HETEROGENEOUS ENTITY MATCHING WITH COMPLEX ATTRIBUTE ASSOCIATIONS USING BERT AND NEURAL NETWORKS

Jiamin.Lu and Shitao. Wang

Key Laboratory of Water Big Data Technology of Ministry of Water Resources
Hohai University, Nanjing, China

ABSTRACT

Across various domains, data from different sources such as Baidu Baike and Wikipedia often manifest in distinct forms. Current entity matching methodologies predominantly focus on homogeneous data, characterized by attributes that share the same structure and concise attribute values. However, this orientation poses challenges in handling data with diverse formats. Moreover, prevailing approaches aggregate the similarity of attribute values between corresponding attributes to ascertain entity similar- ity. Yet, they often overlook the intricate interrelationships between attributes, where one attribute may have multiple associations. The simplistic approach of pairwise attribute comparison fails to harness the wealth of information encapsulated within entities.To address these challenges, we introduce a novel en- tity matching model, dubbed "Entity Matching Model for Capturing Complex Attribute Relationships (EMM-CCAR)," built upon pre-trained models. Specifically, this model transforms the matching task into a sequence matching problem to mitigate the impact of varying data formats. Moreover, by introducing attention mechanisms, it identifies complex relationships between attributes, emphasizing the degree of matching among multiple attributes rather than one-to-one correspondences. Through the integration of the EMM-CCAR model, we adeptly surmount the challenges posed by data heterogeneity and intricate attribute interdependencies. In comparison with the prevalent DER-SSM and Ditto approaches, our model achieves improvements of approximately 4% and 1% in F1 scores, respectively. This furnishes a robust solution for addressing the intricacies of attribute complexity in entity matching.

KEYWORDS

Entity Matching, Attribute Comparision, Attention, Pre-trained Model

1. INTRODUCTION

Knowledge graph update [23] is a dynamic process of maintaining and revising existing knowledge graphs to reflect the ever-changing landscape of real-world knowledge. In this context, entity matching (EM) assumes paramount importance as different data sources continuously evolve, leading to a more complex and challenging knowledge graph update.

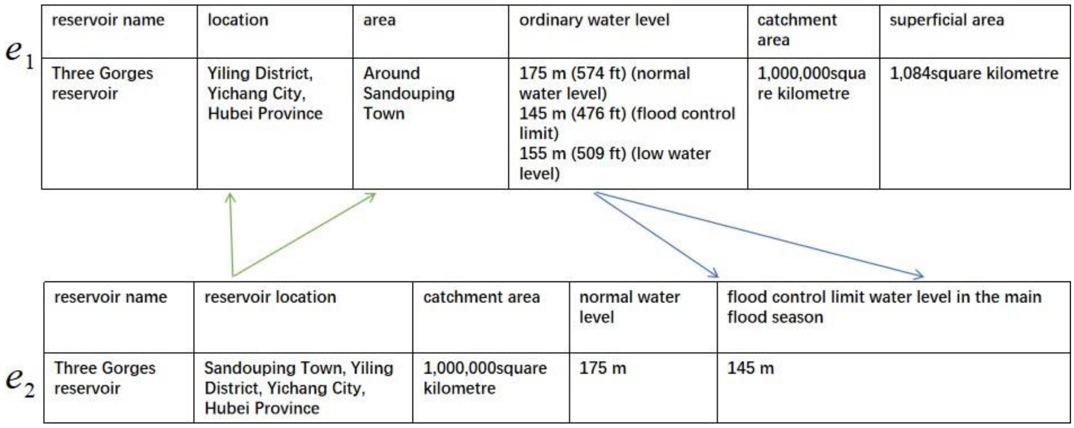
Entity Matching (EM) [7] aims to determine whether different data references point to the same real-world entity. The objective of EM is to ascertain if data belongs to the same hydraulic entity. In entity matching, data can be classified into two categories [26]: homogenous data and heterogeneous data. Homogenous data refers to data with the same schema, meaning they share identical attribute names. Based on the correctness of at- tribute values and their alignment with attribute names, homogenous data can be further categorized into clean data and dirty data. Clean

David C. Wyld et al. (Eds): MLNLP, NWCOM, DTMN, ASOFT, SIGPRO, AIFZ, CSITY, CLSB - 2023
pp. 53-65, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131605

data indicates that attribute values are correctly placed under the appropriate attributes, i.e., attribute values are aligned with corresponding attributes in the schema. Dirty data [27] implies that attribute values might be erroneously placed under the wrong attributes, i.e., attribute values are not aligned with corresponding attributes in the schema. Heterogeneous data, on the other hand, involves dissimilar attribute names and may exhibit one-to-one, one-to-many, or many-to-many correspondence relationships.

Entity matching [22] typically involves two steps: blocking [24] and matching [25]. The purpose of blocking is to reduce computational costs by partitioning records into multiple blocks, where only records within the same block are considered potential matches. Subsequently, within each block, matching is performed to identify valid pairs of matching records, which is a crucial step in the entity matching process. However, in the matching process, prevalent models often encounter attribute matching issues, particularly when dealing with heterogeneous data.

As these models [8] [10] [11] typically concentrate on homogenous data (often directly performing entity matching on structured database tables), they neglect the consideration of heterogeneous data (i.e., data scraped from web pages, where data attributes exhibit substantial variations). As depicted in Fig.1, e_1 and e_2 respectively represent heterogeneous data extracted from Wikipedia and Baidu Baike about the Three Gorges Reservoir. Their attribute names are not identical and complex correspondence relationships exist. For candidate entity pairs (e_1, e_2), conventional EM methods tend to compare tokens based on properties like “Location” and “Reservoir Location”, due to their highest token similarity. However, the attribute “Reservoir Location” encompasses information related to both “Location” and “Region”, and a simplistic token-based similarity assessment between “Reservoir Location” and “Location” neglects the context of “Area” thereby diminishing the matching accuracy.



reservoir name	location	area	ordinary water level	catchment area	superficial area
Three Gorges reservoir	Yiling District, Yichang City, Hubei Province	Around Sandouping Town	175 m (574 ft) (normal water level) 145 m (476 ft) (flood control limit) 155 m (509 ft) (low water level)	1,000,000square kilometre	1,084square kilometre

reservoir name	reservoir location	catchment area	normal water level	flood control limit water level in the main flood season
Three Gorges reservoir	Sandouping Town, Yiling District, Yichang City, Hubei Province	1,000,000square kilometre	175 m	145 m

Fig. 1. Wikipedia and Baidu Baike information about the Three Gorges Reservoir.

In this work, we propose a neural network approach based on pre-trained models to capture attribute matching information for deep entity matching (EM). To establish the correspondence between entity attributes, following the hierarchical structure Token→Attribute→Entity, we compare individual tokens within entities and across entities to obtain token similarity information. By subsequently aggregating the similarity information among tokens, we uncover complex attribute relationships in heterogeneous data. Ultimately, entity similarity is derived by evaluating the similarity of attribute values. To address matching challenges in heterogeneous data, we first learn contextual representations of tokens for a given pair of entities. Subsequently, within each entity, we leverage self-attention mechanisms to ascertain token dependencies, thus determining the significance of tokens within entities. This is followed by cross-entity token alignment using interaction attention mechanisms, yielding token similarity between entities. The aggregated token similarity is then weighted to derive attribute similarity. Concurrently, candidate entity pairs are serialized into sentence inputs for BERT model, generating sentence-level embeddings to mitigate the impact of data heterogeneity. Subsequently, within a Linear layer, heightened emphasis is placed on the matching degree of similar attributes, harnessing more attribute information while disregarding the influence of dissimilar attributes. This comprehensive approach culminates in the determination of entity matching outcomes. Our main contributions can be summarized as follows:

1) We employ BERT for contextual embeddings, enabling richer semantic and contextual information to be learned from a reduced dataset and producing more expressive token embeddings.

2) Building upon the transformation of entity matching into sequence pair classification, we introduce attribute similarity. This inclusion grants heightened focus to similar attributes, effectively harnessing entity attribute information.

3) We crawled data about Songhua River Basin and Liao River Basin from Wikipedia and Baidu Baike, resulting in a dataset encompassing 4039 reservoirs and 6576 river data entries. We constructed a water resources dataset and validated the model's effectiveness and robustness on this dataset.

2. Related Work

Entity Matching (EM), also known as entity resolution or record linkage, has been a subject of research in data integration literature for several decades [5]. To mitigate the high complexity of directly matching every pair of data, EM is typically divided into two steps: blocking and matching.

In recent years, matching techniques have garnered increased research attention [2]. Ditto leverages pre-trained models such as BERT to transform entity matching into a binary classification problem of sequence pairs. It accomplishes this by inserting data attributes and values into special COL and VAL markers, concatenating them into sequence pairs, and then inputting them into the pre-trained model. This enables the model to classify sequence pairs and thus perform entity matching tasks.

DER-SSM [1] introduces and implements soft pattern matching, flexibly associating the relationships between attributes by considering inter-word correlations. It aggregates word information during entity matching to express relationships between attributes, greatly enhancing the effectiveness of entity matching for complex and corrupted data.

JointMatcher [3] employs relevance-aware encoders and numeral-aware encoders to enhance the model's focus on similar segments and numeral segments within sequences, thereby improving the accuracy of entity matching. HHG [4] pioneers the use of graph neural networks to establish a hierarchical structure among words, attributes, and entities. By learning entity embeddings from top to bottom and capturing more contextual information, it enhances the derivation of entity embeddings.

Ditto utilizes the BERT model to transform entity matching into a binary classification problem of sequence pairs, better exploiting the contextual information of tokens. DER- SSM considers soft patterns to establish correspondences between attributes, mitigating the impact of heterogeneous data. Meanwhile, JointMatcher prioritizes the matching degree of similar segments between entities during the matching process. We have comprehensively considered these methods and, based on the foundation of using the BERT model to convert entity matching into binary classification of sequence pairs, we focus on the matching degree of similar attributes to address the issues of inadequate utilization of semantic information and matching of heterogeneous data.

3. PRELIMINARIES

This section provides a formal definition of Entity Matching (EM) and subsequently out- lines an LM-based approach to solving EM.

3.1. Entity Matching

Entity Matching, also known as entity resolution, refers to the process of identifying pairs of records from structured datasets or text corpora that correspond to the same real- world entity. Let D be a collection of records from one or multiple sources, such as rows of relational tables, XML documents, or text paragraphs. Entity Matching typically involves two steps: blocking and matching. In this paper, we focus on the matching step of entity matching. Formally, we define the entity matching problem as follows:

Input: A set M of pairs of records to be matched. For each pair $(e_1, e_2) \in$ M, e $= \{(attr_i, val_i)\}_{1\leqslant i\leqslant k}$ each entity is represented in the form of K key-value pairs, where Key and Value are respectively the attribute name and attribute value of the entity.

Output: A set of pairs of records M^*, where each entity in each pair (e_1, e_2) points to the same entity, indicating the same real-world entity.

In this definition, our input is sufficiently general to apply to both structured and textual data. Attribute names and attribute values can take any form, including even indices or other identifiers, even if their true semantics are not available, such as "$attr_1$" and "$attr_2$".

3.2. Methodology Framework

the schema of an entity represents an abstracted representation of the basic information about that entity. Schema matching is often a necessary prerequisite in the context of Entity Matching (EM), as there might be differences among attributes of different entities. Traditional EM methods typically establish one-to-one mapping relationships between attributes from different entities. However, in reality, the associations between two entity attributes can be intricate, and simple one-to-one mappings may fail to capture these complex relationships. To address EM more effectively, it becomes crucial to consider the intricate associations between attributes during the entity matching process, thereby enhancing the performance of entity matching.

For entities comprising distinct attributes, an entity itself can be seen as an instance of a schema. Given two entities, $e_1 = \left\{ < a_1^s, v_1^s > ... < a_m^s, v_m^s > \right\}$ and $e_2 = \left\{ < a_1^t, v_1^t > ... < a_n^t, v_n^t > \right\}$, $\left\{ a_1^s, a_2^s, ..., a_m^s \right\}$ and $\left\{ a_1^t, a_2^t, ..., a_n^t \right\}$ respectively denote the distinct schemas of the two entities, and each schema is composed of Tokens representing attribute values of the entities.

To achieve this goal, we construct a neural network based on BERT for entity matching. As illustrated in Fig.2, the left part involves a neural network that captures the complex associations between entity attributes. On the right side of the matching process, the final matching results are derived by considering the association matrix, which focuses on the degree of matching between different entity attributes. The matching process of the network mainly comprises the following steps: (1) Token Embedding: Converting Tokens within attribute values into vectors using BERT, while capturing contextual relation- ships between each Token. (2) Token Self-attention: Obtaining attention scores between Tokens of the same entity through self-attention mechanism. (3) Token Aggregation: Aggregating attention scores between Tokens to obtain similarity information. (4) Attribute Inter-Attention: Determining attention scores between Tokens of different entities through interactive attention. (5) Attribute Comparison: Aggregating similarity scores between Tokens within and between entities to create a similarity matrix for attributes. (6) Serialize: Serializing entities in the form of ¡Key, Value¿. (7) Sentence Embedding: Converting the serialized result into sentence vectors. (8) Linear: Focusing on the matching degree of similar attributes within sentence vectors. (9) Softmax: Normalizing the output of the Linear layer, resulting in a match (0/1) output.

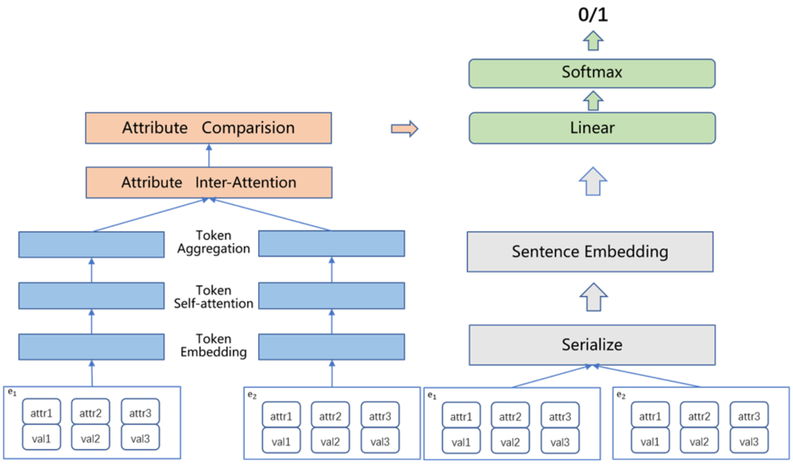


Fig. 2. Model Architecture, On the left is how to capture complex relationships between attributes, and on the right is how to incorporate these complex relationships into the matching process.

4. EM -CAR MODEL

In this section, we will mainly introduce the specific implementation of each step of the model.

4.1. Token Embedding

Each token of both attribute values and attribute names needs to be embedded into a low-dimensional vector for subsequent calculations. Since attribute values are composed of sequences of different tokens, in the embedding process, the same token may have different vector representations in different contexts. Therefore, contextual semantic information should be integrated into the vectors. BERT is a pre-trained deep bidirectional Transformer model that effectively captures the semantic information of each token in its context through unsupervised learning on large-scale corpora. This implies that the token vectors generated by BERT can better represent the meaning of each token. Therefore, we choose to use the pre-trained BERT model for token embedding.

$$H_i^s = BERT(A_i^s V_i^s) \tag{1}$$

In the process of obtaining the attribute similarity matrix, the vector representation of entities is obtained by concatenating different tokens. $H^S = [H_1^s, H_2^s, ..., H_m^s]$, $H^T = [H_1^t, H_2^t, ..., H_n^t]$.

4.2. Token Self-Attention

During the matching process, it is necessary to compare the similarity of each attribute, so we need to determine the significance of attributes within an entity. This is achieved through Token Self-attention, which computes the weights of tokens to aggregate the im- portance of attributes. Its role is to establish relationships between each token and other tokens within the sequence, capturing significant interdependencies to derive the importance of attributes. Through Self-attention, each token can be weighted and combined based on the importance of other tokens in the sequence, thereby better reflecting con- textual information and semantic dependencies. Such an attention mechanism enables the model to dynamically focus on important tokens while disregarding less significant ones. Consequently, token-level self-attention is employed to weight the tokens within an entity. For attributes, their self-attention scores are computed using trainable matrices, as shown in the equation.

$$a_i^s = soft\max((H_i^s)^T W_s H_i^s) \tag{2}$$

4.3. Token Aggregation

To better harness token information for token-level comparisons, we employ Token Aggregation to merge the representations obtained after the Self-Attention operation, creating a comprehensive representation of the entire entity. This aggregation fuses all token in- formation from the sequence into a single vector, enhancing the overall representation of the entity's information. This process involves an attention matrix, but we require a token weight vector for token aggregation. Therefore, we utilize a transformation function *m2v()* to convert it into a token weight vector W_i. *m2v()* By summing the aggregation of each row of a_i^s, a vector is derived, and subsequently, each element of the vector is normalized by dividing it by the maximum element.

$$a_i^{ts} = m2v(a_i^s) \tag{3}$$

4.4. Attribute Inter-Attention

The aforementioned operations yield attribute relationships within individual entities. However, our objective is to capture the complex relationships between attributes in heterogeneous data. Therefore, it is necessary to perform matching across different entities. Through Attribute Inter-Attention in entity matching tasks, interactive attention calculation is applied to different entity attributes. This process yields correspondences between attributes of different entities, enabling the learning of correlations among various at- tributes. As a result, a better grasp of the associations between entities is achieved. This attention mechanism aids in focusing on attributes relevant to matching while disregarding irrelevant ones. Each entity is considered as a sequence concatenated by tokens, inter-entity interaction attention is leveraged to obtain interaction representations between H_i^s and H_t^j. Here, $W_{i \to T}$ denotes the inter-entity interaction attention.

$$\beta^{i \to T} = soft\max((H_i^s)^T W_{i \to T} H^T) \tag{4}$$

$$H_i^s = \beta^{i \to T} H^T \tag{5}$$

4.5. Attribute Comparision

After obtaining the cross-entity correspondences, we need to use these correspondences to calculate the similarity between entity attributes, thus deriving the relationships between attributes. Attribute Comparison involves comparing different attributes of distinct entities during the matching process, computing their similarity or dissimilarity, and thereby gauging the level of association between different attributes. This attribute comparison mechanism aids the model in capturing essential features among attributes in entity matching tasks, further assisting the model in entity-level matching or classification. We apply element-wise absolute difference and Hadamard product to H_i^s and H_t^j , and incorporate the intermediate representation into a highway network. Subsequently, the token-level similarity $\overset{s}{C}$ from *e₁* to *e₂* is the output of the highway network.

$$C = HighwayNet([H^s - H^s],[H^s \otimes H^s]) \tag{6}$$

Lastly, the aggregation of token similarities obtained through the interaction attention mechanism of self-attention yields the similarity between entity attributes.

$$\delta_i^s = \sum_{x\in[1,\left|H_i^5\right|]} C_i^s(x)\alpha_i^{ts}(x)\overset{s}{C} \tag{7}$$

$$R_{ij}^{S\to T} = \sum_{x\in[1,\left|H_i^s\right|]} C_i^s(x)\alpha_i^{ts}(x)\overset{s}{C} \tag{8}$$

4.6. Serialize and Sentence-Embedding

We employ the methods from Ditto [2] to serialize the data and generate sentence embeddings. For each entity pair, we serialize it as follows: *serialize(e)* = *[COL]attr₁[VAL]val₁[COL]attr₂[VAL]va*, Where [COL] and [VAL] are special tokens used to indicate the start of attribute names and values, respectively. For example, the first entry in the second table is serialized as: For each candidate entity pair, *serialize(e₁, e₂)* = *[CLS]serialize(e₁)[SEP]serialize(e₂)[CLS]*, where [SEP] is a special token that separates the two sequences, and [CLS] is a special token required by BERT to encode the sequence pair into a 768-dimensional vector.

4.7. Linear and Softmax

In entity matching tasks, a linear layer is employed to perform a linear transformation on the vectors that have undergone feature extraction and encoding. The formula representing the linear layer for input feature vector X is as follows:

$$L(X,W,b) = X\square W + b \tag{9}$$

Here, W represents the weight matrix to be learned, and b is the bias vector. In this context, the vector X is obtained through previous serialization and sentence embedding, resulting in the embedded vector E for entity pairs. The matrix W corresponds to the obtained entity attribute similarity matrix.

Further applying a softmax function yields the output vector. This vector represents a probability distribution, where each element signifies the probability for the respective category. In this context, the output of 0 signifies a non-match, while 1 signifies a match.

5. EXPERIMENT

In this section, we utilized a dataset to evaluate our EM-CAR model.

5.1. Experiment Dataset

When evaluating our approach, we utilized various types of datasets, including: Two isomorphic public datasets [9] (simple 1:1 attribute associations). Two heterogeneous public datasets (complex associations of 1:m and m:n). A hydraulic heterogeneous dataset (complex associations of 1:m and m:n). The information for the public datasets is presented in the following table. Homogeneous Data: The patterns of homogeneous data involve simple.

associations (1:1). iTunes-Amazon (iA) and DBLP-Schoral1 (DS1) correspond respectively to iTunes-Amazon1 and DBLP-choral1 [5].

Heterogeneous Data: The complex data SM involves complex associations (1:m or m:n). We implemented a variant of the Synthetic Data Generator UIS to generate the UIS1-UIS2 (UU) dataset [10]. The initial five attributes are name, address, city, state, and zip code. Address and city are combined into a new attribute in UIS1 records, while city and state are integrated into a new attribute in UIS2 records. Therefore, the attribute numbering for UU is 4–4. Walmart-Amazon1 (WA1) is a variant of Walmart-Amazon (5-5) [12]. Brand and model are merged into a new attribute in Walmart records, while category and model are integrated into a new attribute in Amazon records. Hence, the attribute numbering for WA1 is 4–4.

Table 1. The “Size” column indicates the size of the “Size” table, “#POS.” represents the number of positive matches, and “#ATTR.” represents the attribute number. The attribute association “m:n” between two patterns is entirely different from the attribute numbering “c-d”. The “m:n” attribute association signifies the presence of at least one complex 1:m or m:n attribute association between two patterns. The attribute numbering “c-d” only denotes that the first pattern has “c” attributes, while the second pattern has “d” attributes.

Type	Dataset	Domain	Size	#POS.	#ATTR.
Same pattern	iTunes-Amazon	Music	539	132	8-8
Same pattern	DBLP-Scholar	Citation	28707	5347	4-4
Different pattern	UIS1-UIS2	Person	12853	2736	4-4
Different pattern	Walmart-Amazon	Electronics	10242	962	4-4

Next, we introduce the composition of the hydraulic dataset, which we crawled separately from Wikipedia and Baidu Baike. We collected data about the Songhua River Basin and the Liao River Basin, including 4039 reservoirs and 6576 river records. Reservoirs include attributes such as reservoir name, location, area, region, normal water level, watershed area, normal storage level, etc. Rivers include attributes such as river name, region, river grade, river length, basin area, etc. As shown in Figure 1, there exist com- plex correspondences among these attributes. During data processing, we labeled data belonging to the same entity as matching and labeled data pointing to different entities as non-matching. Since this data was crawled from web pages based on names, most of the non-matching entities are entities with the same name, and the proportion of same-name entities in the data is not high. Therefore, there are not enough negative examples in the data. We created some negative examples by replacing attribute values with synonyms. Finally, our dataset has a size of 21230, including 5000 positive instances, with 2000 positive instances for reservoirs and 3000 for rivers.

5.2. Implementation and Setup

We implemented our model using PyTorch and the Transformers library. In all experiments, we used the base uncased variants of each model. We further applied mixed- precision training (fp16) optimization to accelerate both training and inference speed. For all experiments, we fixed the maximum sequence length to 256, set the learning rate to 3e-5, and employed a linear decay learning rate schedule. The training process runs for a fixed number of epochs (10, 15, or 40 depending on the dataset size) and returns the checkpoint with the highest F1 score on the validation set.

Comparison Methods. We compare EM-CAR with state-of-the-art EM solutions such as Ditto, the attribute correspondence-aware method DEM-SSR, and the classical method DeepMatcher. Here’s a summary of these methods. We report the average F1 score over 6 repeated runs in all settings.

DeepMatcher: DeepMatcher is a state-of-the-art classical method, customizes an RNN architecture to aggregate attribute values and then compare/align the aggregated representations of attributes.

Ditto: Ditto is a state-of-the-art matching solution that employs all three optimizations, Domain Knowledge (DK), TF-IDF summarization (SU), and Data Augmentation (DA).

DER-SSM: In comparison to Ditto, DER-SSM defines and implements soft pattern matching, obtaining context relations between tokens through BiGRU. It considers soft pattern matching by aggregating token similarity during entity matching based on the context relationships between tokens.

5.3. Experiment Result

The F1 score is used to measure the precision of entity matching (EM) and is the harmonic mean of precision (P) and recall (R). Precision (P) represents the score of correct matching predictions, while recall (R) represents the score of true matches predicted as matches.

Typically, in EM, there are two phases: blocking and matching [2]. Our focus is on the matching phase in entity matching (EM), assuming that blocking has already been performed. We follow the same blocking setup [2], where blocking is applied to generate a candidate set for the dataset. All pairs in the candidate set are labeled. The dataset is then divided into a 3:1:1 ratio for training, validation, and testing.

Table 2. Average F1 Scores of Different Methods.

Type	Dateset	DeepMatcher	DER-SSM	Ditto	EM-CAR

Same pattern	iTunes-Amazon	82.3	85.7	blue89.6	89.5
Same pattern	DBLP-Scholar	85.4	89.2	90.1	blue90.9
Different pattern	UIS1-UIS2	76.2	80.4	85.2	blue86.3
Different pattern	Walmart-Amazon	77.1	81.2	84.4	blue85.3
Water data	SongLiao	74.3	78.2	80.1	blue81.2

To further demonstrate the performance of EM-CAR, we conducted a case study com- paring it with Ditto. First, it should be noted that Ditto directly utilizes context-based embeddings obtained from pre-trained language models (LM) for classification, making it not entirely suited for entity matching (EM) tasks. Specifically, Ditto's embeddings might not be fully optimized for the specific task of entity matching, as they are derived from a broader range of language modeling objectives. This could potentially limit Ditto's ability to capture the nuanced and complex relationships between attributes required for accurate entity matching.

In contrast, our model aims to address this issue by placing greater emphasis on at- tributes with higher similarity. As depicted in the figure, our model takes into account the similarity of attributes, particularly those that are more closely related, This approach allows our model to better capture the nuanced relationships between attributes and im- prove the overall matching accuracy.

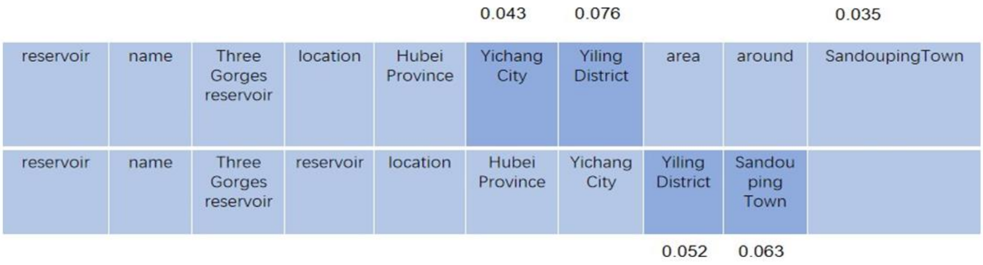


Fig. 3. Attention scores of Ditto.

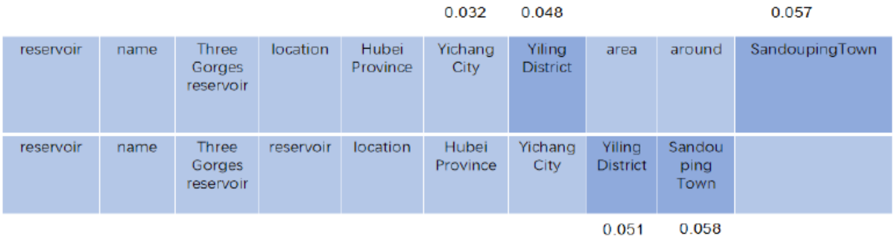


Fig. 4. Attention scores of EM-CAR.

As shown in the Fig.3, when performing matching, Ditto's utilization of pre-trained models can lead to erroneous judgments. This is attributed to the fact that, while determining whether these two entities match, the top two attention scores are placed on the tokens "YichangCity" and "YilingDistrict" while the score for "SandoupingTown" is not as high. Consequently, more attention is directed towards the correspondence between "YichangCity" and "YilingDistrict" as well as "YilingDistrict" and "SandoupingTown".

In contrast, we aim to prioritize the matching probability between "YilingDistrict" and "SandoupingTown" As illustrated in Fig.4, our model focuses more on the matching degree of similar attributes, denoted by (e_1, e_2). This enables our model to appropriately emphasize the similarity between attributes and achieve accurate results.

6. Conclusion

In this paper, We propose EM-CAR, a method that leverages attribute similarity within the context of pre-trained models to address complex entity correspondence. In our approach, we compare the classical DeepMatcher, DER-SSM (which considers soft patterns, i.e., complex attribute correspondences), and Ditto, which employs pre-trained models. We evaluate these methods, including our own, on three types of datasets: homogeneous, heterogeneous, and hydraulic data (heterogeneous). Across all datasets, DeepMatcher achieves the lowest F1 score due to its reliance on a simple CNN network, which struggles to capture semantic information effectively.

For the two homogeneous public datasets, DER-SSM and Ditto exhibit comparable accuracy, as shown in the figure. Homogeneous datasets feature straightforward 1:1 relationships, thus methodological differences have less pronounced impacts. The primary distinction lies in whether a pre-trained model is utilized.

Concerning the two heterogeneous public datasets, DER-SSM initially shows promise, but its use of BIGRU limits semantic context to a local n-character window, resulting in slightly lower accuracy compared to Ditto. In contrast, our model takes into account complex attribute correspondences, placing greater emphasis on matching similar attributes, thereby enhancing accuracy to a certain extent.

On the hydraulic dataset, limited training data affects all three models' performance, resulting in reduced accuracy. However, our model still achieves the highest F1 score. This suggests that prioritizing the matching of similar attributes during the matching process has a positive impact on improving matching accuracy.

In summary, our EM-CAR approach effectively enhances entity matching accuracy by focusing on the similarity between attributes, especially for complex correspondences, as demonstrated across various datasets in comparison to other methods such as DeepMatcher, DER-SSM, and Ditto.

References

[1] Sun, C., & Shen, D, (2022) "Towards deep entity resolution via soft schema matching", *Neurocomputing,* Vol. 471, pp107-117.

[2] Li, Y., Li, J., Suhara, Y., Doan, A., & Tan, W. C, (2020) "Deep entity matching with pre-trained language models", *arXiv preprint arXiv,* Vol. 2004, pp00584.

[3] Ye, C., Jiang, S., Zhang, H., Wu, Y., Shi, J., Wang, H., & Dai, G, (2022) "JointMatcher: Numerically-aware entity matching using pre-trained language models with attention concentration", *Knowledge-Based Systems,* Vol. 251, pp109033.

[4] Fu, C., Han, X., He, J., & Sun, L, (2021) "Hierarchical matching network for heterogeneous entity resolution. *In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp3665-3671.

[5] Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., ... & Raghavendra, V, (2018) "Deep learning for entity matching: A design space exploration", *In Proceedings of the 2018 International Conference on Management of Data,* pp19-34.

[6] Paganelli, M., Del Buono, F., Baraldi, A., & Guerra, F, (2022) "Analyzing how BERT performs entity matching", *Proceedings of the VLDB Endowment,* Vol. 15 No, 8, pp1726-1738.

[7] Li, Y., Li, J., Suhara, Y., Wang, J., Hirota, W., & Tan, W. C, (2021) “Deep entity matching: Challenges and opportunities”, *Journal of Data and Information Quality*, Vol. 13 No. 1, pp1-17.

[8] Peeters, R., & Bizer, C, (2021) “Dual-objective fine-tuning of BERT for entity matching”, *Proceedings of the VLDB Endowment*, Vol. 14, pp1913-1921.

[9] Konda, P., Das, S., Doan, A., Ardalan, A., Ballard, J. R., Li, H., ... & Raghavendra, V, (2016) “Magellan: toward building entity matching management systems over data science stacks”, *Proceedings of the VLDB Endowment*, Vol. 9 No. 13, pp1581-1584.

[10] Thirumuruganathan, S., Parambath, S. A. P., Ouzzani, M., Tang, N., & Joty, S, (2018) “Reuse and adaptation for entity resolution through transfer learning”, *arXiv preprint arXiv*, Vol. 1809, pp11084.

[11] Teong, K. S., Soon, L. K., & Su, T. T, (2020) “Schema-agnostic entity matching using pre-trained language models”, *In Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp2241-2244.

[12] Jurek, A., Hong, J., Chi, Y., & Liu, W, (2017) “A novel ensemble learning approach to unsupervised record linkage”, *Information Systems*, Vol. 71, pp40-54.

[13] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R, (2019) “Albert: A lite bert for self-supervised learning of language representations”, *arXiv preprint arXiv*, Vol. 1909, pp11942.

[14] Steorts, R. C., Ventura, S. L., Sadinle, M., & Fienberg, S. E, (2014) “A comparison of blocking methods for record linkage”, *In Privacy in Statistical Databases: UNESCO Chair in Data Privacy, International Conference*, pp253-268.

[15] Reyes-Galaviz, O. F., Pedrycz, W., He, Z., & Pizzi, N. J, (2017) “A supervised gradient-based learning algorithm for optimized entity resolution”, *Data & Knowledge Engineering*, Vol. 112, pp106-129.

[16] Papadakis, G., Skoutas, D., Thanos, E., & Palpanas, T, (2020) “Blocking and filtering techniques for entity resolution: A survey”, *ACM Computing Surveys (CSUR)*, Vol. 53 No. 2, pp1-42.

[17] Christen, P, (2011) “A survey of indexing techniques for scalable record linkage and deduplication”, *IEEE transactions on knowledge and data engineering*, Vol. 24 No. 9, pp1537-1555.

[18] Gazzarri, L., & Herschel, M, (2021) “End-to-end task based parallelization for entity resolution on dynamic data”, *In 2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp1248-1259).

[19] Gruenheid, A., Dong, X. L., & Srivastava, D, (2014) “Incremental record linkage”, *Proceedings of the VLDB Endowment*, Vol. 7 No. 9, pp697-708.

[20] Bhattacharya, I., & Getoor, L, (2004) “Iterative record linkage for cleaning and integration”, *In Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pp11-18.

[21] Bhattacharya, I., Getoor, L., & Licamele, L, (2006) “Query-time entity resolution”, *In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp529-534.

[22] Christophides, V., Efthymiou, V., Palpanas, T., Papadakis, G., & Stefanidis, K, (2020) “An overview of end-to-end entity resolution for big data”, *ACM Computing Surveys (CSUR)*, Vol. 53 No. 6, pp1-42.

[23] Hogan, A., Blomqvist, E., Cochez, M., d’Amato, C., Melo, G. D., Gutierrez, C., ... & Zimmermann, A, (2021) “Knowledge graphs”, *ACM Computing Surveys (Csur)*, Vol. 54 No. 4, pp.1-37.

[24] Li, B. H., Liu, Y., Zhang, A. M., Wang, W. H., & Wan, S, (2020) “A survey on blocking technology of entity resolution”, *Journal of Computer Science and Technology*, Vol. 35, pp.769-793.

[25] Christophides, V., Efthymiou, V., Palpanas, T., Papadakis, G., & Stefanidis, K, (2019) “End-to-end entity resolution for big data: A survey”, *arXiv preprint arXiv*, Vol. 1905, pp06397.

[26] Agarwal, O., & Nenkova, A, (2023) “Named Entity Recognition in a Very Homogenous Domain”, *In Findings of the Association for Computational Linguistics: EACL*, pp1805-1810.

[27] Brunner, U., & Stockinger, K, (2020) “Entity matching with transformer architectures-a step forward in data integration”, *In 23rd International Conference on Extending Database Technology, Copenhagen*, pp463-473.

AUTHORS

Jiamin. Lu Assistant Professor at Information Department in Hohai University, China. He received his Ph.D degree in Information Science from FernUniversität in Hagen, Germany, 2014. His research interests include parallel processing on MOD (Moving Object Database), data management in Knowledge Graph construction. At present, he is mainly working on the conjunction of big data technologies and smart water applications.

Shitao. Wang He is pursuing a master's degree in Computer Science at Hohai University. His research interests include knowledge graph, entity matching, and natural language processing.