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

A Region Selection Model to Identify Unknown Unknowns in Image Datasets

Xiao Dong[1] and **Huaxiang Zhang**[2] and **Gianluca Demartini**[1]

Abstract. Unknown Unknowns (UUs) are a kind of test data points on which predictive model confidence is high but the prediction incorrect. It is crucial to identify these instances for better understanding the limitation of predictive models and to avoid critical errors. However, existing methods utilize a fixed model to identify UUs in test data, resulting in limited performance and high cost. To address these limitations, we propose a regional candidate selection algorithm that seamlessly combines a deep neural network and humans to identify UUs in visual data. Specifically, we identify several candidate regions in the data space where UUs have high probability of being present. This is achieved by comparing labels learned by a deep network and predictions obtained with the original classification model. Moreover, inspired by active learning, diversity and training loss are utilized to obtain suitable query sequences. We evaluate our method using a publicly available image dataset. Experimental results conducted on this dataset demonstrate the improved performance of the proposed method over different baselines under different conditions and its robustness to noisy labels.

1 Introduction

With the rise of machine learning and artificial intelligence (AI), classification models are often deployed for various practical tasks such as Image Classification [3], Sentiment Analysis [8] and Fake News Detection [11]. When a predictive model is trained on a small dataset, such data may be possibly biased by over-representing certain parts of the data population. When such model is then used in real-world applications, there may be cases on which the model is highly confident about its classification while being wrong. These high-confidence error cases are termed Unknown Unknowns (UUs). This problem is often caused by certain items being under-represented in the training dataset which can be, for example, dominated by white dogs and thus creates 'blind spots' for the model to identify as, for example, black dogs in a cat/dog classification task.

The UU as one kind of blind spot of the trained model, generally stems from a gap existing between the data distribution in the training set and in the test set relative to the instances available for certain areas of the feature space. In some domains, such as AI safety and healthcare, UUs can become critical issues. For instance, a disease recognition model that makes the wrong prediction with high confidence for a certain patient can lead to serious consequences such as the patient being misdiagnosed. Hence, it is important to identify such errors with high confidence before AI models can be deployed to the real world.

However, due to the high confidence of the model about some of its wrong predictions, it is unrealistic to discovery UUs efficiently and effectively without any additional human annotation. By means of crowdsourcing platforms such as Amazon Mechanical Turk[3] or FigureEight[4], it is possible to design a human-in-the-loop solution to identify UUs. However, in a large-scale data scenario, it would be unfeasible to label all data by means of crowdsourcing. Therefore, in this paper, we focus on how to identify UUs under a limited manual annotation budget constrain.

According to Attenberg *et al.*'s research [1], UUs are often concentrated in specific areas of a predictive model's feature space. In their method, a crowdsourcing-based system is developed to identify these UUs based on purely manual annotations. However, a strategy that relies entirely on humans is costly both from a time and financial incentive point of view. To solve this issue, Lakkaraju *et al.* [6] proposed a partition-based utility model to discover UUs for any black-box classifier combining clustering and reinforcement-learning. However, their method assumes that UUs are clustered in certain areas of the feature space and uses a fixed reward strategy for finding each UU sample.

Although this approach to find UUs is more efficient than manual labelling of instances, it cannot distinguish between identified UUs and it always provides the same reward for each found UU. Then, Bansal and Weld [2] develop a novel coverage-based utility model to maximize the coverage of identified UUs. In their model, they select the sample with the highest expected utility gain to be labelled at each step and achieve better coverage results than [6]. However, these kinds of methods identify UUs from all test data. This requires considerable computation time and storage resources. Especially in large-scale data scenarios, these methods often display limited performances. In addition, their predictive model is fixed and its performance cannot be improved as the number of discovered UUs increases. In our work, we exploit the position of UUs in the feature space to identify more UUs with limited resources.

To address these limitations, we propose a two-stage region selection algorithm for accurately and efficiently identifying UUs in visual datasets. In the first stage, benefiting from the feature extraction capability of Convolutional Neural Network (CNN), we employ a pre-trained deep model to extract deep features of each image in the dataset. Then, K-means is utilized to cluster these features and to obtain cluster labels. These obtained cluster labels are compared with the original predictive labels. The different parts of the feature space are selected as the initialized candidate region. In this region,

[1] School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane 4072, Australia. email: dx.icandoit@gmail.com, demartini@acm.org
[2] School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China. email: huaxzhang@163.com

[3] https://www.mturk.com/
[4] https://www.figure-eight.com/

we utilize clustering entropy values to determine which data to sample first. After sorting these samples, we make use of an oracle (i.e., human expert labellers) to collect labels and thus to identify UUs. Subsequently, all instances in this area of the feature space are used to fine-tune the deep network using the same structure of the pre-trained model. In the second stage, we employ the deep model trained in the previous stage to predict the labels of the remaining data points and compare them with the corresponding original predicted labels to obtain dissimilar parts as being a new candidate region of the feature space to contain UUs. Differently from the first stage, in this stage, we employ uncertainty, diversity, and training loss to determine the order in which to query the oracle for labels. After labels are collected for the samples in the candidate area, these samples are fed into the deep network to make it more discriminative and effective. We then repeat this process until the maximum number of oracle queries (pre-defined by a given manual annotation budget) has been reached. During the training process, the selected candidate regions are far smaller than the whole test dataset. This can save manual labelling and computation time. As the process progresses, the deep network becomes more and more discriminative after each retraining phase and enhances its feature representation. Figure 1 illustrates the basic framework of the proposed methodology. The main contributions of this paper are summarized as follows:

- A simple but effective region selection framework to seamlessly integrate deep learning and active learning for efficient Unknown Unknowns identification. With our framework, it is possible to obtain more accurate Unknown Unknown detection with lower annotation and computation time cost.
- A novel candidate region learning strategy on a deep network to progressively identify UUs under the supervision of human annotators. In the training process, the discriminative power of the deep network is gradually enhanced with the increase of the annotated sample size.
- Experimental results on a benchmark dataset demonstrating state-of-the-art performance of the proposed method on visual data.

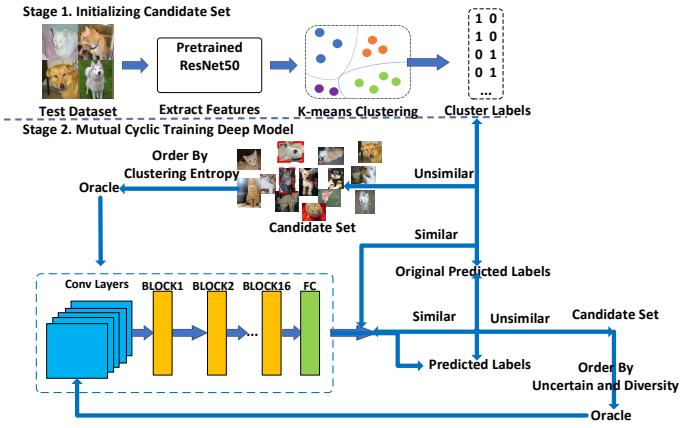


Figure 1. The high-level CReSA framework proposed in this paper.

2 Related Work

Unknown Unknowns are data items that the predictive model is confident on, but fails in classifying correctly. In [1], Attenberg *et al.* design a crowdsourcing-based system called Beat The Machine (BTM) to leverage crowd workers to identity UUs via a gamification setup. Their experimental results demonstrate that such approach is able to identify more UUs than some variants of random sampling. However, due to the complete dependence on human annotators, BTM is not very scalable and, thus, unpractical to be deployed in real-world scenarios for automated discovery of UUs.

To overcome this limitation, in [6] authors propose a two-step approach, in which a non-noisy human Oracle is requested to annotate instances with the goal of identifying UUs. Their methods are based on the assumption that UUs are located not randomly but rather concentrate in a certain area of the feature space. In the first step, test points are clustered into different groups where instances with similar feature values and similar predicted scores are put together. Then, they utilize a multi-armed bandit algorithm called Bandit for Unknown Unknowns (UUB) to find UUs in these groups. However, the reward function for finding a UU in this method is fixed. This leads to the impossibility to identify independently occurring UUs. To address this limitation, authors of [2] designed a coverage-based utility model to maximize data coverage while identifying UUs. Their utility model is based on a custom coverage function. They proved that a greedy strategy can find a cover within a constant factor from the optimal, if the positions of all UUs are known in advance. In this approach, they always select the sample with the highest expected utility gain to be queried first with the human annotation Oracle. This approach achieves better coverage results than [6].

However, all the methods mentioned above rely on a perfect Oracle (e.g., human experts) which may not always be available in real world settings. Besides this, the search space over the entire dataset can be large and thus time-consuming and annotation-heavy to explore. Furthermore, if multiple human annotators coming with various backgrounds and biases, they may provide conflicting labels. In such cases, these UU identification methods could be less effective because of the existing noise and diversity in the collected labels. In our work, we propose a deep learning based candidate region selection approach to find several suitable areas of the feature space where UUs have higher probability of being identified. This approach allows the proposed method to achieve better performance as compared to state of the art baseline UU identification approaches.

Active Learning aims to improve the generalization ability of a supervised model by requiring to only label a limited number of samples in sequence rather than as a batch. Approaches in this area include Uncertainty sampling [7], Query by committee [10], Excepted model change [9], Expected error reduction [14], and Expected variance reduction [13]. These kinds of learning frameworks mainly focus on the Known Unknown identification, that is, finding low confidence cases and collect more labels for those instances in order to increase classification confidence. This is usually done by employing various kinds of information from the predictive model to select which data needs to be labeled by human annotators. Hence, these methods are not applicable to identify Unknown Unknowns directly.

Deep Learning is a popular research topic in machine learning and data mining. It consists of utilizing deep neural networks to learn complex data representation via nonlinear functions. Many pre-trained models such as [12, 4, 5] have been proposed to solve complex vision tasks in an unsupervised scenario. Therefore, such an approach is suitable to potentially capture semantics information in im-

age data. In the absence of category information, it is a method to obtain some prior class distribution information over a test dataset. In our work we use such methods to develop a rich data representation space that allows us to discover UUs.

3 Problem Statement

For any prediction task, a model M is trained on a finite number of data items D_{train}, and applied to predict labels for items in a test dataset $D_{test} \in R^{n \times d}$ with feature $F_{test} = \{f_1, f_2, ..., f_n\}$. For each items the model returns a class label $L_{test} \in R^{n \times c}$ and a predictive score $S \in [0, 1]$, where n is the number of test data points and c is the number of categories respectively. Our aim is to find UUs of the model M in the test dataset D_{test} using a limited number of queries Q to an Oracle. According to the definition of UUs by [1], the instance x_i is a UU only if the predicted label L_{test_i} is incorrect while the corresponding confidence value s_i is larger than a pre-defined threshold γ.

As done in previous work, we assume the prediction task to be a binary classification problem and set the positive class as the critical class where false positive errors are more costly and need to be identified. Thereby, we need to find UUs that are predicted as a positive class items and for which the model confidence scores are larger than the threshold γ in the search space D_{test}. Additionally, to model noise in the labels collected from humans, we also assume that the probability of an Oracle making mistakes (i.e., of returning the incorrect label) is fixed to α for each instance[5].

4 Methodology

In this part, we present the Candidate Region Selection Algorithm (CReSA) for the UU identification problem and give the corresponding empirical and theoretical analysis. To better illustrate the effectiveness of our method, we assume that the prediction model is a black-box and that we have no access to the training set.

4.1 Network Architecture

Our method is initialized with setting network parameters based on the pre-trained ResNet50 model, which has been widely adopted as the basic model in many computer vision approaches. The ResNet50 model contains one convolutional layer (Conv 1), 16 building blocks (block1-5_x) and 1 fully-connected layer (FC 50), where each block has three layers. In this paper, to learn corresponding categories, we modify the network structure by replacing the last fully-connected layer with a fully-connected layer with c hidden units to generate the predicted classes.

4.2 Basic Learning Framework: Cyclic Learning

To quickly find as many UUs as possible, we propose a two-step human-in-the-loop learning approach. In the first step, we utilize a pre-trained network to extract deep features from the test dataset $D_{test} = \{d_i\}_{i=1}^{n}$. K-means clustering is then used to cluster features F_{test} together. This leads to cluster labels $L_{cluster}$ and cluster centers C_{center}, where n is the number of data points and c is the number of classes. The number of clusters is equal to the number of classes. Due to the complex semantic of these features, we always obtain a high clustering performance. After that, we compare the cluster labels $L_{cluster}$ with the predicted class labels L_{test} for the original prediction model M, and thus identify dissimilarities between the two annotated sets. This will result in the candidate set q. In the candidate set, we then utilize clustering entropy scores to decide the order in which to query the Oracle for labels.

In the second step, we train a deep network using the candidate set containing the labels provided by the Oracle and thus obtain prediction probabilities and classification outcome for the rest of dataset $L_{predict}$. In the same way as in the first step, we again compare the labels obtained from the network with the labels predicted by the original model and obtain the set of data points on which the two annotations are different as the candidate discovery set. After that, we utilize uncertainty, diversity, and training loss as indicators to determine the order in which to query labels to the Oracle for the rest data. Meanwhile, based on the training loss, we feed the batch data from simple to hard to classify items. Finally, we repeat these steps until the number of elements in the candidate set becomes zero.

To sort these samples, we utilize clustering entropy in the first step, and uncertainty, diversity and training loss in the second step to generate a ranking of instances ordered from hard to easy. The assumption we make is that hard instances (which are ranked first) are cases which can be easily misclassified and thus should be prioritized in the label collection sequence.

Clustering Entropy

In the area of uncertainty estimation entropy is one of the most important metrics. To determine a suitable Oracle query order for UU discovery, we need to evaluate the randomness in the clustering result. To effectively evaluate the uncertainty of clustering performance, we employ clustering entropy. If the current model is uncertain about its clustering decision for an instance, then acquiring a label for such instance may be helpful in improving the model as it contains various information that the model does not currently understand.

Considering the UU of the original model, we utilize the clustering model information to learn which instances should be queried first with the Oracle. In the clustering task, for any data point d_j, the prediction probability is defined as:

$$\{p_j\}_{i=1}^{c} = \frac{\|f_j - C_{cluster_i}\|_2^2}{\sum_{i=1}^{c} \|f_j - C_{cluster_i}\|_2^2} \tag{1}$$

To obtain the clustering entropy, we put the prediction probability into entropy H equation:

$$H(f_j) = -\sum_{i=1}^{n} p(f_j) \log(p(f_j)) \tag{2}$$

Then, we sort instances according to the entropy values from large to small, and ask the Oracle for correct labels.

Classification Uncertainty and Data Diversity

Similarity to the above approach, to sort the test data, we employ classification uncertainty values to evaluate the stability of each sample. In this paper, we use logistic regression as the prediction model. The corresponding equation is shown as follow:

$$\begin{aligned} r(pred) &= \|p(c = 1|pred) - \frac{1}{c}\| \\ p(c = 1|pred) &= \frac{1}{1 + \exp(-f(pred))} \end{aligned} \tag{3}$$

[5] We could also make such probability value dependent on the specific data item to simulate, for example, the difficulty of labelling a certain item, but we leave such an analysis for future work.

where $pred$ is the prediction of the deep network on dataset X.

In a multi-label scenario, the uncertainty can be rewritten as:

$$r(x) = 1 - \max_{class \in Class} p(class|pred) \tag{4}$$

where $p(c|pred)$ is the probability that x belongs to the class. In this way, we can identify the most uncertain instances by sorting the uncertainty values for each instance in descending order.

Furthermore, to evaluate the classification model effectiveness, we compare the label of the current instance with its neighbors. If the model assigns the correct labels to their neighbors, we could assume the label of the current instance could also be correct.

For any instance x_j, its diversity score can be represented by:

$$d_i(x_j) = \frac{1}{n_k} \sum_{x_k \in N(x_j, n_k)} S(X_j, X_k) I[c_k == c_{i_k}] \tag{5}$$

where $N(x_j, n_c)$ is the n_c nearest neighbors of x_j in the deep features extracted from the t-th retraining of the model, $S(x_j, x_k)$ measures the Euclidean similarity between x_j and x_k, and I[.] is the indicator function which returns 1 if the argument is true and 0 otherwise.

Self-paced Training

To effectively train the network, we employ a self-paced strategy to feed samples from simple to hard to the deep model. This is based on the assumption that if the loss is lower, then the sample is more likely to be classified correctly. In the network training process, we record the loss l_{x_i} value for each instance. In the next retraining phase, we sort the rest of the data in ascending order based on the loss value. The weight w_x for each sample is as follows:

$$w_{x_i} = \frac{1}{\|l_{x_i}\|} \tag{6}$$

Order Rule

The final order is represented as $or(x_j) = w_{x_i} r(x_j) d_i(x_j)$

The main steps of the proposed UU detection approach are summarized in Algorithm 1.

Algorithm 1 Candidate Regional Selection

Require:

Test dataset D_{test}, prediction class label L_{test}, predictive score S and the number of classes c.

Ensure:

Candidate Set Q.

1: Initialize network parameters for ResNet50 network.
2: Extract deep feature F by ResNet, and cluster labels $L_{cluster}$ and cluster center C_{center} with K-means on the original images.
3: Obtain $1st$ candidate set $q = \{L_{cluster} \neq L_{test}\}$ and the rest set $rq = \{L_{cluster} = L_{test}\}$, and sort $H(F)$ in decrease turn.
4: Query q with the Oracle.
5: $Q \leftarrow Q + q$.
6: **repeat**
7: Update ResNet with Q.
8: Obtain new predictive labels PL for the rest set.
9: Update $q = \{L_{predict} \neq L_{rest}\}$ and $rq = \{L_{predict} = L_{rest}\}$.
10: Obtain $r(X)$, $d(X)$, and sort $or(X)$ in decrease turn.
11: Query q with Oracle.
12: $Q \leftarrow Q + q$.
13: **until** Reach the maximum query time for oracle.

5 Experimental Evaluation

Kaggle13: This dataset contains 25k images of cats and dogs. The task is to classify whether an input image is a cat or a dog. In our paper, we set the dog as our critical class. Following previous work experimental setting [6], all black cat images are removed from the training set to create a blind spot in the trained model and generate UUs to be identified. Finally, the size of the test sets is limited by random sampling 5000 images. We construct our search space on these fixed-size test sets.

5.1 Experimental Setting

For the original model to obtain better performance on the validation set, we utilize the deep model to extract image features, as commonly done for image classification tasks. For an image, we utilize the second last layer of a pre-trained ResNet50 model, which is trained on the ImageNet dataset. Meanwhile, we apply a simple five layers CNN with two convolution layers and three fully connected layers as our classifier for the test data.

In our proposed method, there is one additional parameter, that is, the number of neighbors to be considered n_c, which needs to be set. We set n_c = 10, batch size s=20, learning rate lr = $1e$-3 and $EPOCH$=5 for all our experiments. In addition, in order to assess the effectiveness of the proposed UU identification approach, we compare our method with several baselines, that is, Random Sampling, Least average similarity, Max Average Similarity and Most uncertain. 1) Random Sampling: Randomly selects instances from the test data to be queried with the Oracle. 2) Least average similarity: Calculates the average Euclidean distance for each test instance to all training instances, and chooses the instances with the highest distance first. 3) Least maximum similarity: Computes the minimum Euclidean distance of test data instances to all training data instances and chooses instances with the highest distance first. 4) Most uncertain: Ranks the instances in the test dataset by increasing order of the prediction confidence as assigned by the original model. 5) UUB [6]:

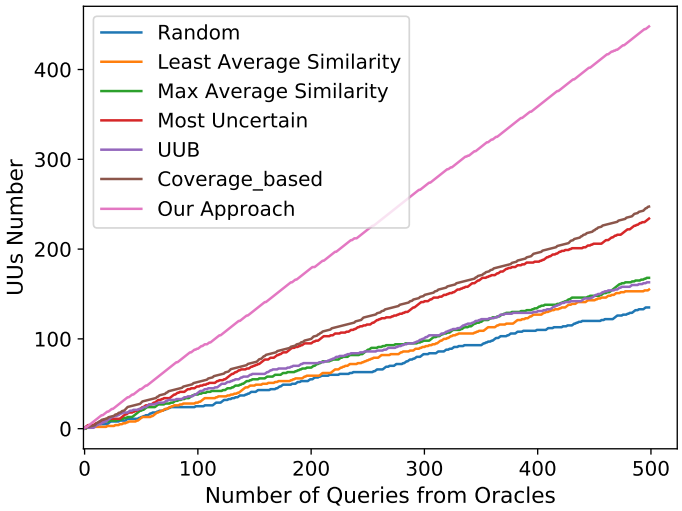
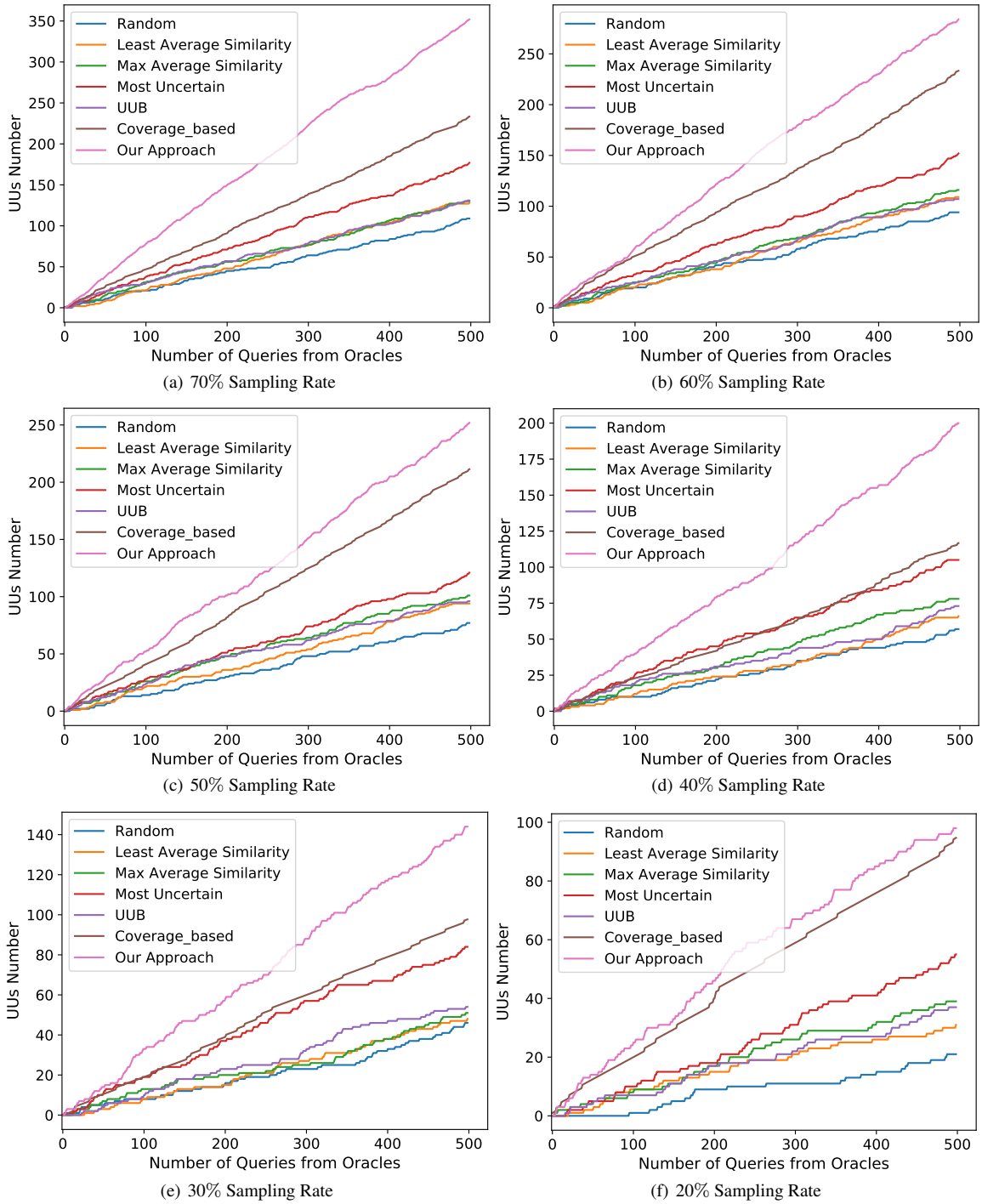


Figure 2. Evaluation of UU discovery performance on Kaggle13.



(a) 70% Sampling Rate (b) 60% Sampling Rate

(c) 50% Sampling Rate (d) 40% Sampling Rate

(e) 30% Sampling Rate (f) 20% Sampling Rate

Figure 3. UU discovery performance with the sampling rate ranging from 20% to 70%.

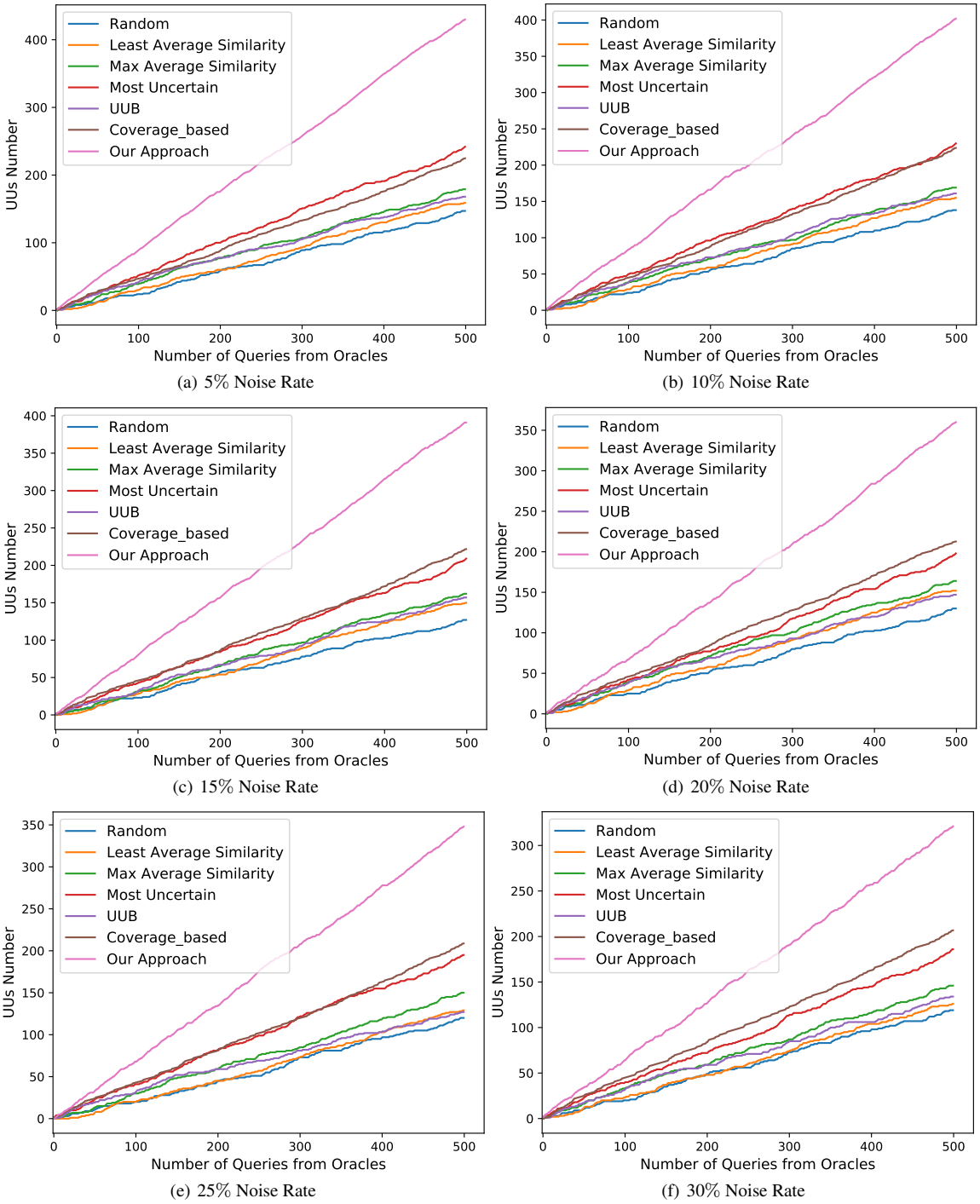
Combines clustering and reinforcement learning to identify UUs using a non-noisy Oracle. 6) Coverage_based [2]: Discovery UUs based on maximizing the coverage utility gains.

Next, we present experimental results looking at the effectiveness of the proposed and baseline approaches under different conditions: a perfect Oracle, a noisy Oracle, and multiple noisy Oracles.

5.2 Experimental Results

Perfect Oracle

In this experiment, our setting assumes all Oracles never make any mistake (we simulate having a reliable expert for the annotation task). Figure 2 shows the results. We can observe that our approach achieves the best UU identification accuracy as compared to other ap-



(a) 5% Noise Rate (b) 10% Noise Rate

(c) 15% Noise Rate (d) 20% Noise Rate

(e) 25% Noise Rate (f) 30% Noise Rate

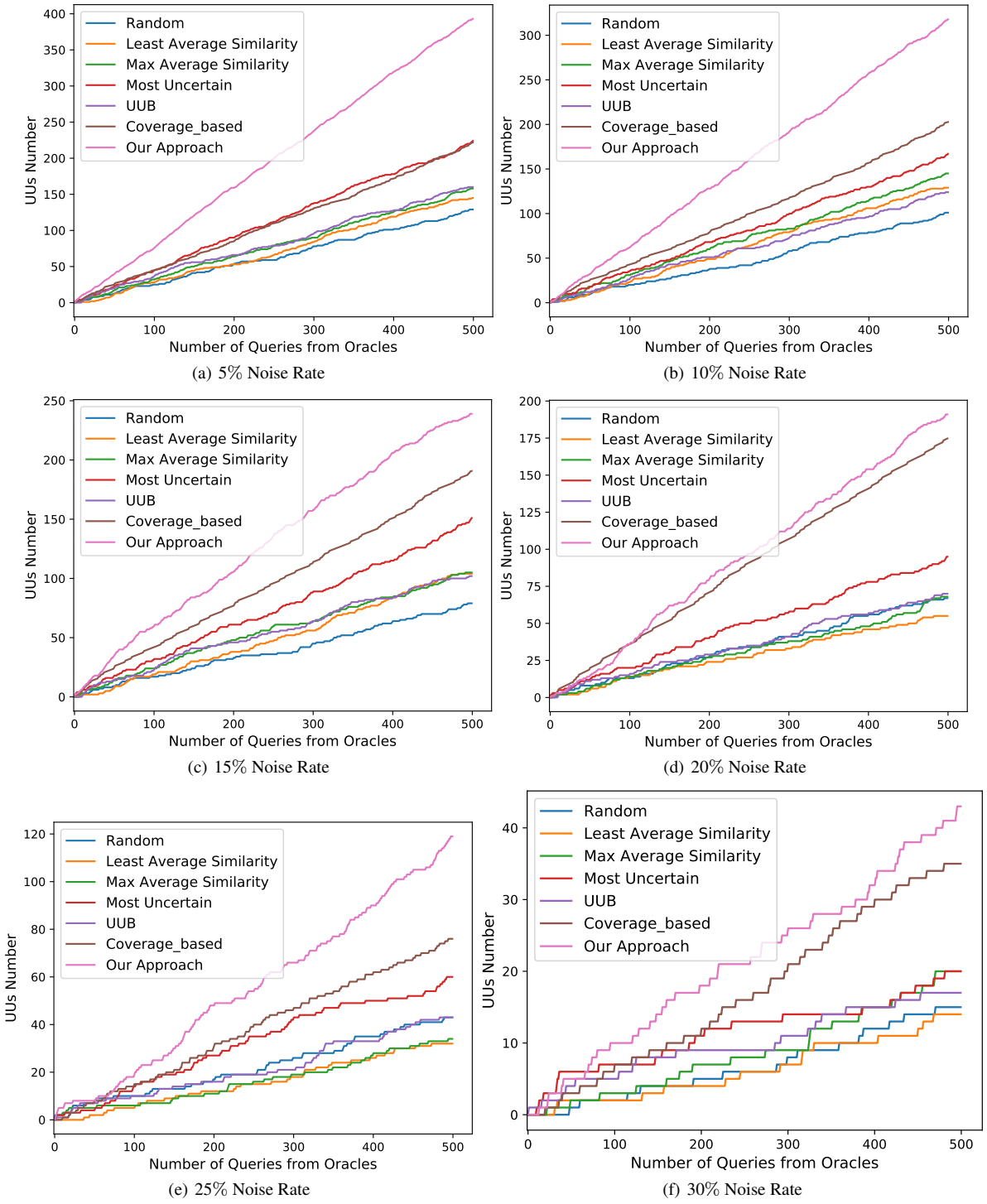
Figure 4. Number of UUs identified by different methods at different Oracle error rates ranging from 5% to 30%.

Method	Time Cost (second)
UUB [6]	36.2341
Coverage_based [2]	7.3215
Our method	3.4598

Table 1. Efficiency comparison (UU identification time in seconds).

proaches. In addition, it is evident that Least Average Similarity and Max Average Similarity outperform Random Sampling. This is because both approaches make use of extra information from the training dataset. We also find that the performance of Coverage_based [2] is similar to that of Most Uncertain and better than other methods.

Besides, we also investigate the method performance as queries increase under different UU sampling rates. The results are presented



(a) 5% Noise Rate

(b) 10% Noise Rate

(c) 15% Noise Rate

(d) 20% Noise Rate

(e) 25% Noise Rate

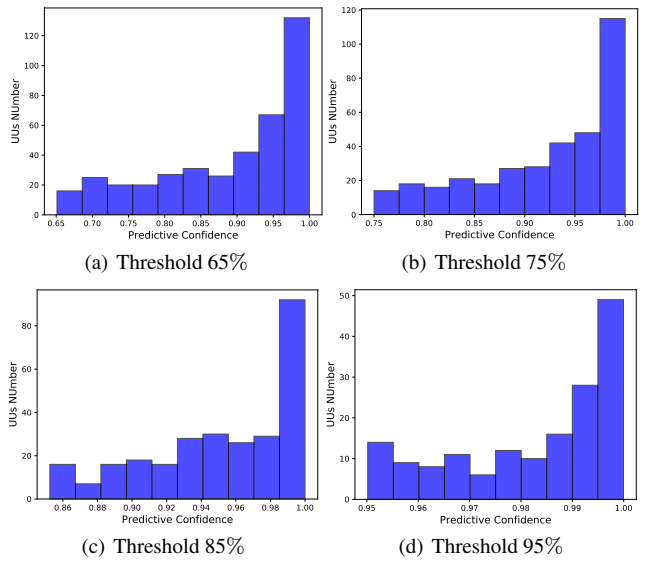
(f) 30% Noise Rate

Figure 5. Number of UUs identified by different methods at different Oracle error rates ranging from 5% to 30% using multiple Oracles.

in Figure 3. As shown, as the density of UUs decreases the number of UU found by all methods gradually decreases. However, due to the discriminative capability of the adopted deep learning model, the performance of our method always outperforms other baselines.

Noisy Oracle

To test the robustness of our approach to noisy labels (thus, simulating a crowdsourcing approach for label collection), we use the number of identified UUs as the evaluation metric. From the result shown in Figure 4, we can observe that the number of UUs identified by our method is higher than that of other approaches. This shows



(a) Threshold 65% (b) Threshold 75%

(c) Threshold 85% (d) Threshold 95%

Figure 6. Predictive confidences under different thresholds.

Sampling Rate	Identification Accuracy (mean ± std)
0.7	0.7160 ± 0.0058
0.6	0.5340 ± 0.0096
0.5	0.5100 ± 0.0216
0.4	0.4140 ± 0.0096
0.3	0.2780 ± 0.0126
0.2	0.2040 ± 0.0138

Table 2. Identification accuracy with different nearest neighbors values.

that our method is more robust to a noisy Oracle. When the error rate reaches 30%, the number of UUs identified by our method is about 320, while the UUs identified by Coverage_based [2] is less than 200.

Multiple Oracles

In a crowdsourcing setting, annotations for the same instance can be collected from several crowd workers. In this setting, we assume that each human annotator provides the same level of annotation quality and thus fix the label noise ratio for each worker. To test the robustness of our approach to multiple noisy Oracles, we again utilize the number of identified UUs as the evaluation metric and present the results in Figure 5. From the results, we can clearly see that the overall accuracy of identifying UUs is lower than the single Oracle setting. This is because in this circumstance, the issued queries are only a third of the former setting, which leads to less instances being labelled. In addition, when the number of queries is less than 100, the performance of our approach becomes worse due to the lack of correct examples in the initial training stage.

Effectiveness and Efficiency Verification

Figure 6 shows the histogram of prediction confidence for identified UUs under different threshold values varying from 65% to 95%. We observed accuracy values of 81.20%, 85.47%, 68.47% and 40.15% respectively. From Figure 6, we can see that our proposed method achieves better performance under higher threshold values. Besides, to show the efficiency of our method, we report the average UU identification time computed over 500 executions in Table 1. These results also demonstrate the high efficiency of the proposed method.

Parameter Sensitivity

We now look at the impact of the nearest neighbors value ranging 10 to 100 in our CReSA framework on the UU identification performance. The results on the Kaggle13 dataset are recorded in Table 2. As shown, our framework always achieves stable performances with low variance under different sampling rates.

6 Conclusions

We have presented a two-stage candidate region selection approach to address the Unknown Unknowns Identification problem under various conditions (i.e., perfect Oracle, noisy Oracle, and multiple noisy Oracles). Our experimental results show that the proposed method outperforms other baseline methods in finding UUs.

Acknowledgments

The work is partially supported by the ARC Discovery Project (Grant No. DP190102141), National Natural Science Foundation of China (Nos. 61772322, U1836216), the major fundamental research project of Shandong, China (No. ZR2019ZD03), and the Taishan Scholar Project of Shandong, China(No. ts20190924).

REFERENCES

[1] Joshua Attenberg, Panos Ipeirotis, and Foster J. Provost, 'Beat the machine: Challenging humans to find a predictive model's ünknown unknowns ¨', *J. Data and Information Quality*, **6**(1), 1–17, (2015).
[2] Gagan Bansal and Daniel S. Weld, 'A coverage-based utility model for identifying unknown unknowns', in *Proceedings of the Thirty-Second AAAI2018*, pp. 1463–1470, (2018).
[3] Robert M. Haralick, K. Sam Shanmugam, and Its'hak Dinstein, 'Textural features for image classification', *IEEE Trans. Systems, Man, and Cybernetics*, **3**(6), 610–621, (1973).
[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, 'Deep residual learning for image recognition', in *CVPR2016*, pp. 770–778, (2016).
[5] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, 'Densely connected convolutional networks', in *CVPR2017*, pp. 4700–4708, (2017).
[6] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Eric Horvitz, 'Identifying unknown unknowns in the open world: Representations and policies for guided exploration', in *AAAI2017*, pp. 2124–2132, (2017).
[7] David D. Lewis and William A. Gale, 'A sequential algorithm for training text classifiers', in *SIGIR1994*, pp. 3–12, (1994).
[8] Bo Pang and Lillian Lee, 'Opinion mining and sentiment analysis', *Foundations and Trends in Information Retrieval*, **2**(1-2), 1–135, (2007).
[9] Burr Settles, Mark Craven, and Soumya Ray, 'Multiple-instance active learning', in *NIPS2007*, pp. 1289–1296, (2007).
[10] H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky, 'Query by committee', in *COLT1992*, pp. 287–294, (1992).
[11] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu, 'Fake news detection on social media: A data mining perspective', *SIGKDD Explorations*, **19**(1), 22–36, (2017).
[12] Karen Simonyan and Andrew Zisserman, 'Very deep convolutional networks for large-scale image recognition', in *ICLR2015*, (2015).
[13] Tong Zhang and F Oles, 'The value of unlabeled data for classification problems', in *ICML2000*, volume 20, (2000).
[14] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani, 'Combining active learning and semi-supervised learning using gaussian fields and harmonic functions', in *ICML2003 workshop*, volume 3, (2003).