, Daniel Greenfeld, and Dor Bank, 'Improved training for self-training by confidence assessments'.

[15] Lu Jiang, Deyu Meng, Shoou-I Yu, Zhenzhong Lan, Shiguang Shan, and Alexander Hauptmann, 'Self-paced learning with diversity', in *Advances in Neural Information Processing Systems*, pp. 2078–2086, (2014).

[16] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann, 'Self-paced curriculum learning', in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, (2015).

[17] Alex Kendall and Yarin Gal, 'What uncertainties do we need in bayesian deep learning for computer vision?', in *Advances in neural information processing systems*, pp. 5574–5584, (2017).

[18] M Pawan Kumar, Benjamin Packer, and Daphne Koller, 'Self-paced learning for latent variable models', in *Advances in Neural Information Processing Systems*, pp. 1189–1197, (2010).

[19] Tuan Manh Lai, Trung Bui, and Sheng Li, 'A review on deep learning techniques applied to answer selection', in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2132–2144, (2018).

[20] Yu-Feng Li and Zhi-Hua Zhou, 'Towards making unlabeled data never hurt', *IEEE transactions on pattern analysis and machine intelligence*, **37**(1), 175–188, (2014).

[21] Liang Lin, Keze Wang, Deyu Meng, Wangmeng Zuo, and Lei Zhang, 'Active self-paced learning for cost-effective and progressive face identification', *IEEE transactions on pattern analysis and machine intelligence*, **40**(1), 7–19, (2017).

[22] Bo Long, Jiang Bian, Olivier Chapelle, Ya Zhang, Yoshiyuki Inagaki, and Yi Chang, 'Active learning for ranking through expected loss optimization', *IEEE Transactions on Knowledge and Data Engineering*, **27**(5), 1180–1191, (2014).

[23] Rottmann Matthias, Kahl Karsten, and Gottschalk Hanno, 'Deep bayesian active semi-supervised learning', in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 158–164. IEEE, (2018).

[24] Rishabh Mehrotra and Emine Yilmaz, 'Representative & informative query selection for learning to rank using submodular functions', in *Proceedings of the 38th international ACM sigir conference on research and development in information retrieval*, pp. 545–554. ACM, (2015).

[25] Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, Jim Glass, and Bilal Randeree, 'Semeval-2015 task 3: Answer selection in community question answering', in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pp. 269–281, (2015).

[26] Kamal Nigam and Rayid Ghani, 'Analyzing the effectiveness and applicability of co-training', in *Cikm*, volume 5, p. 3, (2000).

[27] Guo-Jun Qi and Jiebo Luo, 'Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods'.

[28] Enver Sangineto, Moin Nabi, Dubravko Culibrk, and Nicu Sebe, 'Self paced deep learning for weakly supervised object detection', *IEEE transactions on pattern analysis and machine intelligence*, **41**(3), 712–725, (2018).

[29] Ozan Sener and Silvio Savarese, 'Active learning for convolutional neural networks: A core-set approach', *arXiv preprint arXiv:1708.00489*, (2017).

[30] Ozan Sener and Silvio Savarese, 'A geometric approach to active learning for convolutional neural networks', *arXiv preprint arXiv*, **1708**, 1, (2017).

[31] Burr Settles, 'Active learning literature survey', Technical report, University of Wisconsin-Madison Department of Computer Sciences, (2009).

[32] Aliaksei Severyn and Alessandro Moschitti, 'Learning to rank short text pairs with convolutional deep neural networks', in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pp. 373–382. ACM, (2015).

[33] Yanyao Shen, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar, 'Deep active learning for named entity recognition', *arXiv preprint arXiv:1707.05928*, (2017).

[34] Aditya Siddhant and Zachary C Lipton, 'Deep bayesian active learning for natural language processing: Results of a large-scale empirical study', *arXiv preprint arXiv:1808.05697*, (2018).

[35] Ivan Srba and Maria Bielikova, 'A comprehensive survey and classification of approaches for community question answering', *ACM Transactions on the Web (TWEB)*, **10**(3), 18, (2016).

[36] Yi Tay, Minh C Phan, Luu Anh Tuan, and Siu Cheung Hui, 'Learning to rank question answer pairs with holographic dual lstm architecture', in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 695–704. ACM, (2017).

[37] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui, 'Hyperbolic representation learning for fast and efficient neural question answering', in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 583–591. ACM, (2018).

[38] Xianzhi Wang, Chaoran Huang, Lina Yao, Boualem Benatallah, and Manqing Dong, 'A survey on expert recommendation in community question answering', *Journal of Computer Science and Technology*, **33**(4), 625–653, (2018).

[39] Dingwen Zhang, Deyu Meng, Long Zhao, and Junwei Han, 'Bridging saliency detection to weakly supervised object detection based on self-paced curriculum learning', *arXiv preprint arXiv:1703.01290*, (2017).