End-to-End Multi-Perspective Matching for Entity Resolution

Cheng Fu[1,3*], **Xianpei Han**[1,2*], **Le Sun**[1,2], **Bo Chen**[1], **Wei Zhang**[4], **Suhui Wu**[4] and **Hao Kong**[4]
[1]Chinese Information Processing Laboratory, Institute of Software, Chinese Academy of Sciences
[2]State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences
[4]Alibaba Group, China
{fucheng, xianpei, sunle, chenbo}@iscas.ac.cn, {lantu.zw, linnai.wsh, konghao.kh}@alibaba-inc.com

Abstract

Entity resolution (ER) aims to identify data records referring to the same real-world entity. Due to the heterogeneity of entity attributes and the diversity of similarity measures, one main challenge of ER is how to select appropriate similarity measures for different attributes. Previous ER methods usually employ heuristic similarity selection algorithms, which are highly specialized to specific ER problems and are hard to be generalized to other situations. Furthermore, previous studies usually perform similarity learning and similarity selection independently, which often result in error propagation and are hard to be optimized globally. To resolve the above problems, this paper proposes an end-to-end multi-perspective entity matching model, which can adaptively select optimal similarity measures for heterogenous attributes by jointly learning and selecting similarity measures in an end-to-end way. Experiments on two real-world datasets show that our method significantly outperforms previous ER methods.

1 Introduction

Entity resolution (ER) aims to identify entity records referring to the same real-world entity from different data sources, which plays an important role in data cleaning [Chaudhuri *et al.* 2007], data integration [Sehgal *et al.*, 2006] and knowledge graph integration [Kong *et al.*, 2016]. For example, in Figure 1 the two product records correspondingly from *Walmart* and *Amazon* refer to the same product, and knowing this fact can help in product search, product recommendation, etc. Entity resolution is also known as entity matching [Wang *et al.*, 2011], duplicate record detection [Elmagarmid *et al.*, 2007], and record matching/linkage [Christen, 2012].

One of the main characteristics of ER is that entity records are structural, where each record is composed of one or more <*attribute*, *value*> pairs. Entity attributes are often heterogenous, i.e., they are of different data types. For example, the product records in Figure 1 are composed of five attributes, among which *title* is text, *price* is number, and *category*, *brand* and *model* are selected from fixed string sets.

Title	Category	Brand	Model	Price
Microsoft comfort optical mouse silver blue	Mice	Microsoft	d1t011	19.95

Title	Category	Brand	Model	Price
Comfort opt mse3000 silver blue	Computers	Microsoft	d1t-011	17.99

Figure 1: Two entity records referring to the same mouse, where the above is from *Walmart* and the below is from *Amazon*.

Given two entity records, typical ER approaches first compare values between aligned attributes using specific similarity measures, then aggregate comparison results of all attributes to make final ER decision [Benjelloun *et al.*, 2009; Mudgal et al., 2018]. For example, the two product records in Figure 1 will be identified as referring to the same product because their attribute values are all similar. Due to the heterogeneity of attributes, many similarity measures have been proposed for attribute comparison, including character-based similarities for string attributes [Elmagarmid *et al.*, 2007], semantic similarities for text attributes [Ebraheem *et al.*, 2018; Mudgal *et al.*, 2018], and numeric similarities for number attributes [Koudas *et al.*, 2004], etc.

Given such a diversity of similarity measures, one main challenge of effective ER is how to select appropriate measures for different attributes in different ER problems. For example, in Figure 1 an effective ER system should select numeric measures for product prices, semantic measures for product *titles* and *categories*, and character-based measures for product *brands* and *models*. To address the above challenge, many methods have been proposed. For instance, Kong *et al.* [2016] manually selected similarity measures for different attributes in their model. Chaudhuri *et al.* [2007] proposed a recursive divide and conquer strategy to select similarity measures and thresholds for ER rules. Wang *et al.* [2011] designed three redundancy-based heuristic algorithms to select similarity measures and thresholds for ER rules. One main drawback of these methods is that they all select similarity measures heuristically, which are usually specialized to specific ER problems and are hard to be generalized to other situations. Furthermore, previous studies

[*] Corresponding author.

perform similarity learning and similarity selection independently, which often result in error propagation and are hard to be optimized globally.

In this paper, we propose a neural multi-perspective matching (MPM) model for entity resolution, which can learn to select optimal similarity measures for different attributes in an end-to-end way. Specifically, we design a "*compare-select-aggregate*" neural framework, which first compares aligned attribute values in multiple perspectives using different similarity measures, then adaptively selects the optimal similarity measure for each attribute by designing a gate mechanism, finally aggregates the comparison results of the selected similarity measures from all attributes to make ER decision. In our neural network framework, the above compare, select, and aggregate functions are correspondingly modeled as a comparison layer, a selection layer and an aggregation layer, as shown in Figure 2. We can see that, by modeling the similarity measure selection via a gate mechanism, our approach can adaptively learn to select optimal similarity measures for different attributes in different ER problems, therefore no manual selection and no heuristic rules are needed. Furthermore, all comparison, selection and aggregation components in our neural network framework are learnable and flexible, therefore our method can be globally optimized in an end-to-end way, which prevents the error propagation problem and the local optimal problem. To our best knowledge, this is the first ER work which can jointly learn and select similarity measures in an end-to-end way.

We conduct experiments on two real-world datasets—*Walmart-Amazon* [Konda *et al*., 2016] and *Amazon-Google* [Köpcke *et al*., 2010]. Experiments show that, by learning an end-to-end "*compare-select-aggregate"* model and adaptively selecting similarity measures for different attributes, our method significantly outperforms previous methods and recently proposed neural network models.

2 Entity Resolution via End-to-End Multi-Perspective Matching

This section describes how to resolve entities via our end-to-end multi-perspective matching model. We first introduce our framework in Section 2.1, then present the similarity measures used for multi-perspective comparison in Section 2.2, and then describe the gate mechanism for adaptively similarity measure selection in Section 2.3, finally we describe the aggregation layer of our model in Section 2.4.

Formally, given two entity records $e = \{< A_1, a_1>, \dots < A_m, a_m >\}$and $e' = \{< A_1, a'_1 >, \dots < A_m, a'_m >\}$ correspondingly from two data sources E and E' with aligned attributes $\{A_1, A_2, \dots, A_m\}$, our entity matching model aims to predict the probability that e and e' refer to the same entity $P(y = 1|e, e')$. Because the size of record collections maybe very large (e.g., Amazon contains millions of products), and attributes from different sources maybe unaligned, a whole ER system will usually perform an additional blocking step to find candidate pairs [Papadakis *et al*., 2016] and a schema matching step to align attributes [Bilke and Naumann, 2005]. Like previous entity matching studies [Benjelloun *et al*., 2009; Mudgal et al., 2018], this paper focuses on entity matching and will not describe blocking and schema matching.

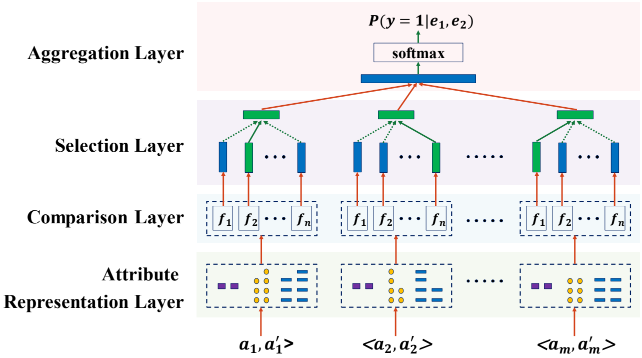

Figure 2: Framework of our end-to-end multi-perspective matching (MPM) model for entity resolution.

2.1 The Entity Matching Framework

Given two entity records, we design a "*compare-select-aggregate*" neural network framework to resolve them. The framework is shown in Figure 2. Specifically, we first compare attribute values in different perspectives using a set of similarity measures (comparison layer), then employ a gate mechanism to select the optimal similarity measures for different attributes (selection layer), finally aggregate all selected comparison results for final ER decisions (aggregation layer). In following we describe our framework layer by layer.

Attribute Representation Layer. Because attributes are heterogenous, we use three representations for each attribute: (1) its numeric value *num* if it is a number; (2) its string value *s*, i.e., as a character sequence; (3) its distributed word vector sequence $\{x_1, x_2, \dots x_k\}$. For example, <*price*, 59.99> can be represented as (59.99, "59.99", {[0.1, 0.12, …, 0.05]}), and <*title*, adobe reader> can be presented as (NAN, "adobe reader", {[0.13, 0.02, …, 0.21], [0.04, 0.11, …, 0.07]}). We use the above three representations because we want to retain as much as information for future comparison: numeric representations are useful for numeric attributes such as *price*, string representations are useful for string attributes such as *brand* and *model*; distributed word vector sequences are useful for capturing semantic similarities between textual attributes, such as *title* and *comment*.

Comparison Layer. This layer compares attribute values in different perspectives using a set of learnable similarity measures. For example, we can compute the similarity between two *prices* using their numeric difference, and compute the similarity between two *titles* using deep learning-based similarity measures such as RNN-based measures employed by Mudgal *et al.* [2018]. For each attribute value pair $< a, a' >$, our system will compare them in multiple perspectives and will output a set of comparison results $[\boldsymbol{r}_1, \boldsymbol{r}_2, \dots, \boldsymbol{r}_n]$ of n similarity measures. The detail of the similarity measures will be described in Section 2.2.

Selection Layer. In this layer, our model adaptively selects the most appropriate similarity for each attribute via a gate mechanism. Its detail will be described in Section 2.3.

Aggregation Layer. In this layer, we first concatenate the selected similarities of all attributes into a comparison vector, and make final decisions by aggregating all similarities using a neural network layer. The output of this layer is the matching probability $P(y = 1|e, e')$, where y = 1 indicates that e and e' refer to the same entity. The detail of this layer will be described in Section 2.4.

Model Learning. Given a training set *D* which contains a set of training instances (e_i, e_i', y_i), where e_i and e_i' are a pair of entity records and $y_i \in \{0,1\}$ is the golden label, we train our model by minimizing the cross entropy loss:

$$loss = -\frac{1}{|D|}\sum_{i=1}^{|D|}[y_i \log p + (1 - y_i)\log(1 - p)]$$

where |*D*| is the number of training examples, and p is the probability of y_i outputted by our model.

We can see that, the proposed framework models attribute comparison, similarity selection and comparison result aggregation as learnable neural layers in a single neural network, correspondingly the comparison layer, the selection layer and the aggregation layer. The main advantage of our framework is that all components can be learned end-to-end, making it can be easily globally optimized. Furthermore, our framework is very flexible: the similarity measure set can be easily extended using new measures; and the aggregation layer can be replaced by any advanced neural network which can output a classification probability.

2.2 Multi-Perspective Attribute Comparison

In this section, we describe the similarity measures used for multi-perspective attribute comparison. Intuitively, the same two values can be similar or dissimilar in different ER problems. For example, from the string perspective, “19.9” and “1.99” are similar, but they are dissimilar from the numeric perspective. Furthermore, as many neural network-based similarity measures have been employed in ER, it is also critical to select the appreciate neural network architectures which can best fit the entity resolution task at hand.

To this end, we found that entity attributes can be roughly categorized into three data types: numeric, string and text. A numeric attribute value is a number which is often used in *price*, *weight*, etc.; a string attribute value is a character sequence which is often used in *brand*, *model,* etc., whose values are selected and can only be selected from a fixed set; and a textual attribute value is often a description such as *title*, *comment*, etc.. Based on the above observations, we employ 8 similarity measures which are described as follows and summarized in Table 1 (Notice that our framework is flexible, it is easy to be extended using new similarity measures).

Data Type	Similarity Measure
Numeric	*rel_diff*: $2\|x - y\|/(\|x\| + \|y\|)$, where x, y are numbers. *abs_norm*: $\|\|x - y\|\|$, where x, y are numbers.
String	*exact_sim*: 1 if two strings are identical, 0 otherwise. *lev_sim*: edit distance proposed in [Levenshtein, 1966]. *jaro_sim*: distance proposed in [Jaro, 1980]. *sw_sim*: distance proposed in [Smith and Waterman, 1981].
Text	*rnn_sim*: a deep learning-based similarity measure used for the model RNN in [Mudgal *et al*., 2018]. *hybrid_sim*: a deep learning-based similarity measure used for the model Hybrid in [Mudgal *et al*., 2018].

Table 1: Similarity measures used in our model.

Numeric measures. We use two measures named *rel_diff* and *abs_norm* to compare numeric attribute values, where *rel_diff* compares the relative difference between two numbers, *abs_norm* measures the absolute norm similarity between two numbers.

String measures. We use four commonly used character-based measures in our model: exact match (*exact_sim*), Levenshtein distance (*lev_sim*) which is an edit distance-based measure, Jaro distance (*jaro_sim*) which is a string measure that was mainly used for comparison of last and first names, Smith-Waterman distance (*sw_sim*) which is an extension of edit distance but with better local string alignment.

Deep learning (DL) based textual measures. DL-based similarities are popular in recent years due to their strong abilities in learning informative representations and capturing semantic similarity. In this paper, we adopt two DL-based similarity measures proposed in [Mudgal *et al.*, 2018] to measure textual attributes. The first one uses a bidirectional RNN for attribute representation learning and an element-wise absolute difference comparison operation to form a comparison for each attribute. It is applied to their model named RNN, we call the measure *“rnn_sim”* in this paper. The second one uses a bidirectional RNN with decomposable attention to implement attribute summarization and a vector concatenation augmented with element-wise absolute difference during attribute comparison to form a comparison for each attribute. It is applied to their model named Hybrid, we call the measure *“hybrid_sim”* in this paper. Empirical results in [Mudgal et *al*., 2018] show that, DL-based models perform better than traditional methods on textual EM tasks (i.e., instances having a few attributes all of which are textual blobs).

2.3 Adaptive Measure Selection via Gate Mechanism

It is obvious that the optimal similarity measures for different attributes in different ER problems are context-sensitive. For example, to resolve the two product records in Figure 1, an effective ER system should select numeric measures to compare their *prices*, and select DL-based measures to model similarity between their *titles*. Furthermore, even we know DL-based measures are suitable for *title* attribute, an ER system still needs to select the best neural network architecture, e.g., CNN-based or RNN-based. Due to the diversity of attribute measures for ER problems, manual selection and heuristic rules are hard to be generalized to measure selection in different situations.

To solve this problem, we design a gate mechanism, which can adaptively learn to select optimal similarity measures for different attributes in different ER problems. The main motivation is that a similarity can be evaluated and selected based on its influence on final ER decisions, i.e., a similarity measure should be selected if it can achieve better performance than other similarity measures in an ER problem.

Specifically, for each attribute A, let the result outputted by the n similarity measures in the comparison layer is $\boldsymbol{r} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \dots, \boldsymbol{r}_n]$, our gate mechanism will use a mask vector $\boldsymbol{g} = [g_1, g_2, \dots, g_n]$ for similarity measure selection, where $g_i = 1$ if i-th similarity measure is selected and otherwise 0. To learn $\boldsymbol{g}$ for attribute A, we first represent A using a vector $\boldsymbol{v} \in \mathbb{R}^d$ which is randomly initialized and will be learned during training, then we estimate the soft selection vector $\boldsymbol{s}$ using:

$$\boldsymbol{s} = softmax(\delta(\,\boldsymbol{vW} + \boldsymbol{b})\,) \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{d\times n}$ and $\mathbf{b} \in \mathbb{R}^n$ are parameters to be learned, $\boldsymbol{s} \in \mathbb{R}^n$, and $s_i = \boldsymbol{s}[i]$ is the probability of the i-th measure to be selected. Using the soft selection vector $\boldsymbol{s}$, we get the final hard selection vector $\boldsymbol{g}$ as:

$$\boldsymbol{g} = h(\boldsymbol{s}) \tag{2}$$

where h is an element-wise function which assigns 1 to g_i if $s_i == \max(\boldsymbol{s})$, otherwise 0. Using the learned hard selection vector $\boldsymbol{g}$, the selection layer will select the comparison result $\boldsymbol{c}$ of attribute A as:

$$\boldsymbol{c} = \boldsymbol{r}[k] \tag{3}$$

where k is the index of the non-zero element in $\boldsymbol{g}$.

The hard selection vector $\boldsymbol{g}$ acts as a gate to control which comparison result in r will be selected for final ER decision. Using the proposed gate mechanism, we turn similarity selection into a learnable component using training data, rather than relies on manual selection or heuristic rules. In this way, the learned selection models will be adaptive for different ER problems and for different attributes. This makes our model can be easily generalized to different situations.

2.4 Aggregation Layer

An entity record usually has multiple attributes, therefore we need to aggregate comparison evidences from all attributes to make final ER decisions.

Because some similarity measures (*rnn_sim, hybrid_sim*) employed in this paper output a comparison vector, rather than a simple similarity value, we first project all similarity values to a d-dimension comparison vector $\boldsymbol{c}'$ which has the same dimension with comparison vectors:

$$\boldsymbol{c}' = \begin{cases} \boldsymbol{c}, & \text{if } \boldsymbol{c} \text{ is a vector} \\ [id,\, c]\ \boldsymbol{T}, & \text{if } \boldsymbol{c} \text{ is a scalar} \end{cases} \tag{4}$$

where $\boldsymbol{c}$ is the comparison result of attribute A outputted by the selection layer, and id is the index of the selected result, $T \in \mathbb{R}^{2\times d}$ are parameters to be learned during model training.

Then the comparison results of all m attributes are concatenated:

$$\boldsymbol{C} = [\boldsymbol{c}_1'; \boldsymbol{c}_2'; \dots; \boldsymbol{c}_m'] \tag{5}$$

Finally, $\boldsymbol{C}$ is feed into a two layer fully-connected ReLU HighwayNet [Srivastava *et al.*, 2015], which will output the matching probability $P(y = 1|e, e')$.

3 Experiments

In this section, we evaluate our method and compare it with previous methods.

3.1 Experimental Settings

Datasets. We conduct experiments on two real-world datasets: *Walmart-Amazon* and *Amazon-Google*. *Walmart-Amazon* contains product resolution data between Walmart and Amazon [Konda et al., 2016]. Amazon-Google contains product resolution data between Amazon and Google [Köpcke *et al.*, 2010]. This paper focuses on entity matching, therefore we use the after-blocking versions of the above two datasets provided by [Mudgal *et al.*, 2018]. Table 2 presents the statistics about the two datasets. Following previous studies, we evaluate all systems using precision (P), recall (R), and F1 score, and F1 is used as primary measure.

Baselines. We compare our method with three baselines:

- Magellan [Konda *et al.*, 2016]: A state-of-the-art non-deep learning ER baseline. Magellan uses record attribute values to automatically generate a large set of features, on basis of which various classifiers could be trained, such as decision tree, random forest, Naive Bayes, SVM and logistic regression, etc. One main difference between Magellan and our model is that it feeds all similarity features into a final classifier while our model selects the optimal one for each attribute. Besides, Magellan employs only traditional features while our model can combine traditional and deep learning-based similarity measures.
- RNN [Mudgal *et al.*, 2018]: A deep learning-based model which represents all attribute values using Bi-GRU, then compares all attributes using an element-wise absolute difference comparison operation, finally uses a multilayer NN to aggregate all comparison results for final prediction. The main difference between RNN and our model

Dataset	Domain	Size	Positive	Attribute
Walmart-Amazon	electronics	10,242	962	5
Amazon-Google	software	11,460	1,167	3

Table 2: Statistics of Walmart-Amazon and Amazon-Google datasets used in our experiments.

is that it compares all attributes using the same similarity measure – *rnn_sim* described in Section 2.2.

- Hybrid [Mudgal *et al.*, 2018]: A deep learning-based model which uses a Bi-GRU with decomposable attention to learn representations of attribute values, then compares all attributes using a two-layer HighwayNet followed by a weighted average operation, finally uses a multi-layer NN to aggregate all comparison results. Similar to RNN, the main difference between Hybrid and our model is that it uses the same similarity measure to compare all attributes, i.e., *hybrid_sim* described in section 2.2.

System Settings. For our system, we use the pretrained FastText 300-dimensional word embedding [Bojanowski *et al.*, 2016] for the two DL based similarity measures: *rnn_sim* and *hybrid_sim*. Hidden size of each GRU layer is set 256. For model learning, we use the same 60%/20%/20% train/dev/test split as in [Mudgal *et al.*, 2018], and use Adam algorithm [Kingma and Ba, 2014] for optimization. We use three model settings in our experiments:

- MPM-ave: A variant of our model, which doesn't conduct similarity selection but simply averages all similarity measure results. For each attribute A, all similarity measures' comparison results outputted by the comparison layer are directly inputted to the aggregation layer. After projecting them to the same vector space, the aggregation layer employs an element-wise average operation to get the final comparison vector of the attribute.
- MPM-soft: A variant of our model, which performs soft similarity measure selection. Specifically, given an attribute A, its final comparison vector is the weighted sum of all comparison vectors from different measures, and the weight of each similarity measure is from the soft selection vector $\boldsymbol{s}$ in Formula 1.
- MPM: The full model proposed in this paper, which performs hard similarity measure selection, i.e., adaptively selects only the optimal similarity measure for each attribute in the selection layer.

3.2 Overall Results

Table 3 shows the performance of our models and all baselines. From Table 3, we can see that:

1) By performing end-to-end multi-perspective entity matching, our model significantly outperforms previous methods. Compared with the state-of-the-art non-deep learning system Magellan, our system achieved 2.4% and 44% F1 improvement on two datasets correspondingly. Compared with the recently proposed neural network baselines (RNN and Hybrid), our system achieves 8.9% and 2% F1 improvement on two datasets correspondingly.

2) Due to the heterogeneity of attributes, multi-perspective matching is critical for entity resolution. Compared with RNN and Hybrid which only use one similarity measure for all attributes, our method can achieve 8.9% and 2% F1 improvement on two datasets correspondingly. We believe this is because DL-based similarity measures only work well on textual data, and cannot accurately capture the similarity between attribute values of other types. Our method achieves more performance improvement on *Walmart-Amazon* than on *Amazon-Google*, we believe it may be because *Walmart-Amazon* (5 attributes) has more attributes than *Amazon-Google* (3 attributes), therefore the heterogeneity of attributes impacts more on the final performance.

System	Walmart-Amazon			Amazon-Google		
	P	R	F1	P	R	F1
Magellan	72.3	71.5	71.9	67.7	38.5	49.1
RNN	70.9	64.6	67.6	69.5	52.6	59.9
Hybrid	78.3	58.3	66.9	61.7	79.1	69.3
MPM-ave	67.0	65.3	66.1	55.2	77.3	64.4
MPM-soft	73.4	64.2	68.5	61.3	73.9	67.1
MPM	74.2	73.1	**73.6**	67.4	73.5	**70.7**

Table 3: The results of our systems and baselines on *Walmart-Amazon* and *Amazon-Google* datasets. For Magellan, RNN and Hybrid baselines, we directly use the performance reported by Mudgal *et al.* [2018] for fair comparison.

3) Similarity selection is critical for entity resolution. Both similarity selection models (MPM-soft, MPM) can achieve significant F1 improvement over the non-selection model (MPM-ave). Specifically, compared with MPM-ave, MPM-soft achieved 3.6% and 4.2% F1 improvements on two datasets correspondingly, and MPM achieved 11.3% and 9.6% F1 improvements correspondingly.

4) Hard similarity selection mechanism is more effective than soft selection. By employing a gate mechanism to select the most appropriate similarity measure for each attribute, MPM outperformed MPM-soft by 7.4% and 5.4% F1 score on two datasets correspondingly.

3.3 Detailed Analysis

Effects of measure selection. To analyze the effects of our selection module, we show the selected similarity measures for each attribute in Table 4. From the results we can see that, our model can accurately select appropriate measure for different attributes, i.e., most selections are reasonable. For instance, for textual attributes *Title*, our model selects the DL-based similarity measure *hybrid_sim* in both datasets. For string attributes (*category, brand, model* in *Walmart-Amazon*,

Dataset	Attribute	Measure	Type
Walmart-Amazon	Title	*hybrid_sim*	DL
	Category	*exact_sim*	String
	Brand	*jaro_sim*	String
	Model	*exact_sim*	String
	Price	*rnn_sim*	DL
Amazon-Google	Title	*hybrid_sim*	DL
	Manufacturer	*exact_sim*	String
	Price	*rel_diff*	Numeric

Table 4: Selected similarity measures and their types for attributes in *Walmart-Amazon* and *Amazon-Google* datasets.

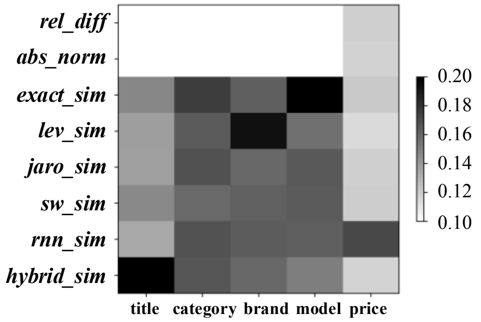

Figure 3: Soft selection result on *Walmart-Amazon* dataset. For each attribute, the stronger the color of a measure, the larger its weight.

and *manufacturer* in *Amazon-Google*), our model selects two string-based similarity measures (*exact_sim* and *jaro_sim*). For numeric attribute *price* in *Amazon-Google*, the numeric measure *rel_diff* is selected. This is intuitive that numeric measures are usually better at capturing number relations. Note that, for the numeric attribute *price* in *Walmart-Amazon*, a DL-based measure *rnn_sim* has been selected instead of a numeric measure. We observed that this is due to product *prices* in this data set are noisy, e.g., the same product may have a large price gap in two sources. Therefore, the numeric measures of *prices* will introduce noise, rather than provide helpful evidence for final decisions.

Effects of soft selection. We also demonstrate the soft attention vectors in MPM-soft in Figure 3. We can see that, for each attribute, our gate mechanism can assign higher weights to appropriate measures. For instance, for attribute *title***,** its highest attention weight is assigned to the DL-based measure *hybrid_sim*; and for the attribute *model*, it assigned the highest weight to the string measure *exact_sim*, which is better at measuring values having only binary relation.

Effects of multi-perspective matching. To further analyze the effects of multi-perspective matching, we conduct similarity measure ablation experiments on our model and its results are shown in Table 5. We can see that all similarity measures contribute to the final ER decisions, i.e., removing measures usually will result in a performance decline. This verified the effectiveness of multi-perspective matching.

	Walmart-Amazon		**Amazon-Google**	
	F1	ΔF1	F1	ΔF1
MPM	**73.6**	-	**70.7**	-
-Numeric	72.2	-1.4	70.1	-0.6
-String	70.6	-3.0	70.9	+0.2
-DL	67.1	-6.5	38.9	-31.8

Table 5: Ablation test on two datasets, removing each type of similarity measures separately.

4 Related Work

The existing ER approaches can be roughly divided into three categories: rule-based, crowdsourcing-based, and machine learning-based (traditional and recent DL-based). Rule-based approaches resolve entity record pairs using matching rules given by domain experts [Hernández and Stolfo, 1995; Arvind *et al*., 2009] or automatically learned from labeled examples [Chaudhuri *et al*., 2007; Wang *et al*., 2011; Singh *et al*., 2017]. Crowdsourcing-based approaches leverage crowd workers for entity matching problems via crowdsourcing platforms [Firmani *et al*., 2016]. Most current machine learning (ML)-based approaches are variants of the Fellegi-Sunter model [Fellegi *et al*., 1969], which treats entity resolution as a classification problem. Traditional ML approaches mostly design similarity measures as features and learn a classifier to resolve entities, which include SVM-based models [Bilenko and Mooney, 2003], active learning-based solutions [Sarawagi and Bhamidipaty, 2002], clustering-based techniques [Cohen and Richman, 2002], and Markov logic-based methods [Singla *et al*., 2006], etc. Recently, due to its strong representation learning ability, deep learning is also used for ER. The main advantage of DL-based methods is that they can better capture semantic similarity between textual attribute values, and can efficiently reduce human cost in ER pipeline [Ebraheem *et al*., 2018; Mudgal *et al*., 2018]. In this paper, we further extend DL-based ER methods, so that it can also perform adaptive measure selection.

Aware of the importance of measure selection, many methods have been proposed. Kong *et al*., [2016] manually selected similarity measures for different attributes. Chaudhuri *et al.* [2007] selected similarity measures and thresholds for ER rules by recursively constructing operation trees via a divide and conquer strategy, where each operation tree corresponds to a union of multiple similarity joins. Wang *et al.* [2011] proposed three heuristic algorithms to detect and eliminate redundancy among similarity functions and thresholds for ER rules. There are also many soft selection approaches which assign an importance score to different measures [Nikolov *et al*., 2012; Jurek *et al*., 2017; Jurek *et al*., 2018]. Compared to previous measure selection studies for ER, this paper is the first study to incorporate similarity measure selection into end-to-end neural frameworks.

5 Conclusions

This paper proposes an end-to-end multi-perspective matching model for entity resolution, which can adaptively select the optimal similarity measures for heterogenous attributes, and jointly learn and select similarity measures in an end-to-end way. Experimental results show that our method can significantly outperform previous methods and adaptively select optimal similarities in different ER problems. For future work, we want to take the dirty data problem into consideration, e.g., missing or noisy attribute values which are common in real-world datasets/applications.

Acknowledgments

This work is supported by the National Key Research and Development Program of China under Grant No.2017YFB1002104; and the National Natural Science Foundation of China under Grants no. 61433015, 61572477 and 61772505, and the Young Elite Scientists Sponsorship Program no. YESS20160177.

References

[Mudgal *et al.*, 2018] Sidharth Mudga, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. Deep Learning for Entity Matching: A Design Space Exploration. In *SIGMOD*, 2018, pp. 19–34.

[Ebraheem *et al.*, 2018] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. Distributed representations of tuples for entity resolution. *PVLDB*, 2018, 11(11): 1454-1467.

[Köpcke *et al.*, 2010] Hanna Köpcke, Andreas Thor, Erhard Rahm. Evaluation of entity resolution approaches on real-world match problems. *PVLDB*, 2010, 3(1-2): 484-493.

[Konda *et al.*, 2016] Pradap Konda, Sanjib Dasl et al. Magellan: Toward building entity matching management systems. *PVLDB*, 2016, 9(12): 1197-1208.

[Bojanowski *et al.*, 2016] Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov. Enriching Word Vectors with Subword Information. *TACL*, 2017, 5: 135-146.

[Kingma and Ba, 2014] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Srivastava *et al.*, 2015] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. In *Proceedings of ICML*, 2015.

[Levenshtein, 1966] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady.* 1966, 10(8): 707-710.

[Jaro, 1980] M. A. Jaro. UNIMATCH, a Record Linkage System: Users Manual. *Bureau of the Census*, 1980.

[Smith and Waterman, 1981] T. F. Smith and M. S. Waterman, Identification of Common Molecular Subsequences. *Journal of molecular biology*, 1981, 147(1): 195-197.

[Wang *et al.*, 2011] Jiannan Wang, Guoliang Li, Jeffrey Xu Yu, and Jianhua Feng. Entity matching: How similar is similar. *PVLDB*, 2011, 4(10): 622-633.

[Elmagarmid *et al.*, 2007] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate record detection: A survey. *Journal of TKDE*, 2006, 19(1): 1-16.

[Christen, 2012] Peter Christen. A survey of indexing techniques for scalable record linkage and deduplication. *Journal of TKDE*, 2011, 24(9): 1537-1555.

[Kong *et al.*, 2016] Chao Kong, Ming Gao, Chen Xu, Weining Qian, and Aoying Zhou. Entity matching across multiple heterogeneous data sources. *DASFAA*, 2016. Springer, Cham, 2016, pp. 133-146.

[Benjelloun *et al.*, 2009] Omar Benjelloun, Hector Garcia-Molina, David Menestrina, Qi Su, Steven Euijong Whang, and Jennifer Widom. Swoosh: a generic approach to entity resolution. *The VLDB Journal*, 2009, 18(1): 255-276.

[Bilke and Naumann, 2005] Alexander Bilke, Felix Naumann. Schema matching using duplicates. In *Proceedings of ICDE*, 2005, pp. 69-80.

[Koudas *et al.*, 2004] Nick Koudas, Amit Marathe, and Divesh Srivastava, Flexible String Matching against Large Databases in Practice. *PVLDB*, 2004.

[Bilenko and Mooney, 2003] Mikhail Bilenko and Raymond J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *SIGKDD*, 2003, pp. 39-48.

[Papadakis *et al.*, 2016] George Papadakis, Jonathan Svirsky, Avigdor Gal, and Themis Palpanas. Comparative Analysis of Approximate Blocking Techniques for Entity Resolution. *PVLDB*, 2016, 9(9): 684-695.

[Sarawagi and Bhamidipaty, 2002] Sunita Sarawagi and Anuradha Bhamidipaty. Interactive deduplication using active learning. In *SIGKDD*, 2002, pp. 269-278.

[Cohen and Richman, 2002] William W. Cohen and Jacob Richman. Learning to match and cluster large high-dimensional data sets for data integration. In *SIGKDD*, 2002, pp. 475-480.

[Chaudhuri *et al.*, 2007] Surajit Chaudhuri, Bee-Chung Chen, Venkatesh Ganti, and Raghav Kaushik. Example-driven design of efficient record matching queries. *PVLDB*, 2007, pp. 327-338

[Sehgal *et al.*, 2006] Vivek Sehgal, Lise Getoor, and Peter D Viechnicki. Entity resolution in geospatial data integration. In *ACM SIGSPATIAL*, 2006, pp. 83-90.

[Singh *et al.*, 2017] R Singh, V Meduri, A K Elmagarmid, S Madden, P Papotti, J Quiané-Ruiz, A Solar-Lezama, and N Tang. Synthesizing entity matching rules by examples. *PVLDB*, 2017, 11(2):189-202

[Hernández and Stolfo, 1995] Mauricio A. Hernández, Salvatore J. Stolfo. The merge/purge problem for large databases. *ACM Sigmod Record*. ACM, 1995, 24(2): 127-138.

[Arvind *et al.*, 2009] Arvind Arasu, Christopher Ré, and Dan Suciu, Large-scale deduplication with constraints using dedupalog, In *Proceedings of ICDE*, 2009, pp. 952–963.

[Fellegi *et al.*, 1969] Ivan P. Fellegi and Alan B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64 (328):1183-1210, 1969.

[Singla *et al.*, 2006] Parag Singla and Pedro Domingos. Entity resolution with markov logic. In *ICDM*, 2006.

[Firmani *et al.*, 2016] Donatella Firmani, Barna Saha, and Divesh Srivastava. Online entity resolution using an oracle. *PVLDB*, 2016, 9(5): 384-395.

[Nikolov *et al.*, 2012] Andriy Nikolov, Mathieu d'Aquin, and Enrico Motta. Unsupervised learning of link discovery configuration. In *ESWC*, 2012.

[Jurek *et al.*, 2017] Anna Jurek, Jun Hong, Yuan Chi, Weiru Liu. A novel ensemble learning approach to unsupervised record linkage. *Information Systems*, 2017, 71: 40-54.

[Jurek *et al.*, 2018] Anna Jurek and Deepak P. It pays to be certain: unsupervised record linkage via ambiguity minimization. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2018.