Active Deep Learning on Entity Resolution by Risk Sampling

Youcef Nafa[a,b], Qun Chen[a,b,*], Zhaoqiang Chen[a,b], Xingyu Lu[a,b], Haiyang He[a,b], Tianyi Duan[a,b] and Zhanhuai Li[a,b]

[a]School of Computer Science, Northwestern Polytechnical University, Xi'an, Shaanxi, China
[b]Key Laboratory of Big Data Storage and Management, Northwestern Polytechnical University, Xi'an, Shaanxi, China

ARTICLE INFO

Keywords:
Active Learning
Deep Learning
Risk Analysis
Entity Resolution

ABSTRACT

While the state-of-the-art performance on entity resolution (ER) has been achieved by deep learning, its effectiveness depends on large quantities of accurately labeled training data. To alleviate the data labeling burden, Active Learning (AL) presents itself as a feasible solution that focuses on data deemed useful for model training.

Building upon the recent advances in risk analysis for ER, which can provide a more refined estimate on label misprediction risk than the simpler classifier outputs, we propose a novel AL approach of risk sampling for ER. Risk sampling leverages misprediction risk estimation for active instance selection. Based on the core-set characterization for AL, we theoretically derive an optimization model which aims to minimize core-set loss with non-uniform Lipschitz continuity. Since the defined weighted K-medoids problem is NP-hard, we then present an efficient heuristic algorithm. Finally, we empirically verify the efficacy of the proposed approach on real data by a comparative study. Our extensive experiments have shown that it outperforms the existing alternatives by considerable margins.

1. Introduction

The purpose of entity resolution (ER) is to identify the equivalent records that refer to the same real-world entity. Considering the running example shown in Fig. 1, ER needs to match the paper records between two tables, R_1 and R_2. A pair $\langle r_{1i}, r_{2j}\rangle$, in which r_{1i} and r_{2j} denote a record in R_1 and R_2 respectively, is called an *equivalent* pair if and only if r_{1i} and r_{2j} refer to the same paper; otherwise, it is called an *inequivalent* pair. In this example, r_{11} and r_{21} are *equivalent* while r_{11} and r_{22} are *inequivalent*. ER can be treated as a binary classification problem tasked with labeling record pairs as *equivalent* or *inequivalent*. Therefore, various learning models have been proposed for ER [9]. As many other classification tasks (e.g. image and speech recognition), the state-of-the-art performance on ER has been achieved by deep learning [35, 14, 36, 16, 54, 31].

Unfortunately, the efficacy of Deep Neural Network (DNN) models depends on large quantities of accurately labeled training data, which may not be readily available in practical scenarios. One possible way to overcome this issue is by active learning, in which data are actively sampled to be labeled by human oracles with the goal of maximizing model performance while minimizing labeling costs. Various sampling strategies have been proposed for active learning over the years coming from different perspectives, e.g. uncertainty [29], representativeness [40] and expected model change [23]. There have also been different combinations of Uncertainty with Representativeness [50, 15] or with Expected Model Change [53] in an attempt to get the best of both worlds. In the traditional setting, AL algorithms typically choose a single point at each iteration; however, this is not feasible for DNN models since 1) a single point is likely to have no statistically significant impact on the accuracy due to the locality of optimization methods, and 2) each iteration requires a full training until convergence which makes it intractable to query labels one-by-one. Hence, most proposed AL algorithms for DNNs [50, 40, 4, 46, 26, 1], take the strategy of batch selection that queries labels for a large subset at each iteration.

Uncertainty, considered the cheapest to obtain, is the mostly used sampling strategy due to its robustness across architectures and domains [51]. Empirical studies [18] have also revealed that it is usually highly competitive with the existing but more complicated alternatives. We note that risk analysis for ER has been recently studied [7, 20, 8] with the latter representing the most recent interpretable and learnable solution, henceforth denoted LearnRisk. Risk analysis estimates the misprediction risk of a classifier when applied to a certain workload. It has been empirically shown [8] that LearnRisk can identify mislabeled instances with considerably higher accuracy than the existing uncertainty measures, which are directly estimated upon classifier outputs. Traditionally, the motivation behind using uncertainty sampling in AL is to make the model more familiar with examples that come from uncertain areas. Risk analysis goes a step further by detecting mispredictions on unseen data regardless of the classifier's self-reported uncertainty. This enables access to more informative examples that could have a positive impact on classifier training. Hence, risk analysis is naturally fit as an AL sampling strategy.

Therefore, in this paper, we propose a novel AL approach of risk sampling for ER. Fig. 2 illustrates the risk sampling framework, which leverages the results of risk analysis in the

*Corresponding author
Email addresses: youcef.nafa@mail.nwpu.edu.cn (Y. Nafa); chenbenben@nwpu.edu.cn (Q. Chen); chenzhaoqiang@mail.nwpu.edu.cn (Z. Chen); matthewlxy@mail.nwpu.edu.cn (X. Lu); haiyanghe@mail.nwpu.edu.cn (H. He); tianyiduan@mail.nwpu.edu.cn (T. Duan); lizhh@nwpu.edu.cn (Z. Li)
URL: chenbenben.org (Q. Chen)
ORCID(s): 0000-0002-9367-3583 (Y. Nafa)

R_1

ID	Title	Author	Venue	Year
r_{11}	Search Engine Guided Neural Machine Translation	Gu, J., Wang, Y., Cho, K. and Li, V.O.	AAAI	2018
r_{12}	Human-Centric Justification of Machine Learning Predictions	Biran, O., & McKeown, K. R.	IJCAI	2017
r_{13}	Learning deep unsupervised binary codes for image retrieval	Chen, J., Cheung, W. K., & Wang, A.	IJCAI	2018

...

R_2

ID	Title	Author	Venue	Year
r_{21}	Search Engine Guided Neural Machine Translation	Jiatao Gu, Yong Wang, Kyunghyun Cho, Victor O. K. Li.	AAAI	2018
r_{22}	Robust unsupervised feature selection	Qian, Mingjie, and Chengxiang Zhai.	IJCAI	2013
r_{23}	Partial multi-label learning	Xie, M. K., & Huang, S. J.	AAAI	2018

...

Figure 1: An Entity Resolution running example.

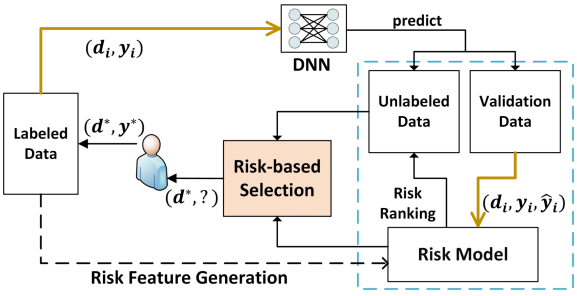


Figure 2: The Framework of Risk Sampling for Active Learning.

sampling step of active learning. Based on the existing core-set characterization [40] for AL, we theoretically derive an optimization model which aims to minimize the core-set loss with non-uniform Lipschitz continuity. Since the defined weighted K-medoids problem is NP-hard, we then present an efficient heuristic algorithm. It is noteworthy that the strategy of risk sampling tends to sample challenging examples for the classifier, and spur it into learning better representations by looking farther than the low confidence regions in the current representation space. The main contributions of this paper can be summarized as follows:

- We propose a novel approach of risk sampling for ER active learning.
- We define a theoretically sound optimization model for risk sampling, and due to its NP-hardness, present an efficient weighted K-medoids algorithm.
- We validate the efficacy of the proposed solution on real datasets. Our extensive experiments have shown that it can effectively accelerate classifier training compared with the existing alternatives. Furthermore, its performance is very robust w.r.t the size of validation data.

2. Related Work

We review related work from the orthogonal perspectives of entity resolution and active learning.

Entity Resolution. Also known as Entity Matching or Record Linkage, ER plays a key role in data integration and has been extensively studied in the literature [10, 11]. Other than the rule-based and probabilistic solutions [30, 42, 43], several machine learning models have been proposed, including Support Vector Machines [9], end-to-end deep neural network architectures [35, 14, 36, 16], and pretrained models [54, 31].

ER remains very challenging in real scenarios due to the prevalence of dirty data. Therefore, there is a need for *risk analysis*, alternatively called trust scoring or confidence ranking in the literature. It encompasses a multitude of methods that all intend to detect situations where a deployed DNN model is prone to misprediction. The proposed solutions range from those simply based on the model's output probabilities to more sophisticated, interpretable, and learnable ones [19, 25, 12, 52, 8]. Among them, LearnRisk [8] is an interpretable and learnable framework for ER that is able to construct a dynamic risk model tuned towards a specific workload. It measures the risk using the VaR (Value-at-Risk) [45] metric from financial investment modeling. It is noteworthy that the concept of risk has also been leveraged in other works for different purposes as demonstrated in [49, 48, 2]. For instance, in [49] , RecRisk considers two risk facets, *Sense Drop* and *Blue Joy*, which affect recommendation quality from the user perspective, and presents an approach to minimize the value of risk facets in personalized recommendations.

Active Learning. Active learning has been extensively researched in the context of machine learning. The most prominent approaches that proved to perform well include *margin-based*, *maximum entropy*, *Query by committee* and *Expected variance reduction* to name a few [41] . However, many of the above methods pose challenges when applied to deep neural networks. The margin-based approaches are hindered by the fact that neural networks have an intractable decision boundary. Query by committee requires maintaining multiple classifiers and retraining them in each iteration which is not very practical. Similarly, the variance reduction methods require classifier retraining for each unlabeled point; this process is prohibitive even for shallow models. Active learning for ER has also received great attention [38, 24, 32, 37, 5]. In the low-resource setting, ER was also tackled using deep transfer active learning [26]. The proposed sampling method relies on Entropy to select a batch combined of uncertain and high-confidence pairs representing both class labels. The automatically labeled high-confidence pairs are added to prevent overfitting to uncertain examples.

More recently, active learning for DNNs has also been studied, mostly focusing on image classification. It has been shown in [17, 28] that applying dropout at test time can approximate Bayesian inference enabling the application of Bayesian methods to deep learning. It has also been shown in [3] that ensemble-based uncertainty performs better and is more calibrated than a single model or dropout-based uncertainty. The framework proposed in [47] automatically selected and pseudo-annotated unlabeled samples in addition to uncertain samples. The work in [13] approximated the

distance to decision boundary by the distance to the nearest adversarial example. The works in [23] and [53] used an expected model change measure which chooses examples that maximize the impact on the learned model weights when labeled. Other recent works include generative data augmentation for AL [46], adversarial network-based discrimination of informative points [44] and detrimental point processes-based batch selection [4] to name a few. There also exist proposals combining uncertainty with representativeness using data representation and entropy such as [50, 15], or relying on gradient-based representation and gradient amplitude as a proxy to uncertainty [1]. It is worthy to point out that the proposed approach of risk sampling can be easily generalized to image classification when provided with effective risk analysis techniques.

3. Preliminaries

In this section, we formally state the AL task, and then introduce the risk analysis technique for ER, LearnRisk.

3.1. Task statement

Suppose that we have a set of record pairs $D = \{d_i, y_i\}$, where a pair d_i can be labeled as *equivalent* ($y_i = 1$) or *inequivalent* ($y_i = 0$). We follow the standard pool-based setting in which the set of training data, D, is partitioned into a small initial labeled set $L = \{d_j, y_j\}$ and an unlabeled set U. We also assume the existence of two other sets: a validation set V that is commonly used for hyperparameter tuning as well as early stopping for DNN classifiers, and an independent test set T used to evaluate the classifier's generalization performance on unseen data.

The task of ER active learning is formally defined as follows:

Definition 1. Provided with the test and validation sets T and V, the labeled set L and the unlabeled set (the pool) U, active learning iteratively selects a batch of data $Q \subseteq U$ that minimizes a specified criterion given a classifier h_L trained on L. At each iteration, once Q is labeled, it is removed from U and added to the labeled set L, i.e. $U \leftarrow U \setminus Q$, $L \leftarrow L \cup Q$; finally, a classifier is retrained on the updated set L.

3.2. Risk Analysis for ER: LearnRisk

Originally proposed in [8], the risk analysis pipeline operates in three main steps: *Risk feature generation* followed by *Risk model construction* and finally *Risk model training*.

3.2.1. Risk feature generation

This step automatically generates risk features in the form of interpretable rules based on one-sided decision trees. The algorithm ensures that the resulting rule-set is both discriminative, i.e, each rule is highly indicative of one class label over the other; and has a high data coverage, i.e, its validity spans over a subpopulation of the workload. As opposed to classical settings where a rule is used to label pairs to be equivalent or inequivalent, a risk feature focuses exclusively on one single class. Consequently, risk features act as indicators of the cases where a classifier's prediction goes against the knowledge embedded in them. An example of such rules is:

$$r_i[Year] \neq r_j[Year] \rightarrow inequivalent(r_i, r_j),$$

where r_i denotes a record and $r_i[Year]$ denotes r_i's $Year$ attribute value. With this knowledge, a pair predicted as *equivalent* whose two records have different publication years is assumed to have a high risk of being mislabeled.

3.2.2. Risk model construction

Once high-quality features have been generated, the latter are readily available for the risk model to make use of, allowing it to be able to judge a classifier's outputs backing up its decisions with human-friendly explanations. To achieve this goal, LearnRisk, drawing inspiration from investment theory, models each pair's equivalence probability distribution (portfolio reward) as the aggregation of the distributions of its compositional features (stock rewards).

Practically, the equivalence probability of a pair d_i is modeled by a random variable p_i that follows a normal distribution $\mathcal{N}(\mu_i, \sigma_i^2)$, where μ_i and σ_i^2 denote its expectation and variance respectively. Given a set of m risk features $f_1, f_2, ..., f_m$, let $\mathbf{w} = [w_1, w_2, ..., w_m]$ denote their corresponding weight vector. Suppose that $\mu_F = [\mu_{f_1}, \mu_{f_2}, \ldots, \mu_{f_m}]^T$ and $\sigma_F^2 = [\sigma_{f_1}^2, \sigma_{f_2}^2, \ldots, \sigma_{f_m}^2]^T$ represent their corresponding expectation and variance vectors respectively, such that $\mathcal{N}(\mu_{f_j}, \sigma_{f_j}^2)$ denotes the equivalence probability distribution of the feature f_j. Accordingly, d_i's distribution parameters are estimated by:

$$\mu_i = \mathbf{b}_i(\mathbf{w} \circ \mu_F)\,;\ \sigma_i^2 = \mathbf{b}_i(\mathbf{w}^2 \circ \sigma_F^2)$$

Where $\circ$ represents the element-wise product and $\mathbf{b}_i$ is a one-hot feature vector.

Besides one-sided decision rules, LearnRisk also incorporates classifier's output probability as one of the risk features. Provided with the equivalence distribution p_i for d_i, its risk is estimated by the metric of Value-at-Risk (VaR) [45]. Note that compared with previous simpler alternatives using a single value to represent equivalence probability, LearnRisk can more accurately capture the uncertainty of the label status by a distribution.

3.2.3. Risk model training

Finally, the risk model is trained on a classifier's validation data to optimize a learn-to-rank objective [6] by tuning the risk feature weight parameters (w_i) as well as their variances (σ_i^2). As for their expectations (μ_i), they are considered as prior knowledge, and are estimated from labeled training data. Once trained, the risk model can be used to assess the misclassification risk on an unseen workload labeled by the classifier.

Table 1
Notation.

Symbol	Description
d_i	a pair of left and right records $\langle \overrightarrow{r_i}, \overrightarrow{r_i} \rangle$
y_i	pair's label
n_A	number of attributes per record
T_{d_i}	total number of tokens in pair d_i
$\overrightarrow{a_k}$ ($\overleftarrow{a_k}$)	k-th attribute of the left (resp. right) record
w_t^k	t-th token for attribute a_k
T_k	number of tokens in attribute a_k
m	word embedding dimension
$X_{d_i} \in \mathbb{R}^{T_{d_i} \times m}$	pair d_i's matrix representation
$X^k \in \mathbb{R}^{T_k \times m}$	representation of a_k
$X_t^k \in \mathbb{R}^m$	t-th token's vector representation in attribute a_k

4. Risk Sampling

In AL, each individual iteration can be seen as a standard supervised learning procedure in which a model is fit to labeled data, then the best configuration is selected based on the performance on a disjoint validation set. As shown in Fig. 2, the incorporation of risk analysis as an extra step into the process is therefore fairly straightforward. In this section, we illustrate the approach of risk sampling on the classical DeepMatcher model [35] for ER, which was built upon recurrent neural networks (RNN). The proposed approach is however similarly applicable to other DNN solutions, provided they can be shown to be Lipschitz-continuous.

In the rest of this section, we first theoretically derive the optimization model for risk sampling based on the core-set characterization, and then due to its NP-hardness, present a heuristic algorithm for its efficient solution. The notation used throughout this section as well as in Appendix is given in Table 1.

4.1. Optimization Model: Theoretical Derivation

Suppose that the ER workload, D, is drawn from a distribution p_Z. Based on the core-set characterization for AL presented in [40], we consider the upper-bound of active learning loss in batch setting defined as

$$
\begin{aligned}
&|E_{d,y\sim p_Z}[l(d,y)]| \\
&\leq \left| E_{d,y\sim p_Z}[l(d,y)] - \frac{1}{n} \sum_{(d_i,y_i)\in D} l(d_i,y_i) \right| \\
&+ \frac{1}{|Q|} \sum_{(d_j,y_j)\in Q} l(d_j,y_j) \\
&\quad + \left| \frac{1}{n} \sum_{(d_i,y_i)\in D} l(d_i,y_i) - \frac{1}{|Q|} \sum_{(d_j,y_j)\in Q} l(d_j,y_j) \right|
\end{aligned}
$$

in which the loss is controlled by the training error of the model on the labeled subset, the generalization error over the full dataset and a term referred to as the core-set loss. Core-set loss is simply the difference between average empirical loss over the set of points which have labels and the average empirical loss over the entire dataset including unlabeled points. Empirically, it is widely observed that DNNs are highly expressive leading to very low training error and they typically generalize well for various classification problems. Hence, the critical part for active learning is the core-set loss. Following this observation, we start off with the core-set loss defined as

$$
\left| \frac{1}{n} \sum_{(d_i,y_i)\in D} l(d_i,y_i) - \frac{1}{|L\cup Q|} \sum_{(d_j,y_j)\in L\cup Q} l(d_j,y_j) \right| \quad (1)
$$

Where l is the loss of the model trained on $L \cup Q$ ($A_{L\cup Q}$). Informally, given an initial labeled set (L) and a budget (b), we are trying to find a set of points to query (Q), such that the learned model's performance on the labeled subset ($L \cup Q$) and that on the whole dataset (D) will be as close as possible. In [40], it has been shown that provided with a λ-Lipschitz continuous convolutional neural network, if a set of balls, denoted by s, with radius δ_s centered at each member of s can cover the entire set D, the core-set loss can be bound with the covering radius δ_s and a term that goes to zero with rate depending only on n.

The existing core-set characterization applies the global Lipschitz value for all unlabeled points. However, it can be observed that, provided a Lipschitz continuous DNN, the local Lipschitz continuities of unlabeled points are usually not uniform, or their local Lipschitz values may be vastly different. Since the DeepMatcher model was built upon RNN, in what follows, we first theoretically establish the Lipschitz continuity of RNN and the DNN model of DeepMatcher, and then derive the optimization model for risk sampling based on non-uniform Lipschitz continuity.

Lipschitz Continuity of RNN. For a generic RNN, we have Lemma 1 on its Lipschitz continuity. We have provided the proofs of the lemmas and theorems in the appendix.

Lemma 1. *The loss function defined as the 2-norm between one-hot class labels and the Softmax outputs of a stable RNN with T time steps and input dimension m, followed by n_{fc} fully connected layers defined over C classes is $\frac{\sqrt{(C-1)\,T\,m}}{C}\,\alpha^{n_{fc}+1}$-Lipschitz.*

Note that α in Lemma 1 is a bound over the operator norms of all trainable matrices in the RNN and fully connected layers. Although α is in general unbounded, it can be made arbitrarily small without changing the loss function's behavior. Moreover, an RNN is said to be *stable* when the gradients cannot explode, which is only valid when $\alpha < 1$ [34]. In order to extend the result in Lemma 1 to the DeepMatcher solution for ER, we define a corresponding neural network model, then show that it is Lipschitz continuous in Theorem 1.

Definition 2. DNN Model for ER. Suppose that each pair, denoted by $d_i = \langle \overrightarrow{r_i}, \overrightarrow{r_i} \rangle$, in an ER workload, consists of n_A attributes per record r_i, where each attribute a_k is a sequence of T_k tokens w_t^k. The model first embeds each attribute a_k as a sequence of vectors using an embedding matrix E ($X_t^k = E[w_t^k]$). Then, each attribute is encoded by a stable RNN

into a representation $\mathbf{s_k} \in \mathbb{R}^m$ as

$$\mathbf{s_k} = RNN(\mathbf{X^k}).$$

Let the attribute similarity layer be defined by a distance function $F_D : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}^m$. The k-th attribute pair similarity $\tilde{s_k}$ between $\overleftarrow{a_k}$ and $\overrightarrow{a_k}$ is then defined as

$$\tilde{\mathbf{s_k}} = F_D(\overleftarrow{\mathbf{s_k}}, \overrightarrow{\mathbf{s_k}}).$$

Finally, the classification layer F_C is defined by a fully-connected neural network followed by a Softmax function. The model takes the aggregated pair similarities as input and returns the match probability p by

$$p = F_C(\{\tilde{\mathbf{s}}_\mathbf{1}, \ldots, \tilde{\mathbf{s}}_{\mathbf{n_A}}\}).$$

The model defined in Definition 2 is consistent with the network structure defined in the RNN variant of DeepMatcher [35]. On its Lipschitz continuity, we have Theorem 1.

Theorem 1. *The loss function defined as the 2-norm between one-hot class labels and the Softmax outputs of an RNN-based ER model as defined in Definition 2 with input representation dimension m and maximal number of tokens per pair $\hat{T}$ is $\frac{\alpha^{n_{fc}+1}}{2}\sqrt{\hat{T}\,m}$-Lipschitz.*

Optimization Model. Based on the Lipschitz continuity of the DNN model for ER, we establish an upper-bound on the core-set loss of active learning in Theorem 2.

Theorem 2. *Given a dataset D of size n containing a labeled subset L and a Lipschitz continuous classifier, the core-set loss of active learning satisfies the following upper-bound:*

$$\left|\frac{1}{n}\sum_{(d_i,y_i)\in D} l(d_i,y_i) - \frac{1}{|L\cup Q|}\sum_{(d_j,y_j)\in L\cup Q} l(d_j,y_j)\right| \leq \frac{1}{n}\sum_{(d_j,y_j)\in L\cup Q}\sum_{(d_i,y_i)\in C_j} \mathbb{L}_i\,||X_{d_i} - X_{d_j}||_2 \quad (2)$$

in which $\mathbb{L}_i$ represents its Lipschitz constant for the loss of the model trained on $L\cup Q$, C_j is D's j-th cluster with $(d_j, y_j) \in L \cup Q$ representing its center and $||.||_2$ is the L_2 norm.

According to Theorem 2, we define the optimization objective for AL as:

$$\min_Q \sum_{(d_j,y_j)\in L\cup Q}\sum_{(d_i,y_i)\in C_j} \mathbb{L}_i\,||X_{d_i} - X_{d_j}||_2. \quad (3)$$

Unfortunately, in Eq. 3, $\mathbb{L}_i$ is not available prior to the selection of Q and the training of $A_{L\cup Q}$. However, it can be observed that given an unlabeled point, its Lipschitz value is closely correlated with its misprediction risk. Indeed, if we consider an unlabeled point d_i's misprediction risk $R_L(d_i)$ as its expected loss, i.e. $R_L(d_i) = E[l(d_i, y_i)]$, its Lipschitz value can be empirically estimated by

$$\mathbb{L}_i' = \frac{R_L(d_i)}{\min_{(d_j,y_j)\in L}||X_{d_i} - X_{d_j}||_2}, \quad (4)$$

Algorithm 1: Weighted fastPAM

Input: D : Full data
L : Initial labeled data
$b > 0$: Query budget
Output: Query Q.

```
Let Q ← top b points ranked by 𝕃_i
Calculate the total deviation TD for the initial solution L ∪ Q by Eq. 6
repeat
    for all x_j ∈ D \ (L ∪ Q) do
        d_j ← 𝕃_j · d_nearest(x_j)
        ΔTD ← (0, .., 0, −d_j, ..., −d_j)
        for all x_o ≠ x_j do
            d_oj ← d(x_o, x_j)
            if n ∈ Q then
                Update ΔTD_n
            if d_oj ≤ d_n then
                Update ΔTD_i for m_i ∈ Q \ {m_n}
        Save best swap (ΔTD*, m*, x*)
    if ΔTD* < 0 then
        Swap(m*,x*)
        TD ← TD + ΔTD*
until ΔTD* ≥ 0
```

in which d_i and d_j denote an unlabeled point and a labeled point, respectively. $R_L(d_i)$ denotes the misprediction risk of d_i. This follows straightforwardly from the Lipschitz constant definition for the DNN loss function ($|l(d_i, y_i) - l(d_j, y_j)| \leq \mathbb{L}\,||X_{d_i} - X_{d_j}||_2$). Since the loss of the labeled pair is assumed to be zero, the loss of the unlabeled pair is estimated via its misprediction risk $R_L(d_i)$. Therefore, we approximate $\mathbb{L}_i$ with its empirical estimation based on the latest classifier, which is conveniently available as shown in Eq. 4. The optimization objective of risk sampling is finally defined as

$$\min_Q \sum_{(d_j,y_j)\in L\cup Q}\sum_{(d_i,y_i)\in C_j} \mathbb{L}_i'\,||X_{d_i} - X_{d_j}||_2. \quad (5)$$

4.2. Algorithm

Clearly, the optimization problem defined in Eq. 5 is a sample-weighted version of the classical k-medoids clustering problem [27] with the addition of the weight $\mathbb{L}_i$ for each non-medoid $\mathbf{x_i}$. Given a specified number of clusters k, k-medoids aims at finding k clusters where each cluster is centered around a point in the data. Due to its NP-hardness [33], the classic way to solve the k-medoids problem is via the heuristic Partitioning Around Medoids (PAM) algorithm [27], or its more recent optimized version, namely, fastPAM [39]. Hence, we adapt the fastPAM algorithm to risk sampling.

In the scenario of risk sampling, the number of clusters is the size of the labeled data in addition to the data to be queried, i.e. $k = |L \cup Q|$. The *total deviation* (TD) objective to be minimized as shown in Eq. 5 is measured by the sum of dissimilarities of each point to the medoid of its

cluster weighted by its corresponding sample-weight with Euclidean distance as its dissimilarity measure, i.e.

$$TD = \sum_{(d_j,y_j)\in L\cup Q} \sum_{(d_i,y_i)\in C_j} \mathbb{L}'_i \, ||X_{d_i} - X_{d_j}||_2 \qquad (6)$$

For risk sampling, we need to only optimize Q while keeping L fixed. As fastPAM, the proposed algorithm similarly consists of two phases, BUILD and SWAP. To keep L fixed, we force the initial solution to contain L in BUILD, and then only allow the points in Q to be swapped out of the solution in the SWAP phase.

The algorithm is sketched in Algorithm 1. The first phase generates an initial solution $L \cup Q$ in line 1. After that, the main search loop for phase two is started at line 3. In each iteration, the algorithm will go through candidate points in line 4, calculating the reduction in the total deviation (ΔTD) for each candidate when swapped in place of any non-labeled medoid ($m \notin L$). Lines 7-12 perform the actual calculation w.r.t each medoid and accumulate the values in the ΔTD vector. The best swap across candidates and medoids is maintained in $(\Delta TD^*, m^*, x^*)$ on line 13. The iteration ends by performing the swap between m^* and x^* as long as it provides a decrease in TD. Otherwise, the algorithm has converged and Q is returned as the selected query.

The asymptotic complexity of each iteration in Algorithm 1 is in the order of $O(b(n-k)^2)$ in the worst case. Usually, the number of iterations is less than k as observed in [39] as well as in our experiments. As a result, the total worst-case complexity can be represented by $O(kb(n-k)^2)$. With the right cashing of the pairwise distances and the values returned by $n = nearest()$, $d_n = d_{nearest}()$, and $d_s = d_{second}()$; the execution time is monopolized by the nested loops. In our implementation, we opted for a GPU-friendly version of the algorithm by transforming the internal loops into matrix operations and processing the candidates in a batch-wise manner. The execution time can be orders of magnitude faster than the CPU implementation. On the other hand, any future algorithm for the k-medoids clustering problem can be easily adapted to risk sampling.

5. Experiments

In this section, we empirically evaluate the performance of risk sampling on real benchmark datasets. It is organized as follows: Subsection 5.1 describes the experimental setting. Subsection 5.2 presents the comparative evaluation results. Subsection 5.3 evaluates the robustness of risk sampling w.r.t the size of validation data.

5.1. Experimental Setting

Our testbed consists of four datasets from three domains: (1) **Publications.** From the literature domain, we used Citeseer-DBLP[1] and DBLP-Scholar[2] datasets; (2) **Products.** We selected a dataset containing the record pairs from Abt.com and Buy.com online shopping websites[2]; (3) **Music.** We manually created the Songs dataset from the 1-Million Songs corpus[1], blocked to generate a dataset of size 30k. The statistics of the test datasets are detailed in 2.

We compare risk sampling, denoted by **Risk**, with the following alternatives:

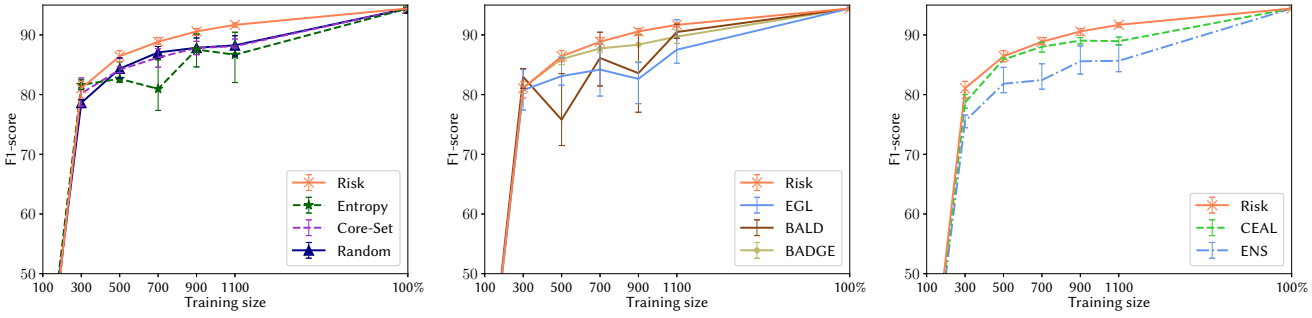
1. **Random sampling**. The commonly used baseline method which selects points uniformly from the unlabeled set;
2. **Maximum Entropy** and **BALD** [21]. Both are based on uncertainty measurement. **Maximum Entropy** samples points with the highest entropy value, while **BALD** chooses points that maximize the mutual information with the model parameters.
3. **ENS**. An ensemble-based uncertainty method that uses an ensemble of N classifiers and averages softmax vectors of each ensemble member as output. Uncertainty is measured using maximum entropy.
4. **CEAL**. Complements uncertain examples selected according to maximum entropy with a set of high-confidence examples which are softly labeled.
5. **Core-Set** [40]. It is the state-of-the-art *Representativeness*-based approach for DNNs;
6. **EGL** [53]. The state-of-the-art approach based on *Expected Model Change*, it chooses points that cause the biggest change to the embedding layer parameters;
7. **BADGE** [1]. A recently proposed approach which trades off between *diversity and uncertainty* by sampling points with diverse gradient embeddings.

These techniques can provide a good coverage of the existing effective AL approaches for deep models. We have implemented the AL solutions upon the hybrid variant of the classical DNN model for ER, DeepMatcher[2]. For the *BALD* method that requires test-time dropout, we use a dropout rate of 0.2 in the inputs to the RNN module in the embedding contextualization and word aggregation layers. The number of McDropout iterations is set to 100. The implementation of *ENS* uses 5 snapshot ensembles [22] as in the original work [3]. Their speed, compared with traditional ensembles, proves very useful in active learning's many iterations. In the implementation of *CEAL*, besides the b samples selected for query, a set of b high-confidence samples is automatically labeled using the DNN model's output probability as a soft label. Because *EGL* requires two backward passes for each example (each pass assumes a different class label), its application to the full unlabeled set can be very time-consuming. Thus, we randomly sample an unlabeled subset on which EGL-based selection is performed. For *Core-Set*, *BADGE*, and *Risk*, we use the representations of the classifier's penultimate representation layer, prior to the classification layer, for both representations and gradients.

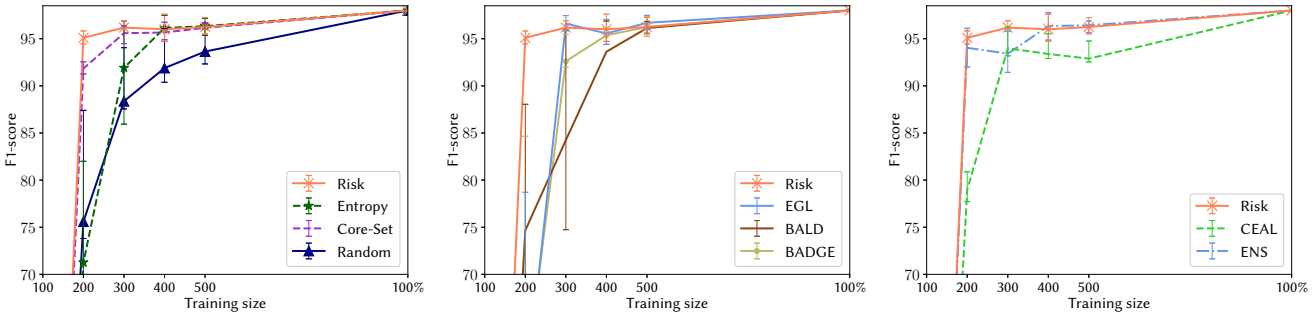
As per Definition 1, we use a labeled seed set for initial model training. We provide 100 labeled examples for publications datasets, 50 examples for Songs, and 575 examples (10% of the unlabeled pool) for Abt-Buy. Similarly, the budget b was chosen to be in a reasonable range w.r.t each specific domain. b cannot be too small that it does not

[1] https://sites.google.com/site/anhaidgroup/useful-stuff/data

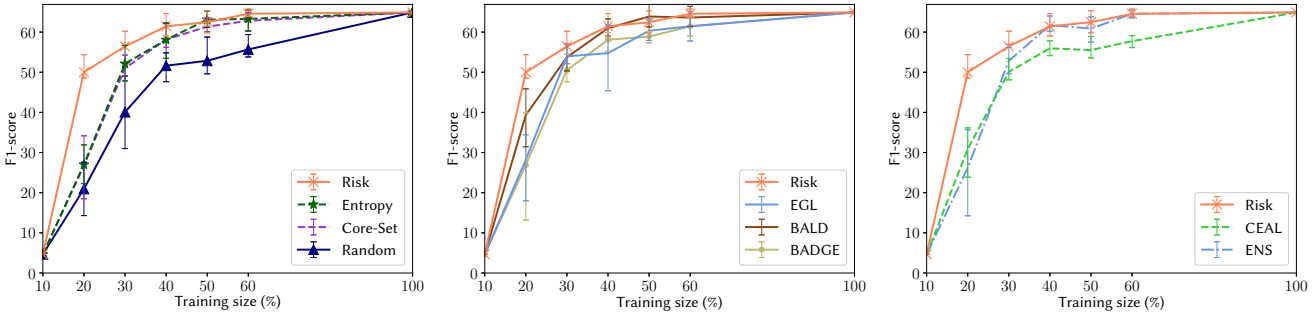
[2] https://github.com/anhaidgroup/deepmatcher/



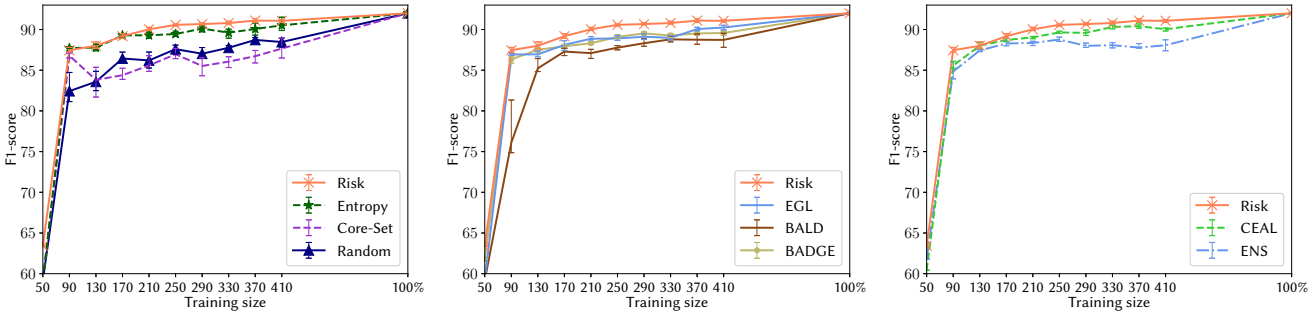
(a) DBLP-Scholar



(b) Citeseer-DBLP



(c) Abt-Buy



(d) Songs

Figure 3: Comparative Evaluation: the comparison on each dataset is split in 3 method groups. Performance is evaluated by test F1-score per training data size. Error bars indicate the upper and lower quintiles among 10 runs.

provide enough data for the DNN model, nor can it be too large that more data is labeled than needed. For example, Songs dataset can converge faster with only a few dozens of pairs while Abt-Buy needs a larger budget to show considerable improvements. This is true regardless of the AL method applied. We use a budget of 100 examples for publications datasets, 20 for Songs and 10% for Abt-Buy.

AL starts with an instance of the deepmatcher model trained on the initial seed set. Every AL iteration consists of running the sampling strategy of choice, retrieving the

Table 2
Dataset statistics.

Dataset	Train			Validation			Test		
	Match	Unmatch	Total	Match	Unmatch	Total	Match	Unmatch	Total
Citeseer-DBLP	118	882	1000	222	1778	2000	827	6173	7000
DBLP-Scholar	3207	14016	17223	1070	4672	5742	1070	4672	5742
Abt-Buy	616	5127	5743	206	1710	1916	206	1710	1916
Songs	3655	8124	11779	1217	2710	3927	1236	2691	3927

labels for the query from the dataset (simulating a human oracle), appending the newly labeled data to the training set and finally retraining a new model from scratch on the full labeled data so far. In each iteration, the risk features are re-extracted from the labeled data and their distributions are re-estimated. Then, the risk model is re-trained on the validation data. Finally, the risk scores for unlabeled data are estimated by the risk model and they are actively sampled by Algorithms 1.

The default hyper-parameters and loss functions for the deepmatcher model training were used as presented in [35]. The deepmatcher model is trained for 20 epochs with a batch size of 32 pairs using the Adam optimizer with a learning rate of 10^{-3}. The risk model is similarly optimized using the Adam optimizer with a learning rate of 10^{-3}, and VaR confidence is set to 0.9. It is trained for 100 epochs with a batch size of 100 pairs. To overcome the randomness caused by different model initializations and training data shuffling, we perform 10 training sessions and report the mean test F1-score. For fair comparison, we make sure that all the methods use the same set of model initializations. For the approaches that require access to the classifier (all except *Random*), we use the model with the best validation performance.

5.2. Comparative Evaluation

The evaluation results have been presented in Fig. 3. Due to the large number of compared methods, we report their performance on each test dataset in three separate subfigures.

It can be observed that random sampling has the overall lowest performance. This confirms the need for active selection. The simple uncertainty method of maximum entropy achieves highly competitive performance on most of the test datasets, e.g. Abt-Buy, Citeseer-DBLP and Songs. While the other uncertainty method of BALD shows slightly higher performance than deterministic maximum entropy on some datasets. However, the improvement is not sufficiently consistent, possibly due to the quality of the MCDropout approximation. On the other hand, ensemble-based uncertainty (ENS) manages to outperform BALD on most datasets and is more stable. It can also be observed that the Core-Set approach can be highly competitive while only considering instance representativeness on most of the test datasets, e.g. Abt-Buy and Citeseer-DBLP. However, purely built upon instance representation, it is not very stable: on Songs, its performance fluctuates wildly. By maximizing the impact

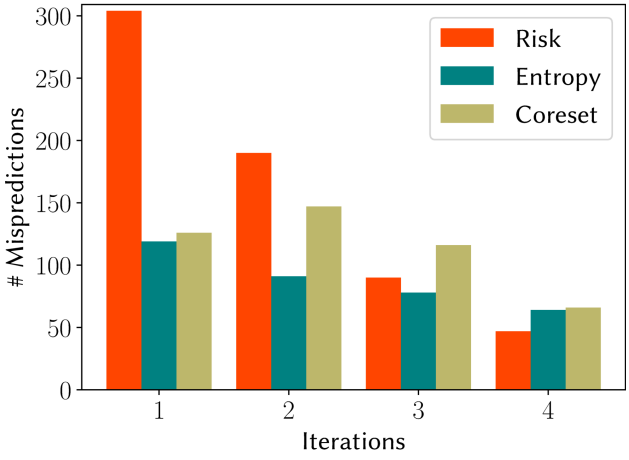
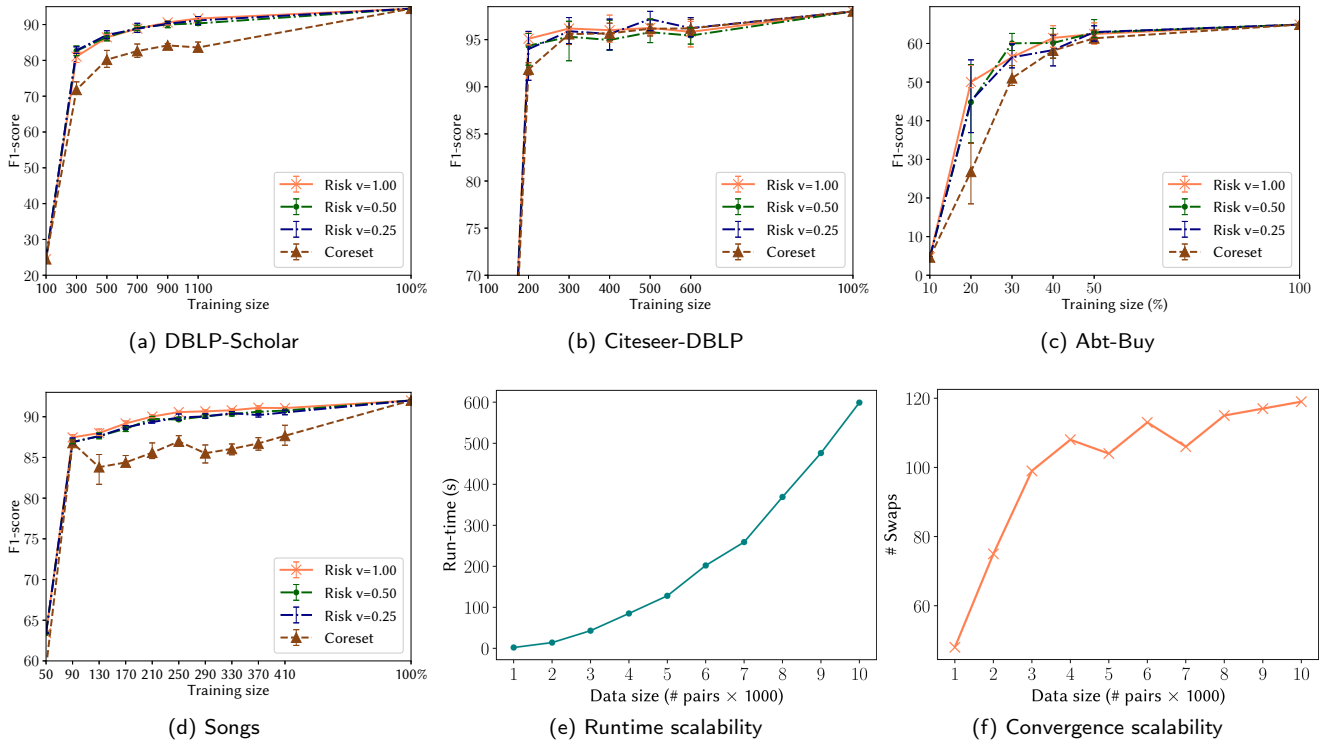


Figure 4: Misprediction selection rate on Abt-Buy.

on the classifier, EGL is also able to positively impact its performance. Similarly, making use of gradient information, BADGE was mostly on par with EGL except on DBLP-Scholar, where the gradient-based diversification gave a better and more stable performance. Althouth CEAL labels more data given the same budget as other methods, thanks to soft-labels, it outperforms uncertainty methods only slightly.

It is clear that risk sampling is able to consistently increase the classifier's performance across the test datasets. It can be observed that the performance margins between risk sampling and alternative methods are considerable in most cases, especially in earlier iterations (low training sizes). This result clearly demonstrates that exposing the classifier to high-risk examples in an early stage can effectively accelerate training. Coupled with the representativeness achieved by core-set clustering, it is able to maintain an advantage over alternative methods. Finally, as shown in Fig. 3, the error bar plots for risk sampling are relatively short, even for the Abt-Buy products dataset which seems to show high variance overall. This means that the data selected via risk sampling yields less variance in the classifiers across random initializations.

An Illustrative Example. The major difference of risk sampling from previous alternatives is the criterion of *misprediction risk*. Therefore, we illustrate the efficacy of risk sampling by examining the number of mispredictions in the selected batches on the Abt-Buy dataset, which is the most challenging one. The results are reported in Fig. 4. It can be seen that risk sampling ends up selecting batches domi-



(a) DBLP-Scholar (b) Citeseer-DBLP (c) Abt-Buy

(d) Songs (e) Runtime scalability (f) Convergence scalability

Figure 5: a-d: Robustness Evaluation of Risk Smapling. Evaluates Risk sampling with different validation data ratios (0.5, 0.75, and 1.). Core-set is plotted for performance reference. **e-f**: Risk Sampling scalability. Evaluates the scalability in terms of runtime and number of swaps.

nated by mispredictions. For reference, maximum entropy, which is likely to select mispredictions (since many uncertain points might turn out to be mispredicted), does not pick up as many as risk sampling. The same can be said about the core-set approach which only considers instance representation. The decreasing number of mispredictions throughout iterations is due to the reduction of such cases in the unlabeled pool that we are sampling from. Combined with the observation on their comparative performance in the first two iterations, these results clearly indicate that misprediction risk is an informative measure for AL.

5.3. Robustness w.r.t Validation Data Size

Since risk sampling leverages validation data, we further investigate its performance robustness w.r.t the size of validation data. To this end, we re-run the AL experiment by varying the validation data ratio used for risk training among 0.25, 0.50 and 1. The results on all datasets are presented in Fig. 5a-d. For performance reference, we also plot the result of the core-set approach in the figure. It can be observed that the performance of risk sampling is overall very robust across ratios, and it consistently outperforms the core-set approach. It is noteworthy that our evaluation results are consistent with those reported in [8], which showed that the performance of LearnRisk is very robust w.r.t the size of validation data. These experimental results bode well for the application of risk sampling in real scenarios.

5.4. Risk Sampling Efficiency

In this section we evaluate the efficiency of the risk sampling algorithm presented in Subsection 4.2. To this end, we evaluate its scalability w.r.t the total data size (n) both in terms of total runtime and number of swaps till convergence. We fix the number of clusters $k = 200$ ($|L| = 100, |Q| = 100$) and variate the data size on the large dataset of DBLP-Scholar using the risk scores and data representations from the first iteration of active learning. The runtimes for the different data sizes are presented in Fig. 5e. Knowing that the algorithm's time complexity of $O(kb(n-k)^2)$ is dependent on $n-k$, it is clear that the combination of a small k (200) and a large n (10000), which produces an extremely high value for $n-k$, still converges in a reasonable time. Moerover, the number of iterations does not exceed k as assumed in Subsection 4.2. Note that the AL iteration runtime is monopolized by the DNN model training, during which the sampling step takes way less time.

Moreover, the plot presenting the number of swaps needed until convergence as a function of data size is given in Fig. 5f. It clearly shows that the number of swaps increases at a slow rate with larger data set size (n). Meaning that the execution time is greatly due to the time needed for the search for each swap.

6. Conclusion

In this work, we propose a novel strategy of risk sampling for active learning that selects representative points with high misclassification risk for labeling. Built upon the core-set characterization for AL, we theoretically derive an optimization model based on an upper-bound of the core-set loss with non-uniform Lipschitz continuity. Due to the NP-hardness of the defined problem, we then present an efficient algorithm for its solution. Finally, our empirical study has validated the efficacy of the proposed approach. For future work, it is worthy to point out that risk sampling is generally applicable to other classification tasks; their technical solutions however need further investigations.

7. Acknowledgements

This work was supported by the National Key Research and Development Program of China [grant number 2018YFB1003400]; the National Natural Science Foundation of China [grant numbers 61732014, 61672432]; the Fundamental Research Funds for the Central Universities [grant number 3102019DX1004]; and the Natural Science Basic Research Plan in Shaanxi Province of China [grant number 2018JM6086].

A. Proof of Lemma 1

We use the following definition of RNN:

$$\mathbf{h_t} = \sigma(W \cdot \mathbf{h_{t-1}} + U \cdot \mathbf{x_t})$$

s.t. $\mathbf{h_0} = \Phi; U \in \mathbb{R}^{m \times m'}, W \in \mathbb{R}^{m' \times m'}$ and σ is an L_σ-Lipschitz activation function. Note that, the commonly used activation functions for RNNs (ex. tanh) are 1-Lipschitz (i.e. $L_\sigma = 1$).

Proof of Lemma 1. Let $X \in \mathbb{R}^{T \times m}$ be an input sequence of size T (i.e. $X = \{\mathbf{x_1}, ..., \mathbf{x_T}\}$). For two distinct inputs X,X' generating hidden states $\mathbf{h_t}$,$\mathbf{h'_t} \in \mathbb{R}^{m'}$ respectively, we have:

$$||\mathbf{h_T} - \mathbf{h'_T}||_p \leq L_\sigma \, ||W||_p \, ||\mathbf{h_{T-1}} - \mathbf{h'_{T-1}}||_p + L_\sigma \, ||U||_p \, ||\mathbf{x_T} - \mathbf{x'_T}||_p$$

By unfolding the right-hand side in the above inequality,

$$||\mathbf{h_T} - \mathbf{h'_T}||_p \leq L_\sigma^T \, ||W||_p^T \, ||\mathbf{h_0} - \mathbf{h'_0}||_p + L_\sigma^T \, ||W||_p^{T-1} \, ||U||_p \, ||\mathbf{x_1} - \mathbf{x'_1}||_p + \cdots + L_\sigma \, ||U||_p \, ||\mathbf{x_T} - \mathbf{x'_T}||_p$$

For $||U||_p, ||W||_p \leq \alpha$,

$$||\mathbf{h_T} - \mathbf{h'_T}||_p \leq \sum_{t=1}^{T} \alpha^{T-t+1} \, L_\sigma^{T-t+1} \, ||\mathbf{x_t} - \mathbf{x'_t}||_p$$

When $p = 2$, for an L_2-regularized and stable RNN [34] ($\alpha \leq 1$) we have $\max_{t \in [1,T]} \alpha^t = \alpha$,

$$||\mathbf{h_T} - \mathbf{h'_T}||_2 \leq \alpha \sum_{t=1}^{T} ||\mathbf{x_t} - \mathbf{x'_t}||_2$$

Then by applying Cauchy-Schwartz inequality,

$$||\mathbf{h_T} - \mathbf{h'_T}||_2 \leq \alpha \sqrt{T} \, ||X - X'||_F$$

For a fully-connected network module F_C with n_{fc} layers, trainable parameters w_{fc} and σ activations, followed by a C-class Softmax function [40],

$$||F_C(\mathbf{h_T}, w_{fc}) - F_C(\mathbf{h'_T}, w_{fc})||_2 \leq \frac{\sqrt{C-1}}{C} \, \alpha^{n_{fc}+1} \, \sqrt{T} \, ||X - X'||_F$$

For $w = \{w_{fc}, W, U\}$ and knowing that for a matrix $X \in \mathbb{R}^{n \times m}$: $||X||_F \leq \sqrt{m} \, ||X||_2$,

$$||RNN(X, w) - RNN(X', w)||_2 = ||F_C(\mathbf{h_T}, w_{fc}) - F_C(\mathbf{h'_T}, w_{fc})||_2 \leq \frac{\sqrt{(C-1)\,T\,m}}{C} \, \alpha^{n_{fc}+1} \, ||X - X'||_2$$

For any fixed y, using the reverse triangle inequality we get,

$$|l(X, y, w) - l(X', y, w)| = \Big| ||RNN(X, w) - y||_2 - ||RNN(X', w) - y||_2 \Big| \leq ||RNN(X, w) - RNN(X', w)||_2 \leq \frac{\sqrt{(C-1)\,T\,m}}{C} \, \alpha^{n_{fc}+1} \, ||X - X'||_2$$

B. Proof of Theorem 1

Here, we study the Lipschitz continuity for the DNN model defined in Definition 2. We suppose a distance function $F_D(\overrightarrow{s_k}, \overrightarrow{s_k}) = \left|\overrightarrow{s_k} - \overrightarrow{s_k}\right|$ as used by the DeepMatcher model.

Proof of Theorem 1. We start with the expression

$$||\mathbf{s_k} - \mathbf{s'_k}||_2 \leq \alpha \sqrt{T_k} \, ||X^k - X^{k\prime}||_F$$

Let $\tilde{X^k} = \{\overleftarrow{X^k}, \overrightarrow{X^k}\} \in \mathbb{R}^{(\overleftarrow{T_k}+\overrightarrow{T_k}) \times m}$,

$$||\tilde{\mathbf{s_k}} - \tilde{\mathbf{s_k}}'||_2 \leq ||\overleftarrow{s_k} - \overleftarrow{s_k}'||_2 + ||\overrightarrow{s_k} - \overrightarrow{s_k}'||_2 \leq \alpha \Big(\sum_{t=1}^{\overleftarrow{T_k}} ||\overleftarrow{x_t} - \overleftarrow{x_t}'||_2 + \sum_{t=1}^{\overrightarrow{T_k}} ||\overrightarrow{x_t} - \overrightarrow{x_t}'||_2 \Big) \leq \alpha \sqrt{\overleftarrow{T_k} + \overrightarrow{T_k}} \, ||\tilde{X^k} - \tilde{X^k}'||_F$$

Finally, the classifier module F_C takes in the concatenated similarities $S = \{\tilde{\mathbf{s_k}}\}_{k=1}^{n_A}$. Let $X_d = \{X^k\}_{k=1}^{n_A} \in \mathbb{R}^{T_d \times m}$ be the representation for pair d, s.t $T_d = \sum_{k=1}^{n_A} (\overleftarrow{T_k} + \overrightarrow{T_k})$. And let $\hat{T} = \max_{d_i} T_{d_i}$ be the maximal pair length in D. Then, the resulting similarity matrix satisfies,

$$||S - S'||_F \leq \alpha \sqrt{\hat{T}} \, ||X_d - X'_d||_F$$

The final expression for the loss function following the same steps as in the proof of Lemma 1 and setting $C = 2$:

$$|l(d, y, w) - l(d', y, w)| \leq \frac{\alpha^{n_{fc}+1}}{2} \, \sqrt{\hat{T}\,m} \, ||X_d - X'_d||_2$$

C. Proof of Theorem 2

PROOF OF THEOREM 2. Let $(d_i, y_i) \in U$, $(d_j, y_j) \in L$ be an unlabeled and a labeled pair respectively. Let $l(d, y)$ be an $\mathbb{L}$-Lipschitz continuous loss function for any pair d with ground-truth label y w.r.t the model $A_{L\cup Q}$ trained on $L \cup Q$. We have:

$$|l(d_i, y_i) - l(d_j, y_j)| \leq \mathbb{L}_i \, ||X_{d_i} - X_{d_j}||_2$$

Where $\mathbb{L}_i$ represents the Lipschitz bound over the slope of the loss landscape between d_i and d_j ($\mathbb{L}_i \leq \mathbb{L}$). Let $\{C_1, C_2, \ldots, C_{|L\cup Q|}\}$ represent a clustering of D ($D = \bigcup_j C_j$) where each cluster C_j is centered around $(d_j, y_j) \in C_j$. Using triangle inequality and summing over $(d_i, y_j) \in C_j$,

$$\left| \sum_{(d_i,y_i)\in C_j} l(d_i, y_i) - |C_j| \cdot l(d_j, y_j) \right| \leq \sum_{(d_i,y_i)\in C_j} \mathbb{L}_i \, ||X_{d_i} - X_{d_j}||_2$$

By summing over all clusters C_j and applying triangle inequality, then multiplying both sides by $\frac{1}{n}$,

$$\left| \frac{1}{n} \sum_{(d_i,y_i)\in D} l(d_i, y_i) - \frac{1}{n} \sum_{(d_j,y_j)\in L\cup Q} |C_j| \, l(d_j, y_j) \right| \leq \frac{1}{n} \sum_{(d_j,y_j)\in L\cup Q} \sum_{(d_i,y_i)\in C_j} \mathbb{L}_i \, ||X_{d_i} - X_{d_j}||_2$$

Assuming zero loss for labeled data, i.e. $\forall (d_j, y_j) \in L \cup Q : l(d_j, y_j) = 0$, the cluster-weighted loss average and the simple loss average are equal, yielding:

$$\left| \frac{1}{n} \sum_{(d_i,y_i)\in D} l(d_i, y_i) - \frac{1}{|L \cup Q|} \sum_{(d_j,y_j)\in L\cup Q} l(d_j, y_j) \right| \leq \frac{1}{n} \sum_{(d_j,y_j)\in L\cup Q} \sum_{(d_i,y_i)\in C_j} \mathbb{L}_i \, ||X_{d_i} - X_{d_j}||_2$$

CRediT authorship contribution statement

Youcef Nafa: Conceptualization, Formal analysis, Writing - Original Draft . **Qun Chen:** Conceptualization, Supervision, Writing - Review & Editing. **Zhaoqiang Chen:** Methodology. **Xingyu Lu:** Software, Investigation. **Haiyang He:** Software, Investigation. **Tianyi Duan:** Software. **Zhanhuai Li:** Project administration, Funding acquisition.

References

[1] Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A., 2020. Deep batch active learning by diverse, uncertain gradient lower bounds., in: ICLR. URL: https://openreview.net/forum?id=ryghZJBKPS.

[2] Bates, S., Angelopoulos, A., Lei, L., Malik, J., Jordan, M.I., 2021. Distribution-free, risk-controlling prediction sets. arXiv:2101.02703.

[3] Beluch, W.H., Genewein, T., Nürnberger, A., Köhler, J.M., 2018. The power of ensembles for active learning in image classification, in: CVPR, pp. 9368–9377. URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Beluch_The_Power_of_CVPR_2018_paper.html.

[4] Bıyık, E., Wang, K., Anari, N., Sadigh, D., 2019. Batch Active Learning Using Determinantal Point Processes. arXiv e-prints arXiv:1906.07975.

[5] Bogatu, A., Paton, N.W., Douthwaite, M., Davie, S., Freitas, A., 2020. Cost-effective variational active entity resolution. arXiv:2011.10406.

[6] Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G., 2005. Learning to rank using gradient descent, in: Proceedings of the 22nd International Conference on Machine Learning, Association for Computing Machinery, New York, NY, USA. p. 89–96. URL: https://doi.org/10.1145/1102351.1102363, doi:10.1145/1102351.1102363.

[7] Chen, Z., Chen, Q., Hou, B., Ahmed, M., Li, Z., 2018. Improving machine-based entity resolution with limited human effort: A risk perspective, in: Proceedings of the International Workshop on Real-Time Business Intelligence and Analytics, Association for Computing Machinery. doi:10.1145/3242153.3242156.

[8] Chen, Z., Chen, Q., Hou, B., Li, Z., Li, G., 2020. Towards interpretable and learnable risk analysis for entity resolution, in: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, Association for Computing Machinery. p. 1165–1180. doi:10.1145/3318464.3380572.

[9] Christen, P., 2008. Automatic record linkage using seeded nearest neighbour and support vector machine classification, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 151–159. doi:10.1145/1401890.1401913.

[10] Christen, P., 2012. Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer Publishing Company, Incorporated. doi:10.1007/978-3-642-31164-2.

[11] Christophides, V., Efthymiou, V., Stefanidis, K., 2015. Entity Resolution in the Web of Data. Morgan & Claypool Publishers.

[12] Corbière, C., Thome, N., Bar-Hen, A., Cord, M., Pérez, P., 2019. Addressing Failure Prediction by Learning Model Confidence. Curran Associates Inc., Red Hook, NY, USA.

[13] Ducoffe, M., Precioso, F., 2018. Adversarial active learning for deep networks: a margin based approach. ArXiv abs/1802.09841.

[14] Ebraheem, M., Thirumuruganathan, S., Joty, S., Ouzzani, M., Tang, N., 2018. Distributed representations of tuples for entity resolution. Proc. VLDB Endow. 11, 1454–1467. doi:10.14778/3236187.3236198.

[15] Elhamifar, E., Sapiro, G., Yang, A., Sasrty, S.S., 2013. A convex optimization framework for active learning, in: 2013 IEEE International Conference on Computer Vision, pp. 209–216.

[16] Fu, C., Han, X., Sun, L., Chen, B., Zhang, W., Wu, S., Kong, H., 2019. End-to-end multi-perspective matching for entity resolution, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization. pp. 4961–4967. doi:10.24963/ijcai.2019/689.

[17] Gal, Y., Ghahramani, Z., 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: Proceedings of The 33rd International Conference on Machine Learning, pp. 1050–1059.

[18] Gissin, D., Shalev-Shwartz, S., 2019. Discriminative active learning. arXiv preprint arXiv:1907.06347 .

[19] Hendrycks, D., Gimpel, K., 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks, in: 5th International Conference on Learning Representations.

[20] Hou, B., Chen, Q., Chen, Z., Nafa, Y., Li, Z., 2020. r-humo: A risk-aware human-machine cooperation framework for entity resolution with quality guarantees. IEEE Transactions on Knowledge and Data Engineering 32, 347–359. doi:10.1109/TKDE.2018.2883532.

[21] Houlsby, N., Huszár, F., Ghahramani, Z., Lengyel, M., 2011. Bayesian active learning for classification and preference learning. arXiv preprint arXiv:1112.5745 .

[22] Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J.E., Weinberger, K.Q., 2017. Snapshot ensembles: Train 1, get M for free, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net. URL: https://openreview.net/forum?id=BJYwwY9ll.

[23] Huang, J., Child, R., Rao, V., Liu, H., Satheesh, S., Coates, A., 2016. Active learning for speech recognition: the power of gradients. arXiv preprint arXiv:1612.03226 .

[24] Isele, R., Bizer, C., 2013. Active learning of expressive linkage rules using genetic programming. Journal of web semantics 23, 2–15.

[25] Jiang, H., Kim, B., Guan, M.Y., Gupta, M., 2018. To trust or not to trust a classifier, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, Curran Associates Inc.. p. 5546–5557.

[26] Kasai, J., Qian, K., Gurajada, S., Li, Y., Popa, L., 2019. Low-resource deep entity resolution with transfer and active learning, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy. pp. 5851–5861. URL: https://www.aclweb.org/anthology/P19-1586, doi:10.18653/v1/P19-1586.

[27] Kaufman, L., Rousseeuw, P.J., 1987. Clustering by means of medoids.

[28] Kirsch, A., van Amersfoort, J., Gal, Y., 2019. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning, in: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 7026–7037.

[29] Lewis, D.D., Gale, W.A., 1994. A sequential algorithm for training text classifiers, in: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Springer-Verlag, Berlin, Heidelberg. p. 3–12.

[30] Li, L., Li, J., Gao, H., 2015. Rule-based method for entity resolution. IEEE Transactions on Knowledge and Data Engineering 27, 250–263.

[31] Li, Y., Li, J., Suhara, Y., Doan, A., Tan, W.C., 2020. Deep entity matching with pre-trained language models. Proceedings of the VLDB Endowment 14, 50–60. doi:10.14778/3421424.3421431.

[32] Meduri, V.V., Popa, L., Sen, P., Sarwat, M., 2020. A comprehensive benchmark framework for active learning methods in entity matching, in: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, Association for Computing Machinery. p. 1133–1147. doi:10.1145/3318464.3380597.

[33] Megiddo, N., Supowit, K., 1984. On the complexity of some common geometric location problems. SIAM J. Comput. 13, 182–196.

[34] Miller, J., Hardt, M., 2019. Stable recurrent models, in: International Conference on Learning Representations. URL: https://openreview.net/forum?id=Hygxb2CqKm.

[35] Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., Deep, R., Arcaute, E., Raghavendra, V., 2018. Deep learning for entity matching: A design space exploration, in: Proceedings of the 2018 International Conference on Management of Data.

[36] Nie, H., Han, X., He, B., Sun, L., Chen, B., Zhang, W., Wu, S., Kong, H., 2019. Deep sequence-to-sequence entity matching for heterogeneous entity resolution, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Association for Computing Machinery. p. 629–638. doi:10.1145/3357384.3358018.

[37] Qian, K., Popa, L., Sen, P., 2017. Active learning for large-scale entity resolution, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Association for Computing Machinery. p. 1379–1388. doi:10.1145/3132847.3132949.

[38] Sarawagi, S., Bhamidipaty, A., 2002. Interactive deduplication using active learning, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 269–278.

[39] Schubert, E., Rousseeuw, P.J., 2019. Faster k-medoids clustering: Improving the pam, clara, and clarans algorithms, in: Amato, G., Gennaro, C., Oria, V., Radovanović, M. (Eds.), Similarity Search and Applications, Springer International Publishing. pp. 171–187.

[40] Sener, O., Savarese, S., 2018. Active learning for convolutional neural networks: A core-set approach, in: International Conference on Learning Representations. URL: https://openreview.net/forum?id=H1aIuk-RW.

[41] Settles, B., 2012. Active Learning. Morgan & Claypool Publishers.

[42] Singh, R., Meduri, V., Elmagarmid, A., Madden, S., Papotti, P., Quiané-Ruiz, J.A., Solar-Lezama, A., Tang, N., 2017. Generating concise entity matching rules, in: Proceedings of the 2017 ACM International Conference on Management of Data, p. 1635–1638.

[43] Singla, P., Domingos, P., 2006. Entity resolution with markov logic, in: Sixth International Conference on Data Mining (ICDM'06).

[44] Sinha, S., Ebrahimi, S., Darrell, T., 2019. Variational adversarial active learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).

[45] Tardivo, G., 2002. Value at risk (var): The new benchmark for managing market risk. Journal of Financial Management & Analysis 15.

[46] Tran, T., Do, T.T., Reid, I., Carneiro, G., 2019. Bayesian generative active deep learning, PMLR. pp. 6295–6304. URL: http://proceedings.mlr.press/v97/tran19a.html.

[47] Wang, K., Zhang, D., Li, Y., Zhang, R., Lin, L., 2017. Cost-effective active learning for deep image classification. IEEE Transactions on Circuits and Systems for Video Technology 27, 2591–2600. doi:10.1109/TCSVT.2016.2589879.

[48] Williamson, R., Menon, A., 2019. Fairness risk measures, in: Chaudhuri, K., Salakhutdinov, R. (Eds.), Proceedings of the 36th International Conference on Machine Learning, PMLR. pp. 6786–6797. URL: https://proceedings.mlr.press/v97/williamson19a.html.

[49] Xiao, Y., Pei, Q., Yao, L., Wang, X., 2020. Recrisk: An enhanced recommendation model with multi-facet risk control. Expert Systems with Applications 158, 113561. URL: https://www.sciencedirect.com/science/article/pii/S0957417420303857, doi:https://doi.org/10.1016/j.eswa.2020.113561.

[50] Yang, L., Zhang, Y., Chen, J., Zhang, S., Chen, D.Z., 2017. Suggestive annotation: A deep active learning framework for biomedical image segmentation, in: Medical Image Computing and Computer Assisted Intervention - MICCAI 2017.

[51] Yang, Y., Loog, M., 2018. A benchmark and comparison of active learning for logistic regression. Pattern Recognition 83, 401–415.

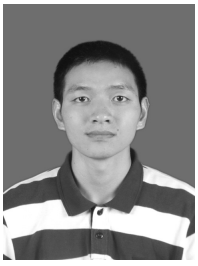
[52] Zhang, P., Wang, J., Farhadi, A., Hebert, M., Parikh, D., 2014. Predicting failures of vision systems, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[53] Zhang, Y., Lease, M., Wallace, B.C., 2017. Active discriminative text representation learning, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pp. 3386–3392.

[54] Zhao, C., He, Y., 2019. Auto-em: End-to-end fuzzy entity-matching using pre-trained deep models and transfer learning, in: The World Wide Web Conference, Association for Computing Machinery. p. 2413–2424. doi:10.1145/3308558.3313578.

Youcef Nafa is a Ph.D student in the School of Computer Science in Northwestern Polytechnical University. His research interests include deep learning and artificial intelligence.

Qun Chen is a professor in the School of Computer Science in Northwestern Polytechnical University. His current research interests include gradual machine learning and risk analysis for AI.



Zhaoqiang Chen is a PhD student in the School of Computer Science, Northwestern Polytechnical University. His research interets include data quality and risk analysis for artificial intelligence.

Xingyu Lu is a master student in the School of Computer Science, Northwestern Polytechnical University. His research interests include artificial intelligence and network security.

Haiyang He is a master student in the School of Computer Science, Northwestern Polytechnical University. His research interests include artificial intelligence and risk analysis.

Tianyi Duan is an undergraduate student in the school of Computer Science, Northwestern Polytechnical University. His research interests include data science and artificial intelligence.

Zhanhuai Li is a professor in the School of Computer Science in Northwestern Polytechnical University. His research interests include data storage and management. He has served as Program Committee Chair or Member in various conferences and committees.