Empowering Transformer with Hybrid Matching Knowledge for Entity Matching

Wenzhou Dou[1], Derong Shen[1(✉)], Tiezheng Nie[1], Yue Kou[1], Chenchen Sun[2], Hang Cui[3], and Ge Yu[1]

[1] School of Computer Science and Engineering, Northeastern University, Shenyang, China
{shenderong,nietiezheng,kouyue,yuge}@cse.neu.edu.cn
[2] School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China
suncc@email.tjut.edu.cn
[3] University of Illinois at Urbana-Champaign, Champaign, USA
hangcui2@illinois.edu

Abstract. Transformers have achieved great success in many NLP tasks. The self-attention mechanism of Transformer learns powerful representation by conducting token-level pairwise interactions within the input sequence. In this paper, we propose a novel entity matching framework named GTA. GTA enhances Transformer for relational data representation by injecting additional hybrid matching knowledge. The hybrid matching knowledge is obtained via graph contrastive learning on a designed hybrid matching graph, in which the dual-level matching and multiple granularity interactions are modeled. In this way, GTA utilizes the prelearned knowledge of both hybrid matching and language modeling. This effectively empowers Transformer to understand the structural features of relational data when performing entity matching. Extensive experiments on open datasets show that GTA effectively enhances Transformer for relational data representation and outperforms state-of-the-art entity matching frameworks.

Keywords: Entity matching · Transformer · Pretrained language model · Hybrid matching graph · Graph contrastive learning

1 Introduction

Entity matching (EM), also known as entity resolution and record linkage, aims to identify records referring to the same real-world entity. Served as a long-standing critical problem [5] in data integration [9,24] and data cleaning [1], EM has been studied for many years [11] in plenty of fields such as e-commerce, medical treatment, etc. Recently, deep learning technologies achieved great success in database research, and have been a common way to solve EM tasks.

Figure 1 shows an example of EM tasks. Given a candidate record pair, the goal of EM is to determine whether they are referring to the same real-world

A. Bhattacharya et al. (Eds.): DASFAA 2022, LNCS 13247, pp. 52–67, 2022.
https://doi.org/10.1007/978-3-031-00129-1_4

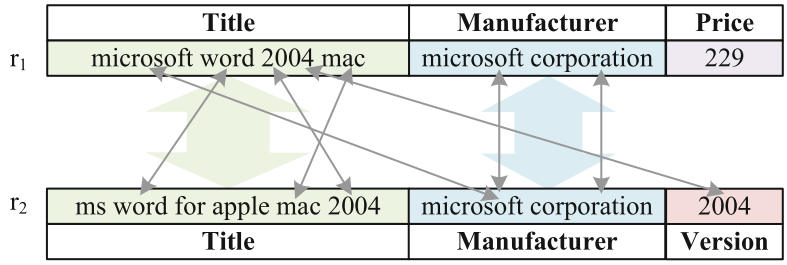

Fig. 1. An example of EM tasks. Records r_1 and r_2 can be seen as a matching pair according to dual levels—attribute level (*title*, *manufacturer*) and token level (*microsoft*, *word*, *2004*, etc.).

entity. Due to the inherent hierarchical structure of the relational records, r_1 and r_2 can be compared at two levels—attribute level and token level. All the neural matching information between attribute values (e.g., values of *title* and *manufacturer*) and tokens (e.g., *microsoft*, *word*, *2004*) jointly determines whether they match or not. Although the two values of attribute *title* are not exactly the same, the same attribute-aligned tokens (e.g., *word*, *2004*, *mac*, etc.) and attribute-unaligned tokens (e.g., *microsoft* in attribute *title* and *manufacturer*), as well as the same values of aligned attributes (e.g., *manufacturer*), provide enough matching signals for the final result.

Depending on the level of comparisons, existing EM works can be divided into three categories: attribute-centric, token-centric, and hybrid-centric. Attribute-centric solutions [12,27] usually follow an alignment-comparison-summarization paradigm. They compare the aligned attributes and aggregate the similarity vectors to form the input to a binary classifier. However, these methods may fall flat when encountering situation like schema heterogeneity (e.g., attributes *price* and *version* in Fig. 1), which is a widespread scenario in real-world applications. Thus, recent works are mainly token-centric [22] or hybrid-centric [15,32,33], which additionally consider token-level matching information to provide EM signals.

Recently, Transformer-based works [2,23,34] have made great progress in EM tasks. They are token-centric solutions that cast EM as a sequence pair classification problem to leverage the pretrained language models (Pretrained LMs or PLMs) like BERT [8], RoBERTa [25] and DistilBERT [30], etc. By representing the candidate record pairs using BERT's [CLS] *RECORD1* [SEP] *RECORD2* [SEP] input schema, they conduct token-level comparisons between the two records via self-attention mechanism. However, vanilla Transformer is not quite suitable for EM tasks, for the reasons that (1) self-attention mechanism is originally designed for the semantic interaction of token level, and (2) the masked language model (MLM) training objective concerns on token-level prediction[1]. These two properties determine that Transformer is good at token-

[1] RoBERTa has proved that removing next sentence prediction (NSP) training objective can improve downstream task performance.

level interaction and less capable of relational data representation, especially for the scenarios in Fig. 1. Since different levels (i.e., attribute and token) contain various abstractions of knowledge [32], relational data's hierarchical feature is important and indispensable in EM tasks.

In this paper, our goal is to overcome the vanilla Transformer's insufficiency for relational data representation by injecting additional hybrid matching knowledge. The modified Transformer is adapted to regard input entries as relational records rather than natural language sentences to perform EM tasks. Thus, the key issues in this paper we need to solve are (1) how to obtain hybrid matching knowledge, i.e., Record-Token-Record (R-T-R) and Record-Attribute-Record (R-A-R), and embed the learned knowledge into representations of attributes and tokens, and (2) how to inject the above hybrid matching knowledge into Transformer for downstream EM tasks. Following this insight, we propose **GTA**, a novel EM framework that comprehensively integrates the hybrid matching knowledge of relational data and the tremendous language knowledge of pretrained LM. GTA is a **G**raph-**T**ransformer-**A**ssembled architecture, which consists of two parts, namely Graph-based Hybrid Embedding (GHE for short) module and Adaptive Transformer-based Matching (ATM for short) module. The GHE module models both attribute-level and token-level matching into the graph topology, and then employs graph contrastive learning [26] to encode structural features into the embeddings of attributes and tokens. And then, the ATM module modifies Transformer's input format and embedding layer to absorb the prelearned knowledge in GHE module. In this way, the input embeddings of Transformer are structure-aware, which is to the benefit of the fine-tuning process of the pretrained LM for EM tasks.

The main contributions can be summarized as follows:

- We propose GTA, a novel EM framework that empowers Transformer with hybrid matching knowledge. GTA is a Graph-Transformer-Assembled architecture, which consists of GHE module for feature learning and ATM module for Transformer's fine-tuning for EM tasks.
- In GHE module, we model the hybrid matching (i.e., Record-Attribute-Record, Record-Token-Record) on the graph topology, ensuring the interactions between multiple granularities (i.e., Attribute-Attribute, Attribute-Token, Token-Token) simultaneously. We then design graph contrastive learning as a pretext task to conduct neural message passing to obtain structure-aware embeddings for attributes and tokens.
- In ATM module, we modify Transformer in two places—input format adaptation and prelearned knowledge injection, enabling Transformer to regard input entries as relational records rather than natural language sentences to perform EM tasks.
- We conduct extensive experiments on structured and dirty datasets and demonstrate that our proposed GTA framework effectively improves Transformer's representation for relational data, and outperforms state-of-the-art EM methods.

2 Related Work

Entity matching has attracted a lot of attention. Existing works can be classified into rule-based EM [7,13,31,37], crowdsourcing-based EM [16,18,36] and learning-based EM [3,12,15,20,23,27,33,34]. Learning-based EM has achieved great success recently. We then briefly introduce them according to the technologies (ML, DL, Transformer) they use.

Magellan [20] is a classical non-neural EM system based on machine learning (ML), which provides a variety of classifiers (decision tree, random forest, and SVM, etc.) to be trained on automatically generated features for EM. The tools it provides can significantly accelerate the entire EM pipeline.

With the rapid development of deep learning (DL) and its success in NLP, researchers introduce DL technologies into EM tasks to compare and aggregate information of relational data. DeepER [12] is one of the earliest methods to adopt word embeddings and LSTM neural networks to train an EM model. DeepMatcher [27] designs a space for DL-based EM, and proves that DL outperforms ML-based solutions on textual and dirty data. MCA [38] is an integrated multi-context attention framework for EM tasks that considers self-attention, pair-attention, and global-attention for three types of context. Both HierMatcher [15] and HAN [33] are end-to-end solutions for EM, which consider hybrid matching information (token and attribute levels). The difference is that HierMatcher performs token-level and attribute-level matching successively, and HAN solves the two in a two-tower mode.

Graph Neural Networks (GNNs) attract great attention due to their success in learning structural features in a lot of areas [14,17,29], thus recent works introduce GNNs to EM tasks. EMBDI [3] is a generic framework for obtaining local embeddings for data integration tasks, which leverages a compact tripartite graph to represent syntactic and semantic relationships between cell values. GraphER [22] encodes the semantic and structural features into an Entity Record Graph (ER-Graph) and trains an Entity Record GCN (ER-GCN) to obtain soft-structural embeddings for EM tasks. And GNEM [4] designs a record pair graph that allows each record pair to interact with relevant records and conducts the pairwise matching decision by borrowing valuable information from other pairs.

Recently, Transformer draws a great deal of concerns in both NLP [8] and CV [10] fields, due to the outstanding performance of self-attention mechanism in acquiring contextual information. Brunner et al. [2] proves the feasibility of adapting the pretrained LMs (e.g., BERT [8]) to EM tasks. DITTO [23] leverages pretrained LMs to solve EM tasks with three additional optimizations—domain knowledge injecting, text summarization, and data augmentation. BERT-ER [21] improves BERT-based EM model by delaying and enhancing BERT's interaction part, together with a blocking module to improve the EM efficiency. RPT [34] proposes a pretrained tuple-to-tuple model that supports several data preparation tasks like data cleaning, entity resolution, and information extraction, etc.

Our work is inspired by this paper [39] and Transformer-based related works [2,23,34]. The personalized dialogue generation model proposed in [39] enriches the dialogue context by additionally encoding the speakers' persona with dia-

logue histories, so as to enhance the Transformer's representation of dialogue context. So we carry on this idea to enhance Transformer for relational data representation and verify its feasibility in EM tasks. The difference is that the additional knowledge we inject into Transformer is obtained in a graph contrastive learning manner and meanwhile encoded into embeddings of attributes and tokens. The knowledge learning process requires no supervision and manual intervention. And then the modified Transformer is adapted to absorb these structure-aware embeddings to perform the downstream EM tasks.

3 Entity Matching via GTA Framework

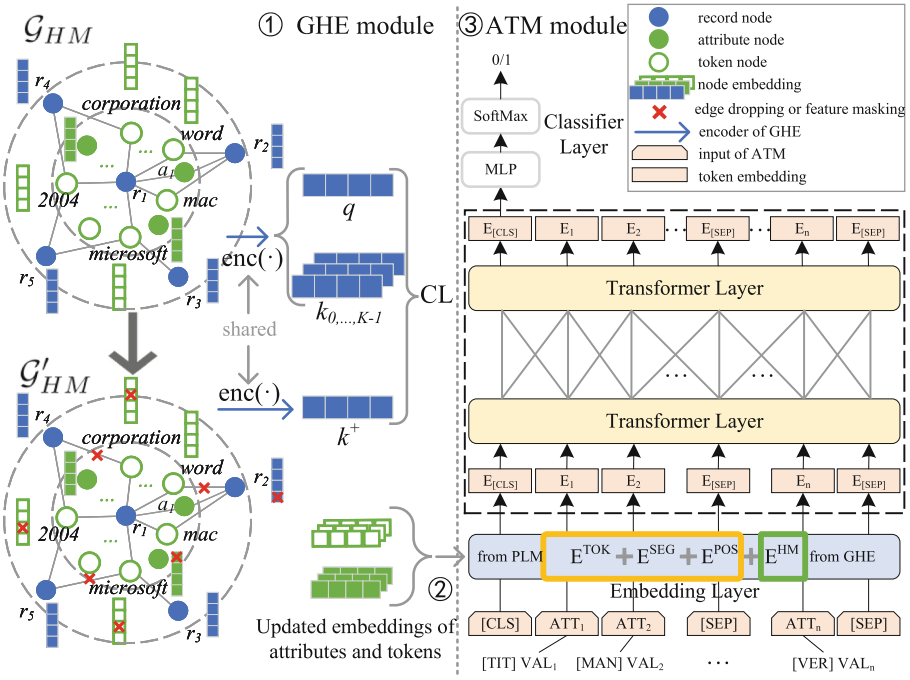

Fig. 2. The framework of GTA. This framework contains a Graph-based Hybrid Embedding (GHE) module for obtaining the hybrid matching knowledge, and an Adaptive Transformer-based Matching (ATM) module for the downstream EM tasks.

The framework of GTA is shown in Fig. 2. The entire workflow is ① GHE module first constructs the hybrid matching graph $\mathcal{G}_{HM}$ and its augmented view $\mathcal{G}'_{HM}$ to model the dual-level matching and multiple granularity interactions, and then performs contrastive learning (CL) on $\mathcal{G}_{HM}$ and $\mathcal{G}'_{HM}$ to obtain the updated embeddings. ② GTA extracts the updated embeddings of attributes and tokens and then maps them to the pretrained LM in ATM module. ③ ATM module

modifies Transformer to absorb these structure-aware embeddings of attributes and tokens and finally fine-tunes the pretrained LM for EM tasks.

We first start with the preliminaries and then introduce the components of GTA framework, including a Graph-based Hybrid embedding (GHE) module and an Adaptive Transformer-based Matching (ATM) module.

3.1 Preliminaries

Let $r_1 = \{a_{11}, a_{12}, ..., a_{1m}\}$ and $r_2 = \{a_{21}, a_{22}, ..., a_{2n}\}$ be a candidate record pair from data sources S and S' separately. Each attribute value a_{1i} (or a_{2j}, $i \in [1, m]$ and $j \in [1, n]$) comprises a sequence of tokens like $\{t_1, t_2, ..., t_T\}$. Each token t_p ($p \in [1, T]$) in the sequence can be one of the string or numeric type.

Definition 1. *Entity matching. Given two relational data sources S and S', an entity matching framework takes as input a pair of records (e.g., r_1 from S and r_2 from S') and outputs the matching probability $P(y = 1|r_1, r_2)$.*

Definition 2. *Hybrid Matching Graph. A hybrid matching graph $\mathcal{G}_{HM} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set. Each node $v \in \mathcal{V}$ can be one of record, attribute, or token. And each edge $e \in \mathcal{E}$ connects a record node with its contained attribute node or token node. $\mathcal{G}_{HM}$ is designed to model the dual-level matching and multiple granularity interactions of attributes and tokens. Each relevant record pairs are connected within two hops by their common attribute nodes or token nodes. And multiple granularity interactions of attribute-attribute, attribute-token, and token-token can also be conducted within two hops via the corresponding record nodes.*

3.2 Graph-Based Hybrid Embedding Module

The GHE module is designed to obtain hybrid matching knowledge in the manner of graph contrastive learning.

Hybrid Matching Graph Construction. As can be seen on the left side of Fig. 2, we first design a hybrid matching graph $\mathcal{G}_{HM}$ to model the relational data's dual-level matching process. In $\mathcal{G}_{HM}$, when two records have common attributes or tokens, they will be indirectly connected within two hops via the corresponding attribute nodes and token nodes, thus the neural matching information can be passed through both attribute level and token level. For instance, the dual-level matching information of r_1 and r_2 can be passed through both (1) attribute value a_1—*microsoft corporation* and (2) tokens—*word* and *mac*.

Beyond that, Fig. 3 shows more comprehensive analyses in attribute and token views. Since the record view in Fig. 3(a) has been analyzed before, we will not dwell on it. In Fig. 3(b) attribute view, the meta-path Attribute-Record-Attribute (e.g., a_1-r_1-a_2) implements the attribute-attribute interaction within a record (e.g., r_1). And the meta-path Attribute-Record-Token (e.g., a_1-r_1-*word*) implements the attribute-token interaction within a record (e.g., r_1). And in Fig. 3(c) token view, the meta-path Token-Record-Token (e.g., *word*-r_1-*mac*) implements the token-token interaction within a record (e.g., r_1). These

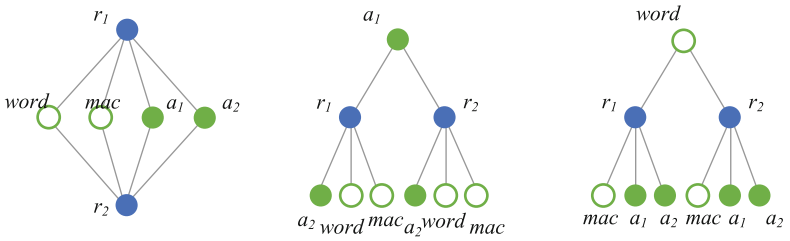

Fig. 3. The neural message passing process in different views—record view, attribute view, and token view.

meta-paths jointly ensure dual-level matching and multiple granularity interactions, enabling the attribute nodes and token nodes to perceive structural features for EM tasks.

Sampling and Training. We adopt contrastive learning as a pretext task to perform graph representation learning on $\mathcal{G}_{HM}$. First, we apply two ways of stochastic perturbation (edge dropping and node feature masking) on $\mathcal{G}_{HM}$, thus we get two views, one is an original view $\mathcal{G}_{HM}$, and the other is an augmented view $\mathcal{G}'_{HM}$. And then, we employ a 2-layer GCN with residual connections as an encoder to obtain node embeddings for both $\mathcal{G}_{HM}$ and $\mathcal{G}'_{HM}$:

$$\mathcal{F}(X^{(l)}) = \sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}X^{(l)}W^{(l)}), \tag{1}$$

$$X^{(l+1)} = \mathcal{F}(X^{(l)}) + \sigma(X^{(l)}), \tag{2}$$

where $\widetilde{A} = A + I$ is the adjacency matrix with self-connections of the hybrid matching graph. $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$ is the degree matrix. $X^{(l)} \in \mathbb{R}^{N\times D}$ is the node embedding matrix and $W^{(l)} \in \mathbb{R}^{D\times D}$ is a trainable weight matrix for the l-th layer. $\sigma(\cdot)$ is a nonlinear activation function, like ELU [6]. The GCN settings are designed based on the following considerations. Since all dual-level matching and multiple granularity interactions can be conducted within two hops on $\mathcal{G}_{HM}$ and $\mathcal{G}'_{HM}$, the 2-layer GCN is suitable for two-hop neural message passing. And the residual connections are designed to prevent the over-smoothing problem.

The sampling strategy for graph contrastive learning is as follows. For each record embedding $q \in X$ in the original view $\mathcal{G}_{HM}$ that acts as a query vector, we sample its corresponding augmented embedding $k_+ \in X'$ in $\mathcal{G}'_{HM}$ as a positive example. As for negative examples, we randomly select K record nodes which are far more than 2 hops (i.e., at least 4 hops) from the original record node, we denote each negative example as $k_i \in X$. The training objective is InfoNCE loss [28]:

$$\mathcal{L}_{HM} = -\log \frac{\exp(q^\top k_+/\tau)}{\sum_{i=0}^{K-1} \exp(q^\top k_i/\tau)}, \tag{3}$$

where τ is the temperature hyperparameter that acts as an adjusting factor to control the strength of penalties on hard negative examples [35].

By closing the distances of the original records with their augmented ones, and meanwhile pulling far from irrelevant negative records, the interactions within the candidate record pairs and their contained attributes and tokens are conducted in the graph contrastive learning process. And the updated attribute and token embeddings are endowed with structural features which help to improve the downstream Transformer-based EM performance.

3.3 Adaptive Transformer-Based Entity Matching

The above GHE module obtains hybrid matching information and multiple granularity interactions in a contrastive learning manner, we then extract and inject the knowledge into a Transformer-based pretrained language model to perform EM tasks. To absorb the prior knowledge, we modify the vanilla Transformer architecture in two places—input format adaptation and prelearned knowledge injection.

Input Format Adaptation. We first convert the raw record pair to a specific format that Transformer can absorb. By appending special tokens [CLS] at the beginning and [SEP] as a separator, we cast EM as a sentence pair classification task that can utilize pretrained knowledge in LMs (e.g., BERT, RoBERTa, and DistilBERT). Apart from that, we add a more fine-grained attribute-specific token before each attribute value to enable Transformer to conduct attribute-level interactions. For example, given a sequence as the value of attribute *title*, we add a special token [TIT] before it to get an attribute-aware entry. Take the record pair in Fig. 1 as an example, after the processing, the record pair can be converted into

[CLS] *RECORD1* [SEP] *RECORD2* [SEP],

where *RECORD1* refers to

[TIT] *microsoft word 2004 mac* [MAN] *microsoft corporation* [PRI] *229*,

and *RECORD2* refers to

[TIT] *ms word for apple mac 2004* [MAN] *microsoft corporation* [VER] *2004*.

So far, the record pairs as Transformer's inputs have been tokenized to accommodate structural perception. The reason for adding attribute-specific tokens (e.g., [TIT]) is that we can slightly enable self-attention mechanism to focus on the interactions of aligned attributes and meanwhile empower the attribute-specific tokens with the prior knowledge learned in the GHE module. The details will be illustrated in the next subsection.

Prelearned Knowledge Injection. After adapting the input format for Transformer, we then describe how to inject the prelearned knowledge into Transformer to perform EM tasks. Transformer-based LMs (e.g., BERT) design input

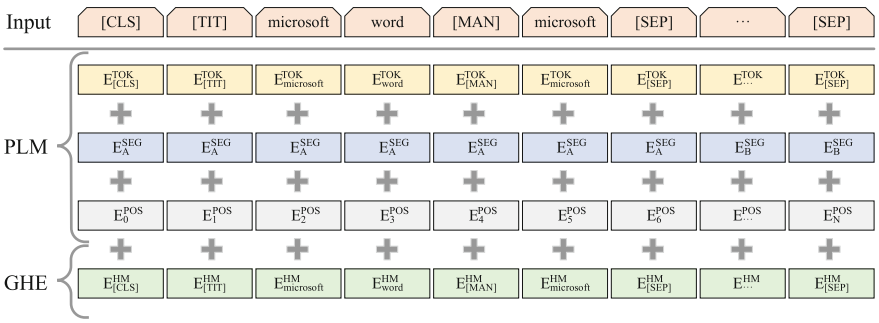

Fig. 4. ATM's input embeddings. The token, segment and position embeddings are inherent in pretrained language model (PLM), and the hybrid matching embeddings are extracted from the graph-based hybrid Embedding (GHE) module. Notice that, some tokens are omitted in this figure.

embedding E^{INPUT} as three components, they are token embedding E^{TOK}, segment embedding E^{SEG}, and position embedding E^{POS}. We retain the above three embeddings and additionally add the hybrid matching embeddings E^{HM} as the supplements to improve Transformer's EM performance.

In detail, we first extract all the updated attribute and token embeddings from GHE module. For different types of E^{HM}, we design different methods for knowledge injection. For example in Fig. 4, the token type embeddings (e.g., $E^{HM}_{microsoft}$ and E^{HM}_{word}) are directly mapped and injected without any additional processing. The attribute-specific embeddings (e.g., $E^{HM}_{[TIT]}$ and $E^{HM}_{[MAN]}$) are generated by averaging all corresponding attribute embeddings in GHE module. And the inherent [CLS] and [SEP] embeddings (i.e., $E^{HM}_{[CLS]}$ and $E^{HM}_{[SEP]}$) are copied from the PLM, which are equal to $E^{TOK}_{[CLS]}$ and $E^{TOK}_{[SEP]}$. Notice that, we do not add E^{HM} to the other three embeddings (i.e., E^{TOK}, E^{SEG} and E^{POS}) directly, but normalize and then scale E^{HM} using the scaling factor ψ before that. This helps to avoid the internal covariate shift problem and control the influence strength of GHE module by adjusting ψ, following the intuition that excessive knowledge injection may collapse the LM's inherence of language modeling.

Fine-Tuning Pretrained LM for EM Tasks. After hybrid matching knowledge injection, we fine-tune the pretrained LM on EM datasets to output the final decisions. The training objective is cross-entropy loss:

$$\mathcal{L}_{CLS} = -\frac{1}{|B|}\sum_{i=1}^{|B|}[y_i \log p_i + (1-y_i)\log(1-p_i)], \tag{4}$$

where $|B|$ is the size of training batch, $y_i \in \{0, 1\}$ is the label of i-th training pair, and p_i is the i-th output probability.

4 Experimental Evaluation

In this section, we evaluate our proposed GTA framework and compare it with existing works.

4.1 Experimental Settings

Datasets and Metric. We evaluate GTA framework on five open datasets[2] proposed in DeepMatcher. The datasets can be divided into two types (structured or dirty) and cover a variety of domains including software, beer, restaurant, and citation. The dataset sizes vary from 450 to 12,363 to evaluate the scalability of GTA. Positive ratios (# Pos./Size) cover the range from 0.10 to 0.18 shows that EM is an unbalanced binary classification task. All EM datasets are split into 60%/20%/20% for training, validation, and test, which are the same as Deepmatcher [27]. The details of EM datasets are shown in Table 1. Following the previous works, we use F1 score as the metric to evaluate the EM performance.

Table 1. Details of EM datasets.

Dataset	Type	Domain	Size	# Pos.	# Att.
Amazon-Google (AG)	Structured	Software	11,460	1,167	3
BeerAdvo-RateBeer (BR)	Structured	Beer	450	68	4
Fodors-Zagats (FZ)	Structured	Restaurant	946	110	6
DBLP-ACM (DA_1)	Structured	Citation	12,363	2,220	4
DBLP-ACM (DA_2)	Dirty	Citation	12,363	2,220	4

Implementation Details and Training Settings. The GTA framework consists of two modules, GHE module and ATM module. For GHE module, we first separate each record into several attribute values. For each sequence of attribute value, we tokenize it with the pretrained tokenizer of the corresponding LM in ATM module. This helps a lot for the token mapping in knowledge injection process. For ease of knowledge injection, we adopt the same embedding dimension $D = 768$ for all the nodes in the hybrid matching graph $\mathcal{G}_{HM}$. We build $\mathcal{G}_{HM}$ and train the GHE module using Deep Graph Library[3]. The details of $\mathcal{G}_{HM}$ are shown in Table 2. The probabilities of stochastic perturbation (i.e., edge dropping and node feature masking) are both 20%, the number of negative samples is $K = 64$, and the temperature hyperparameter τ is 0.1 for the InfoNCE loss. For ATM module, we fine-tune the pretrained LMs (bert-base-uncased, roberta-base, distilbert-base-uncased) to perform EM using Hugging Face[4]. Experiments show that RoBERTa achieves the best EM performance, thus the experiment results we report are all using RoBERTa as the pretrained LM. The whole GTA framework is implemented using PyTorch as a backend.

[2] https://github.com/anhaidgroup/deepmatcher/blob/master/Datasets.md.
[3] https://www.dgl.ai/.
[4] https://huggingface.co/.

Table 2. Details of hybrid matching graph $\mathcal{G}_{HM}$.

Dataset	# Node	# Edge
Amazon-Google (AG)	17,549	50,999
BeerAdvo-RateBeer (BR)	25,969	100,038
Fodors-Zagats (FZ)	7,669	15,878
DBLP-ACM (DA_1)	19,069	98,427
DBLP-ACM (DA_2)	19,435	95,345

We train 500 epochs for GHE module and fine-tune 20 epochs for ATM module. The batch size $B = 64$ and learning rate $lr = 3e - 5$ are set for both GHE and ATM modules. Adam algorithm [19] with warming up and linear decay is used for optimization. The hybrid matching embeddings E^{HM} are scaled by multiplying a scaling factor $\psi = 0.2$ for Fodors-Zagats dataset and $\psi = 0.1$ for other datasets in knowledge injection. We conduct the experiments on a workstation with Intel Xeon W-2255 CPU @ 3.70 GHz and NVIDIA RTX A4000 with 16 GB memory.

The EM frameworks to be compared are as follows:

- **Magellan** [20]**:** A state-of-the-art ML-based (non-DL) system for EM tasks. It provides a variety of classifiers (decision tree, Naive Bayes, SVM, etc.) to be trained on automatically generated features.
- **RNN** [27]**:** A DL-based EM solution proposed in DeepMatcher that adopts Bi-GRU as an encoder to represent attribute values. Then it takes element-wise absolute difference as the comparison result to form the input of the classifier.
- **Attention** [27]**:** A DL-based EM solution proposed in DeepMatcher that adopts decomposable attention to implement attribute summarization and vector concatenation to perform attribute comparison.
- **Hybrid** [27]**:** A DL-based EM solution proposed in DeepMatcher that adopts Bi-GRU with decomposable attention to represent attribute values. Then it takes vector concatenation augmented with element-wise absolute difference as the input of the classifier.
- **MCA** [38]**:** A DL-based EM solution that designs multi-context attention network. MCA fully takes into account self-attention, pair-attention, and global-attention from three types of context.
- **HierMatcher** [15]**:** A DL-based EM solution that jointly considers hierarchical levels of matching granularity (token, attribute, and entity). It designs a cross-attribute token alignment module and attribute-aware attention mechanism that can solve EM in heterogeneous and dirty scenarios.
- **DITTO** [23]**:** A Transformer-based EM solution that leverages domain knowledge, text summarization, and data augmentation to improve pre-trained language models' ability for EM tasks.
- **BERT-ER** [21]**:** A Transformer-based solution that improves EM performance by delaying and enhancing BERT's interaction part, together with an adaptive blocking module to improve EM efficiency.

- **Baseline:** We directly fine-tune the pretrained LM as a baseline. This can be regarded the same as paper [2].

4.2 Main Results

Table 3 shows the results of GTA and existing works, including ML-based, DL-based and Transformer-based EM frameworks. We also set a Baseline which directly fine-tunes the pretrained LM to perform EM tasks without any optimization.

Table 3. EM performance of GTA and existing works. All the experimental results of the comparing methods are derived from the original papers, and the best results are bolded. We calculate $\Delta F1$ between our proposed GTA with Baseline which directly fine-tunes pretrained LM to perform EM tasks.

	AG	BR	FZ	DA_1	DA_2
Magellan	49.1	78.8	100	98.4	91.9
RNN	59.9	72.2	100	98.3	97.5
Attention	61.1	64.0	82.1	98.4	97.4
Hybrid	69.3	72.7	100	98.4	98.1
MCA	70.3	78.8	–	98.6	–
HierMatcher	74.9	–	–	98.8	98.1
DITTO	75.6	94.4	100	99.0	99.0
BERT-ER	75.3	87.5	–	98.7	–
Baseline	74.1	86.7	98.1	98.8	98.9
GTA	**76.2**	**96.3**	**100**	**99.1**	**99.0**
$\Delta F1$	+2.1	+9.6	+1.9	+0.3	+0.1

In detail, we can draw conclusions as below:

- GTA outperforms or reaches state-of-the-art results compared with existing EM frameworks. Compared to ML-based work (i.e., Magellan), GTA achieves 10.5 average F1 improvement, and compared to DL-based work (i.e., Hybrid), GTA achieves 6.4 average F1 improvement. And for HierMatcher, which is a hybrid-centric EM framework, GTA outperforms 1.3, 0.3, 0.9 F1 improvement on *Amazon-Google*, *DBLP-ACM*$_1$ and *DBLP-ACM*$_2$ dataset respectively. We can draw a conclusion that compared with these ML-based and DL-based works, GTA achieves improvement by additionally utilizing Transformer architecture and pretrained LM's prior knowledge, thus performs better in record sequence representation.
- GTA shows competitive performance compared to Transformer-based works (Baseline, DITTO and BERT-ER) and outperforms Baseline by an average of 2.8 F1 score. Due to the introduction of graph-based hybrid embedding (GHE) module, GTA encodes the dual-level matching information and multiple granularity interactions into a hybrid matching graph $\mathcal{G}_{HM}$. By conducting contrastive learning as a pretext task on $\mathcal{G}_{HM}$, structural features of

relational data can be obtained without any labeled data. Then the hybrid matching embeddings E^{HM} of attributes and tokens act as additional features to enhance Transformer's representation for relational data and improve the final EM performance.

- GTA shows robustness on dirty data. As can be seen in Table 2, GTA achieves state-of-the-art result on $DBLP\text{-}ACM_2$ dirty dataset. GTA's robust performance can be ascribed to two reasons: (1) both instance-level and hidden-level data augmentation on the hybrid matching graph $\mathcal{G}_{HM}$, this ensures that the prelearned hybrid matching embedding is generalized enough and not specialized to explicit attribute, token instances or feature dimensions, and (2) comparison and aggregation on both attribute-level and token-level, which ensures the attribute-unaligned tokens' interaction to perform comparison.

4.3 Detailed Analysis

Ablation Study. To evaluate the contribution of each component, we conduct an ablation study on GTA framework by ablating a specific component of GTA. GTA (-HM) refers to dropping hybrid matching knowledge injection. GTA (-AST) refers to dropping attribute-specific tokens. And GTA (-HM-AST) refers to dropping both of them. According to the results in Table 4, we can draw a conclusion that the additional attribute-specific tokens can accomplish finer separating to split various attributes, enabling Transformer to regard input entries as relational records. And the injected additional knowledge can significantly improve Transformer-based EM performance.

Table 4. Ablation study results compared with the full GTA framework.

	AG	BR	FZ	DA_1	DA_2
GTA	76.2	96.3	100	99.1	99.0
GTA (-HM)	74.3	87.3	98.1	98.9	98.9
	−1.9	−9.0	−1.9	−-0.2	−0.1
GTA (-AST)	75.6	95.4	99.8	98.9	99.0
	−0.6	−0.9	−0.2	−0.2	−0
GTA (-HM-AST)	74.1	86.7	98.1	98.8	98.9
	−2.1	−9.6	−1.9	−0.3	-0.1

Various LMs in GTA. We evaluate GTA using various BERT-like language models (i.e., BERT, RoBERTa, DistilBERT). As can be seen in Table 5, GTA (RoBERTa) shows the leading F1 score in our GTA framework. This can be attributed to that RoBERTa pretrains longer time on more data with bigger batch size than BERT and DistilBERT. RoBERTa also removes the next sentence prediction (NSP) objective of BERT, considering that the NSP objective may harm the downstream task performance. GTA (DistilBERT) also achieves good results, although the trainable parameter of DistilBERT (66M) is about

half of RoBERTa (125M). For the challenging dataset *Amazon-Google*, GTA (RoBERTa) outperforms GTA (DistilBERT) by 4.8 F1 score, but for datasets like *Fodors-Zagats*, *DBLP-ACM*$_1$ and *DBLP-ACM*$_2$, the margins of F1 score are 0.8, 0.4 and 0.6 separately.

Table 5. GTA's F1 score using various language models.

	AG	BR	FZ	DA_1	DA_2
GTA (BERT)	72.6	95.9	98.4	98.8	98.8
GTA (RoBERTa)	76.2	96.3	100	99.1	99.0
GTA (DistilBERT)	71.4	93.9	99.2	98.7	98.4

5 Conclusion and Outlook

In this paper, we propose a novel entity matching framework named GTA. GTA verifies the feasibility of knowledge injection for Transformer to perform EM tasks. By injecting additional hybrid matching knowledge, which is obtained via graph contrastive learning in a designed hybrid matching graph, GTA enhances Transformer for relational data representation. This enables Transformer to regard input entries as relational records to aggregate both attribute-level and token-level matching information. Compared with existing EM works, our proposed GTA framework effectively improves Transformer's representation for relational data and achieves state-of-the-art results on open datasets. Just like other related works, we hope to shed some light on this direction by conducting researches on AI4DB, and making contributions to both DB and AI communities.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (62172082, 62072084, 62072086, U1811261).

References

1. Abedjan, Z., et al.: Detecting data errors: where are we and what needs to be done? Proc. VLDB Endow. **9**(12), 993–1004 (2016)
2. Brunner, U., Stockinger, K.: Entity matching with transformer architectures-a step forward in data integration. In: International Conference on Extending Database Technology, Copenhagen, 30 March–2 April 2020. OpenProceedings (2020)
3. Cappuzzo, R., Papotti, P., Thirumuruganathan, S.: Creating embeddings of heterogeneous relational datasets for data integration tasks. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, pp. 1335–1349 (2020)
4. Chen, R., Shen, Y., Zhang, D.: GNEM: a generic one-to-set neural entity matching framework. In: Proceedings of the Web Conference 2021, pp. 1686–1694 (2021)

5. Christen, P.: Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection (2012)
6. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (ELUs). arXiv preprint arXiv:1511.07289 (2015)
7. Dalvi, N., Rastogi, V., Dasgupta, A., Das Sarma, A., Sarlós, T.: Optimal hashing schemes for entity matching. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 295–306 (2013)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186 (2019)
9. Dong, X.L., Srivastava, D.: Big data integration. In: 2013 IEEE 29th international conference on data engineering (ICDE), pp. 1245–1248. IEEE (2013)
10. Dosovitskiy, A., et al.: An image is worth 16×16 words: transformers for image recognition at scale. In: International Conference on Learning Representations (2020)
11. Dunn, H.L.: Record linkage. Am. J. Public Health Natl. Health **36**(12), 1412–1416 (1946)
12. Ebraheem, M., Thirumuruganathan, S., Joty, S., Ouzzani, M., Tang, N.: Distributed representations of tuples for entity resolution. Proc. VLDB Endow. **11**(11), 1454–1467 (2018)
13. Elmagarmid, A., Ilyas, I.F., Ouzzani, M., Quiané-Ruiz, J.A., Tang, N., Yin, S.: NADEEF/ER: generic and interactive entity resolution. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, pp. 1071–1074 (2014)
14. Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., Yin, D.: Graph neural networks for social recommendation. In: The World Wide Web Conference, pp. 417–426 (2019)
15. Fu, C., Han, X., He, J., 0001, L.S.: Hierarchical matching network for heterogeneous entity resolution. In: IJCAI, pp. 3665–3671 (2020)
16. Gokhale, C., et al.: Corleone: hands-off crowdsourcing for entity matching. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, pp. 601–612 (2014)
17. Jin, W., et al.: Graph representation learning: foundations, methods, applications and systems. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 4044–4045 (2021)
18. Marcus, A., Wu, E., Karger, D., Madden, S., Miller, R.: Human-powered sorts and joins. Proc. VLDB Endow. **5**(1) (2011)
19. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (Poster) (2015)
20. Konda, P., et al.: Magellan: toward building entity matching management systems. Proc. VLDB Endow. **9**(12), 1581–1584 (2016)
21. Li, B., Miao, Y., Wang, Y., Sun, Y., Wang, W.: Improving the efficiency and effectiveness for BERT-based entity resolution. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 13226–13233 (2021)
22. Li, B., Wang, W., Sun, Y., Zhang, L., Ali, M.A., Wang, Y.: GraphER: token-centric entity resolution with graph convolutional neural networks. In: AAAI, pp. 8172–8179 (2020)

23. Li, Y., Li, J., Suhara, Y., Doan, A., Tan, W.C.: Deep entity matching with pre-trained language models. Proc. VLDB Endow. **14**(1), 50–60 (2020)
24. Li, Y., Li, J., Suhara, Y., Wang, J., Hirota, W., Tan, W.C.: Deep entity matching: challenges and opportunities. J. Data Inf. Qual. (JDIQ) **13**(1), 1–17 (2021)
25. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
26. Liu, Y., Pan, S., Jin, M., Zhou, C., Xia, F., Yu, P.S.: Graph self-supervised learning: a survey. arXiv preprint arXiv:2103.00111 (2021)
27. Mudgal, S., et al.: Deep learning for entity matching: a design space exploration. In: Proceedings of the 2018 International Conference on Management of Data, pp. 19–34 (2018)
28. Oord, A.V.D., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
29. Peng, Y., Choi, B., Xu, J.: Graph learning for combinatorial optimization: a survey of state-of-the-art. Data Sci. Eng. **6**(2), 119–141 (2021)
30. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
31. Singh, R., et al.: Synthesizing entity matching rules by examples. Proc. VLDB Endow. **11**(2), 189–202 (2017)
32. Sun, C.C., Shen, D.R.: Mixed hierarchical networks for deep entity matching. J. Comput. Sci. Technol. **36**(4), 822–838 (2021)
33. Sun, C., Shen, D.: Entity resolution with hybrid attention-based networks. In: Jensen, C.S., et al. (eds.) DASFAA 2021. LNCS, vol. 12682, pp. 558–565. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-73197-7_37
34. Tang, N., et al.: RPT: relational pre-trained transformer is almost all you need towards democratizing data preparation. Proc. VLDB Endow. **14**(8), 1254–1261 (2021)
35. Wang, F., Liu, H.: Understanding the behaviour of contrastive loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2495–2504 (2021)
36. Wang, J., Kraska, T., Franklin, M.J., Feng, J.: CrowdER: crowdsourcing entity resolution. Proc. VLDB Endow. **5**(11) (2012)
37. Wang, J., Li, G., Yu, J.X., Feng, J.: Entity matching: how similar is similar. Proc. VLDB Endow. **4**(10), 622–633 (2011)
38. Zhang, D., Nie, Y., Wu, S., Shen, Y., Tan, K.L.: Multi-context attention for entity matching. In: Proceedings of The Web Conference 2020, pp. 2634–2640 (2020)
39. Zheng, Y., Zhang, R., Huang, M., Mao, X.: A pre-training based personalized dialogue generation model with persona-sparse data. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 9693–9700 (2020)