CORDEL: A Contrastive Deep Learning Approach for Entity Linkage

Zhengyang Wang
Texas A&M University
College Station, TX, USA
zhengyang.wang@tamu.edu

Bunyamin Sisman, Hao Wei, Xin Luna Dong
Amazon.com
Seattle, WA, USA
{*bunyamis, wehao, lunadong*}*@amazon.com*

Shuiwang Ji
Texas A&M University
College Station, TX, USA
sji@tamu.edu

***Abstract*—Entity linkage (EL) is a critical problem in data cleaning and integration. In the past several decades, EL has typically been done by rule-based systems or traditional machine learning models with hand-curated features, both of which heavily depend on manual human inputs. With the ever-increasing growth of new data, deep learning (DL) based approaches have been proposed to alleviate the high cost of EL associated with the traditional models. Existing exploration of DL models for EL strictly follows the well-known twin-network architecture. However, we argue that the twin-network architecture is sub-optimal to EL, leading to inherent drawbacks of existing models. In order to address the drawbacks, we propose a novel and generic contrastive DL framework for EL. The proposed framework is able to capture both syntactic and semantic matching signals and pays attention to subtle but critical differences. Based on the framework, we develop a contrastive DL approach for EL, CORDEL, with a simple yet powerful variant called CORDEL-Sum. We evaluate CORDEL with extensive experiments conducted on both public benchmark datasets and a real-world dataset. CORDEL outperforms previous state-of-the-art models by 5.2% on public benchmark datasets. Moreover, CORDEL yields a 29.4% improvement over the current best DL model on the real-world dataset, while reducing the number of training parameters by 96.8%.**

***Keywords*-entity linkage; twin network; deep learning**

I. INTRODUCTION

Entity linkage (EL), also known as entity matching, record linkage, entity resolution, and duplicate detection, refers to the task of determining whether two data records represent the same real-world entity. EL has been a fundamental problem in data cleaning and integration. Models for EL have evolved with the development of machine learning [1]. However, because of the explosion in the volume and diversity of data, we are still far away from solving EL. Newly generated data may have different data distributions, requiring new models and thus a lot of human resources. For example, traditional machine learning models, such as support vector machines and random forests, usually require humans to hand-craft features for different data to maximize the model accuracies.

Compared with traditional machine learning methods, deep learning (DL) is known to be capable of extracting task-specific features from raw data automatically through the learning process. In addition, the development of distributed representations enables DL models to process textual data directly [2]. These properties are highly desirable for EL.

Our work is not the first DL approach for EL. Existing DL methods for EL [3]–[6] employ the twin-network architecture in Figure 1(a), which is commonly used for other matching tasks in NLP in the literature. In NLP, the twin-network architecture is usually employed for semantic matching tasks such as question answering that require matching abstract text representations. However, semantic matching is not effective on many EL tasks. For example, in product EL tasks, the record pair (Black ink tank, Canon) and (Cyan ink tank, Canon), where the attributes are (Product title, Brand), is a non-match since they have different colors. However, the words representing different colors are semantically close to each other, making it difficult to distinguish this pair based on semantic matching. Another example is the record pair (Coca-Cola 12 fl oz 8 pack, Coca-Cola) and (Coca-Cola 12 fl oz 6 pack, Coca-Cola), where the only difference lies in the number of bottles in a pack. It is a non-match as well, even though words representing numbers have similar semantic meanings. In addition to these non-match cases, semantic matching could also fail on matches. For instance, the beer product record pair (Amber ale, Third Base Sports Bar & Brewery) and (American red ale, Third Base Sports Bar & Brewery) is a match. But the word 'American' in one record is not semantically similar to any word in the other record, which may confuse semantic matching models. Besides these examples, recent studies have also shown that deep neural networks work like low-pass filters and have the effect of smoothing out small differences [7]. Since the comparisons in the twin-network architecture is made after the records are projected onto the embedding space, small but crucial differences may be ignored, resulting in failures.

Because of these limitations of the twin-network architecture, existing DL models for EL do NOT show consistently improved performance over current non-DL machine learning models on EL tasks. The fact that DL models may cause decreased performance in some cases hinders the use of these models for EL in practice.

In order to develop more effective and practical DL models for EL, we propose to jump out of the existing DL framework based on the twin-network architecture. Instead, we propose a new contrastive DL framework for EL, as

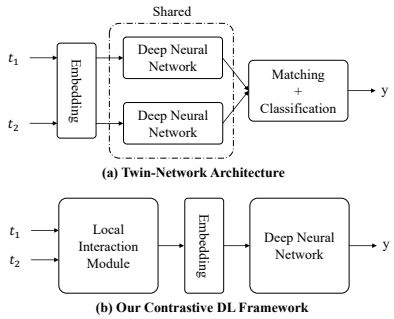

Figure 1. Two types of DL architectures for matching tasks. (a) The twin architecture employed by existing DL models for EL. (b) Our proposed contrastive DL framework for EL followed by CORDEL.

shown in Figure 1(b)[1]. In contrast to the twin-network architecture, our framework is able to capture both syntactic and semantic signals. More importantly, our framework avoids the smoothing effect of deep neural networks and pays attention to subtle but critical differences. As an instantiation of this contrastive DL framework, we build a powerful DL model called CORDEL (COntRastive Deep Entity Linkage)[2]. Our contributions can be summarized in three aspects:

- We propose a novel and generic contrastive DL framework for EL, as shown in Figure 1(b). Our contrastive framework addresses the limitations of the twin-network architecture in Figure 1(a) by capturing both syntactic and semantic signals and paying attention to subtle but critical differences between entities.
- We propose a powerful DL model called CORDEL (COntRastive Deep Entity Linkage) as an instantiation of our proposed contrastive DL framework, as illustrated in Figure 2. Concretely, we develop a simple yet powerful variant of CORDEL, called CORDEL-Sum.
- We perform extensive experiments on both public benchmark datasets and a large real-world dataset. CORDEL is able to outperform previous state-of-the-art models by 5.2% on public benchmark datasets. CORDEL also yields a 29.4% improvement over the current best DL model on the real-world dataset, while reducing 96.8% training parameters. In addition, CORDEL shows great stability over different runs. These results indicate that CORDEL is a reliable, efficient, and effective DL approach for EL.

II. Related Work

In the literature, most existing DL models for EL follow the twin-network architecture in Figure 1(a) [3]–[6]. DEEPMATCHER [3] proposed a general twin-network template of DL models for EL, with four different instantiations: SIF, RNN, Attention, and Hybrid. DEEPER [4] shared high similarities with the SIF and RNN versions of DEEPMATCHER in terms of both the network architectures and performance. Seq2SeqMatcher [5] augmented the twin-network architecture by proposed a sequence-to-sequence alignment layer, which shared certain similarities with the DEEPMATCHER-Attention. AutoEM [6] explored the transfer learning settings while still employing twin-network based DL models.

III. Method

A. Problem Definition

We focus on EL that refers to the matching task between two data records. In detail, data records are saved by following a certain schema. That is, given an ordered set of pre-defined attributes, data are stored by putting its values under corresponding attributes. For example, the product record (Black ink tank, Canon) is saved with pre-defined attributes (Product title, Brand).

Formally, given pre-defined attributes $A_1, A_2, \ldots, A_m$, a data record t can be represented as a tuple $(t[A_1], t[A_2], \ldots, t[A_m])$, where $t[A_i]$, $i = 1, 2, \ldots, m$ refers to the value of the attribute A_i in the record t. In an EL dataset, all the records should have the same schema, that is, the same set of attributes in the same order. The EL task is to determine whether a pair of records t_1 and t_2, where $t_1 \neq t_2$, refer to the same real-world entity. It is formulated as a binary classification problem:

$$y = F(t_1, t_2) \in \{0, 1\}, \tag{1}$$

where F represents a model for EL that outputs a binary prediction y. In practice, it is common to let F first output a continuous number $y \in [0, 1]$, and set a threshold to translate it into the binary classification result. The continuous output is called the matching score and can be interpreted as the likelihood of t_1 and t_2 being a match.

B. Contrastive DL Framework

We first propose a novel and generic contrastive DL framework specially designed for EL, upon which we develop CORDEL. The framework is illustrated in Figure 1(b). We describe it component by component in this section.

Local interaction module (LIM): In order to allow syntactic signal to be captured, our contrastive DL framework avoids projecting inputs into the embedding space at the beginning. Instead, it first employs a LIM to enable the two input records to interact with each other in the raw string level. The LIM compares and contrasts the input records in terms of string tokens, where the tokens can be characters, words, and phrases. After the LIM, all the string tokens from two input records are re-grouped, where each group captures specific syntactic signals. As a result, the outputs of the LIM are simply several groups of string tokens. Our instantiation,

[1]Our contrastive DL framework does not correspond to the contrastive learning in the fields of deep metric learning and self-supervised learning. The "contrastive" here refers to contrasting one input to the other in the raw string level, as explained in Sections III-B and III-C.

[2]An extended version of our work is available at https://arxiv.org/abs/2009.07203.

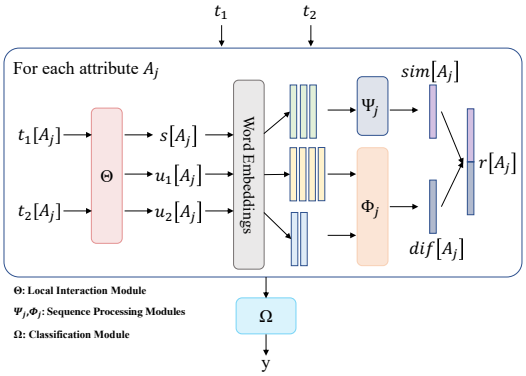

Figure 2. An illustration of our CORDEL described in Section III-C. It follows the proposed contrastive DL framework introduced in Section III-B. We provide simple yet powerful options for Ψ_j, Φ_j, and Ω in Section III-D, leading to CORDEL-Sum.

CORDEL, explores a simple LIM that simply separates the different words from the shared words appearing in both records, as introduced in Section III-C.

Embedding: With syntactic signals captured by the LIM through grouping, distributed embeddings [2] of string tokens allow semantic signals to be taken into consideration by the following deep neural network. Therefore, our framework has an embedding layer after the LIM, which transforms each string token into a numeric vector embedding through distributed representations. The outputs of the embedding layer are thus sequences of vector embeddings corresponding to groups of string tokens. Note that, as the syntactic signals are encoded by the grouping, they will not be lost through the embedding layer. In other words, both syntactic and semantic signals are captured in the outputs of the embedding layer.

Deep neural network: Finally, a deep neural network is applied on top of the embedding layer to process both syntactic and semantic signals and make the prediction. As the inputs are sequences of vector embeddings, the deep neural network can be decomposed into three parts: sequence processing, information aggregation, and classification. First, for each group of vector embeddings, a sequence processing module is employed to summarize the information into a fixed-size vector representation. Next, the information from different groups needs to be aggregated, and then serves as inputs to a classification module.

The proposed contrastive DL framework is the first DL framework for EL that considers both syntactic and semantic signals. In the next section, we propose a powerful DL model as an instantiation of this framework, called CORDEL.

C. An Instantiation - CORDEL

An illustration of the proposed CORDEL (CONtRastive Deep Entity Linkage) is provided in Figure 2. Specifically, under our proposed contrastive DL framework for EL, we develop a simple yet effective LIM followed by a carefully designed deep neural network.

Local interaction module (LIM): The LIM of CORDEL is designed based on human intuition: given an input record pair, we tend to treat the differences between two records as signals for a non-match, and regard the common part as signals for a match. Therefore, our LIM simply separates the different words from the shared words appearing in both records. This results in re-clustering the tokens into three groups: two groups of unique words in either record, and one group of shared words. Specifically, the proposed LIM is achieved through simple set operations, as described below.

Formally, let t_1 and t_2 denote the input record pair, where $t_i = (t_i[A_1], t_i[A_2], \ldots, t_i[A_m])$, $i = 1, 2$, and each attribute value $t_i[A_j]$, $i = 1, 2$, $j = 1, 2, \ldots, m$, is a sequence of words. Our LIM Θ of CORDEL contrasts attribute-wise local tokens. For each attribute A_j, $j = 1, 2, \ldots, m$, the two sequences of words $t_1[A_j]$ and $t_2[A_j]$ are compared in terms of the exact matching between token sets. After Θ, the tokens in $t_1[A_j]$ and $t_2[A_j]$ are distributed into three groups:

$$(s[A_j], u_1[A_j], u_2[A_j]) = \Theta(t_1[A_j], t_2[A_j]), \tag{2}$$

where $s[A_j]$ contains shared words appearing in both $t_1[A_j]$ and $t_2[A_j]$, and $u_i[A_j]$, $i = 1, 2$, includes the unique words that are only in $t_i[A_j]$. In other words, the comparison step Θ can be written as

$$\begin{aligned} s[A_j] &= t_1[A_j] \cap t_2[A_j], \\ u_1[A_j] &= t_1[A_j] \setminus s[A_j], \\ u_2[A_j] &= t_2[A_j] \setminus s[A_j]. \end{aligned} \tag{3}$$

Embedding: Accordingly, CORDEL employs pre-trained word embeddings to transform the outputs of Θ into word embeddings. Without loss of clarity, the same notations $(s[A_j], u_1[A_j], u_2[A_j])$ are used to denote the corresponding three sequences of word embeddings.

Deep neural network: We introduce the corresponding deep neural network in the order of sequence processing, information aggregation, and classification.

(1) *Sequence processing*: For each attribute A_j, two sequence processing modules, Ψ_j and Φ_j, are used to generate an attribute similarity representation vector $sim[A_j]$ and an attribute difference representation vector $dif[A_j]$ from $(s[A_j], u_1[A_j], u_2[A_j])$, respectively:

$$sim[A_j] = \Psi_j(s[A_j]), \tag{4}$$

$$dif[A_j] = \Phi_j(u_1[A_j], u_2[A_j]). \tag{5}$$

Note that we use one sequence processing module Φ_j to process two groups $u_1[A_j]$ and $u_2[A_j]$ instead of two distinct ones. This is because both groups include different words, which can be viewed as one group as well. Here, the attribute similarity representation vector $sim[A_j]$ encodes information from shared words under the attribute A_j in both records, serving as evidence that supports the prediction

of the input record pair as a match. On the contrary, the attribute difference representation vector $dif[A_j]$ encodes information from different words under the attribute A_j in either record, supporting the opposite prediction.

(2) *Information aggregation*: In order to aggregate information, CORDEL concatenates $sim[A_j]$ and $dif[A_j]$ as the attribute representation vector $r[A_j]$:

$$r[A_j] = \text{Concat}(sim[A_j], dif[A_j]). \tag{6}$$

(3) *Classification*: Finally, a classification module Ω takes all m attribute representation vectors as inputs and performs a binary classification task:

$$y = \Omega(r[A_1], r[A_2], \ldots, r[A_m]) \in [0, 1], \tag{7}$$

where y is the predicted matching score. A threshold can be set to translate the matching scores into binary classification results. The classification module Ω has to merge m vectors first and makes the prediction. In DL, it is common to let Ω output two numbers, use the Softmax function to normalize them, and treat one of them as the y in Eqn. (7). With the true label y^* from the training dataset, CORDEL can be trained with the cross-entropy loss through back-propagation.

D. CORDEL-*Sum*

In order to demonstrate the effectiveness of our proposed CORDEL, we build CORDEL-Sum, an extremely simple variant of CORDEL, by specifying Ψ_j in Eqn. (4), Φ_j in Eqn. (5), and Ω in Eqn. (7).

CORDEL-Sum employs summation followed by a one-layer mutlilayer perceptron (MLP) for both Ψ_j and Φ_j. Summation, although without any training parameters, is a powerful process in DL models for classification tasks [2], [8]. The one-layer MLP is used to perform dimension reduction, which avoids having an excessive number of parameters in the following classification module Ω. Specifically, we have

$$sim[A_j] = \Psi_j(s[A_j]) = \sigma(W^{\Psi_j} \cdot \sum_{s \in s[A_j]} s),$$

$$dif[A_j] = \Phi_j(u_1[A_j], u_2[A_j]) = \sigma(W^{\Phi_j} \cdot \sum_{u \in u_1[A_j] \cup u_2[A_j]} u),$$

where W^{Ψ_j} and W^{Φ_j} represent corresponding one-layer MLPs, and σ refers to an activation function. The bias terms are omitted. In particular, Φ_j sums all the input word embeddings from both sequences of difference words. It is worth noting that the one-layer MLPs are independent for each attribute A_j, leading to $2m$ one-layer MLPs in total.

Afterwards, Ω of CORDEL-Sum is simply implemented as a concatenation of m input vectors followed by a two-layer MLP with two output units:

$$y = \text{MLP}(\text{Concat}(r[A_1], r[A_2], \ldots, r[A_m])). \tag{8}$$

CORDEL-Sum is extremely light-weight yet powerful. The training parameters only lie in $2m$ one-layer MLPs plus a two-layer MLP. As shown in Section IV, CORDEL-Sum achieves significantly improved performance over current non-DL and DL models. The success of CORDEL-Sum demonstrates the power of our proposed CORDEL.

E. Analysis of CORDEL

We analyze the CORDEL and demonstrate its advantages.

By taking the LIM Θ, CORDEL takes syntactic signals from raw strings into consideration. Meanwhile, semantic signals are still captured through word embeddings. On one hand, Θ helps CORDEL avoid mistakes caused by the fact that some semantically similar words are the key evidence for the prediction of a non-match. Taking the example of (Coca-Cola 12 fl oz 8 pack, Coca-Cola) and (Coca-Cola 12 fl oz 6 pack, Coca-Cola), the words '8' and '6' will be put into the groups of unique words in either record, and encoded by the attribute difference representation vector $dif[A_j]$. In the case that '8' and '6' have similar word embeddings as they are semantically close, CORDEL is still able to know that there is a numeric difference between the two input records, while the twin networks are not sensitive to such a difference. On the other hand, CORDEL is also effective in the case that semantically different but unimportant words make the model fail to identify a true match. As the final classifier takes both the attribute similarity representation vector $sim[A_j]$ and the attribute difference representation vector $dif[A_j]$ into consideration, CORDEL is able to determine whether the captured differences serve as important evidence for the prediction.

In addition, CORDEL is unaffected by the smoothing effect of deep neural networks. The differences are isolated from the common parts of the input record pair and processed separately. Therefore, no matter how small the differences are, CORDEL is capable of capturing them.

IV. EXPERIMENTAL STUDIES

In this section, we conduct thorough experiments to evaluate our proposed CORDEL and show its superiority in the following aspects:

- On public benchmark datasets, CORDEL outperforms existing non-DL and DL models.
- On a real-world dataset, CORDEL achieves better performance over existing DL models in terms of two practical evaluation metrics as well as great stability over independent training runs.
- CORDEL is a much more efficient DL approach in terms of required computational resources.

A. Experimental Setup

Baselines: We select non-DL and DL baselines.

- The non-DL baseline is Magellan [1], the state-of-the-art machine learning based approach for EL. In particular, Magellan selects the best classifier from decision tree, random forest, Naive Bayes, support

Table I
STATISTICS OF PUBLIC BENCHMARK DATASETS PROVIDED BY [3] AND OUR REAL-WORLD MUSIC DATASET.

Type	Dataset	Domain	#Pairs	#Matches	#Attrs
Public	BeerAdvo-RateBeer	beer	450	68	4
	iTunes-Amazon	music	539	132	8
	Fodors-Zagats	restaurant	946	110	6
	DBLP-ACM	citation	12,363	2,220	4
	DBLP-Scholar	citation	28,707	5,347	4
	Amazon-Google	software	11,460	1,167	3
	Walmart-Amazon	electronics	10,242	962	5
Real-World	Amazon-Wikipedia	music	∼0.4M	∼0.2M	10

vector machine and logistic regression. The features used in Magellan are designed by experts.

- The DL baseline is DEEPMATCHER [3], which represents a wide range of twin-network based DL models for EL. DEEPMATCHER has four versions, named SIF, RNN, Attention, and Hybrid, with increasing complexity. DEEPER [4] and Seq2SeqMatcher [5] can be regarded as extensions of DEEPMATCHER.

CORDEL: We evaluate CORDEL-Sum in our experiments. As described in Section III-D, the training parameters of CORDEL-Sum only lie in $2m$ one-layer MLPs plus a two-layer MLP, where m is the number of attributes in the dataset. The output dimension is set to 64 for the $2m$ one-layer MLPs. The dimension of the hidden layer in the two-layer MLP is set to 256. For fair comparison, the distributed representations used to transform words into word embeddings are 300-dimensional pretrained FastText embeddings [2], which is the same as DEEPMATCHER [3]. The embeddings are not fine-tuned during training. CORDEL is trained by the Adam optimizer with a learning rate of 0.0001. The training batch size is set to 64 for public datasets and 256 for the real-world dataset.

Datasets: Experiments are performed on public benchmark datasets and a real-world dataset.

Public Benchmark Datasets: We conduct experiments on the public datasets provided by [3]. In particular, we focus on 7 structured EL datasets. These public datasets cover a wide range of EL tasks in different domains. The statistics of these datasets are provided in Table I. Following [3], we divide each dataset into training, validation, and evaluation splits with the ratio of 3:1:1. In the experiments on these public datasets, we follow [3] to employ the F_1 score as the evaluation metric, which allows the direct comparison between our proposed CORDEL and baselines.

Real-world Dataset: We collect a real-world EL dataset in the music domain. Specifically, music records are crawled and sampled from Amazon and Wikipedia [9]. That is, in a record pair t_1 and t_2 from this dataset, t_1 is from Amazon and t_2 is from Wikipedia. We have 10 attributes describing basic information about the music track records. In order to obtain the training dataset, we sample 0.4 million record pairs involving 822,276 distinct entities and employ a noisy strong key to label them. Meanwhile, the testing dataset contains record pairs that are manually labelled by human annotators, ensuring that the evaluation is accurate.

	t1	t2
Title	canon cli-226 black ink tank	canon cli-226 cyan ink tank 4547b001
Category	printers	inkjet printer ink
Brand	canon	canon
Model Number	4546b001	4547b001
Price	13.97	11.99

	t1	t2
Beer Name	Green Lakes Organic Ale	Deschutes Green Lakes Non-Organic Ale
Brew Factory Name	Deschutes Brewery	Deschutes Brewery
Style	American Amber / Red Ale	Amber Ale
ABV	5.20%	6.40%

Figure 3. Case studies on public benchmark datasets. Both of them are non-matches, with subtle but critical differences. CORDEL makes the correct prediction in both cases, while DEEPMATCHER fails.

We adopt more comprehensive and practical evaluation metrics for experiments on this real-world dataset: Area Under the Precision-Recall Curve (PRAUC) and Recall when Precision=95% (R@P=95%). As most EL datasets are imbalanced, PRAUC is known to be more suitable for evaluating binary classifiers on imbalanced datasets [10]. R@P=95% is a practical evaluation metric for EL. Data integration typically requires high precision. That is because a low-precision approach for EL would result in wrongly merges records, causing unrecoverable data loss.

B. Results on Public Datasets

Results on the 7 structured EL datasets are reported in Table II. The results of baselines are provided by [3]. Notably, CORDEL-Sum achieves the state-of-the-art performance on 5 out of 7 datasets. On DBLP-Scholar, CORDEL-Sum is the second best model while the best model DEEPMATCHER-Hybrid has 32x more parameters, as shown in Section IV-C1. On Walmart-Amazon, CORDEL-Sum outperforms all versions of the DL baseline. In terms of the average F_1 scores, CORDEL-Sum yields a 5.2% improvement over the previous state-of-the-art model.

1) Case Studies: We perform case studies to show why CORDEL achieves better performance. Specifically, we examine examples in the testing dataset, where CORDEL makes the correct prediction but DEEPMATCHER fails. Figure 3 provides two representative examples from Walmart-Amazon and BeerAdvo-RateBeer, respectively. Both of them are non-matches, with subtle but critical differences. However, DEEPMATCHER identifies them as matches, indicating its inability to capture those subtle but critical differences between input records. On the contrary, CORDEL has an outstanding ability to handle such cases.

C. Results on the Real-world Dataset

To further demonstrate the advantages of CORDEL over the DL baseline, we perform experiments on a real-world EL dataset, which casts more challenges compared to the public benchmark datasets. In particular, a practical DL

Table II
COMPARISONS BETWEEN CORDEL AND BASELINES ON STRUCTURED EL DATASETS FROM [3] IN TERMS OF THE F_1 SCORE. THE BEST AND SECOND BEST RESULTS ARE HIGHLIGHTED WITH BOLDFACE AND UNDERLINE, RESPECTIVELY. IN PARTICULAR, WHEN CORDEL SETS THE NEW STATE-OF-THE-ART RECORD, THE RELATIVE IMPROVEMENT RATE AGAINST THE PREVIOUS BEST PERFORMANCE IS COMPUTED.

Dataset	Magellan [1]	DEEPMATCHER [3]				CORDEL-Sum
		SIF	RNN	Attention	Hybrid	
BeerAdvo-RateBeer	78.8	58.1	72.2	64.0	72.7	**88.9** $\uparrow_{12.8\%}$
iTunes-Amazon	91.2	81.4	88.5	80.8	88.0	**100.0** $\uparrow_{9.6\%}$
Fodors-Zagats	100.0	100.0	100.0	82.1	100.0	**100.0** $\uparrow_{0.0\%}$
DBLP-ACM	98.4	97.5	98.3	98.4	98.4	**99.2** $\uparrow_{0.8\%}$
DBLP-Scholar	92.3	90.9	93.0	93.3	**94.7**	94.0
Amazon-Google	49.1	60.6	59.9	61.1	69.3	**70.2** $\uparrow_{1.3\%}$
Walmart-Amazon	**71.9**	65.1	67.6	50.0	66.9	68.7
Average F_1	83.1	79.1	82.8	75.7	84.3	**88.7** $\uparrow_{5.2\%}$

Table III
COMPARISONS BETWEEN CORDEL AND BASELINES ON A REAL-WORLD DATASET. THE RELATIVE IMPROVEMENT RATES AGAINST THE PREVIOUS BEST MODEL, DEEPMATCHER-HYBRID, ARE COMPUTED.

Model	PRAUC	R@P=95%	#Params
DEEPMATCHER-SIF	88.1 ± 2.9	43.5 ± 17.0	728,802
DEEPMATCHER-Hybrid	90.5 ± 1.9	52.7 ± 25.1	22,151,812
CORDEL-Sum	**91.6** ± **0.3** $\uparrow_{1.2\%}$	**68.2** ± **2.4** $\uparrow_{29.4\%}$	**713,730** $\downarrow 96.8\%$

approach for EL needs to be stable, *i.e.*, different training runs should lead to similar inference performance. This stability is crucial to make DL models reliable.

In order to evaluate the stability, we repeat each experiment for 10 times independently and report the mean and standard deviation over 10 runs. For the baseline DEEPMATCHER, we choose the simplest DEEPMATCHER-SIF and the most powerful DEEPMATCHER-Hybrid.

The comparisons between CORDEL and DEEPMATCHER are summarized in Table III. CORDEL has better and more stable performance in terms of both PRAUC and R@P=95%.

1) Efficiency Analysis: We compare the number of training parameters between CORDEL and DEEPMATCHER in the last column of Table III. We can see that even the simplest DEEPMATCHER-SIF has more parameters than CORDEL, while CORDEL yields much better performance as shown in the experiments above. In addition, the existing state-of-the-art DL approach, DEEPMATCHER-Hybrid, has millions of training parameters, preventing it from being applied on large-scale datasets. On the contrary, CORDEL is a light-weight and efficient DL approach.

V. CONCLUSIONS

In this work, we propose a novel contrastive DL approach for EL, called CORDEL. We point out the limitations of current twin-network DL models and motivate our work. We perform extensive experiments on both public benchmark datasets and a large real-world dataset. The experimental results show the effectiveness of CORDEL with significant and consistent improvements in performance. Moreover, CORDEL is more efficient as a light-weight DL approach, and more reliable with stable performance.

ACKNOWLEDGMENT

The authors would like to thank Christos Faloutsos, Yifan Ethan Xu, and Jialong Zhang for valuable suggestions.

REFERENCES

[1] P. Konda *et al.*, "Magellan: Toward building entity matching management systems," *Proceedings of the VLDB Endowment*, vol. 9, no. 12, pp. 1197–1208, 2016.

[2] A. Joulin *et al.*, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017, pp. 427–431.

[3] S. Mudgal *et al.*, "Deep learning for entity matching: A design space exploration," in *Proceedings of the 2018 ACM International Conference on Management of Data*, 2018, pp. 19–34.

[4] M. Ebraheem *et al.*, "Distributed representations of tuples for entity resolution," *Proceedings of the VLDB Endowment*, vol. 11, no. 11, pp. 1454–1467, 2018.

[5] H. Nie *et al.*, "Deep sequence-to-sequence entity matching for heterogeneous entity resolution," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 629–638.

[6] C. Zhao and Y. He, "Auto-em: End-to-end fuzzy entity-matching using pre-trained deep models and transfer learning," in *The World Wide Web Conference*, 2019, pp. 2413–2424.

[7] H. NT and T. Maehara, "Revisiting graph neural networks: All we have is low-pass filters," *arXiv preprint arXiv:1905.09550*, 2019.

[8] K. Xu *et al.*, "How powerful are graph neural networks?" in *Proceedings of the International Conference on Learning Representations*, 2019.

[9] Q. Zhu *et al.*, "Collective multi-type entity alignment between knowledge graphs," in *The World Wide Web Conference*. Association for Computing Machinery, 2020.

[10] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, no. 3, 2015.