# Hierarchical Matching Network for Heterogeneous Entity Resolution

**Cheng Fu**[1,3*], **Xianpei Han**[1,2*], **Jiaming He**[4] and **Le Sun**[1,2]
[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]Brandeis University
{fucheng, xianpei, sunle}@iscas.ac.cn, germing@cs.brandeis.edu

## Abstract

Entity resolution (ER) aims to identify data records referring to the same real-world entity. Most existing ER approaches rely on the assumption that the entity records to be resolved are homogeneous, i.e., their attributes are aligned. Unfortunately, entities in real-world datasets are often *heterogeneous*, usually coming from different sources and being represented using different attributes. Furthermore, the entities' attribute values may be redundant, noisy, missing, misplaced, or misspelled—we refer to it as the dirty data problem. To resolve the above problems, this paper proposes an end-to-end hierarchical matching network (HierMatcher) for entity resolution, which can jointly match entities in three levels—*token, attribute,* and *entity*. At the token level, a cross-attribute token alignment and comparison layer is designed to adaptively compare heterogeneous entities. At the attribute level, an attribute-aware attention mechanism is proposed to denoise dirty attribute values. Finally, the entity level matching layer effectively aggregates all matching evidence for the final ER decisions. Experimental results show that our method significantly outperforms previous ER methods on homogeneous, heterogeneous and dirty datasets.

## 1 Introduction

Entity resolution (ER) aims to identify entity records referring to the same real-world entity from different data sources, which is important 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 records from Walmart and Amazon refer to the same product, and knowing this fact can help in product search, product recommendation, etc.

Currently, most existing ER approaches rely on the assumption that all entity records to be resolved are homogeneous, i.e., they consist of the same attributes. Based on this assumption, given two entity records, typical ER approaches first compare values between aligned attributes, then aggregate comparison results of all attributes to make the final ER decision [Benjelloun *et al.*, 2009; Mudgal *et al.*, 2018; Bhattacharya and Getoor, 2007; Mudgal *et al.*, 2018; Bhattacharya and Getoor, 2007; Ailon *et al.*, 2008]. Take the two entity records in Figure 1 as an example, the similarities between their aligned attributes (e.g., $< title, description >$ and $< price, retailprice >$) will be first computed independently and then aggregated to predicate whether they refer to the same laptop bag.

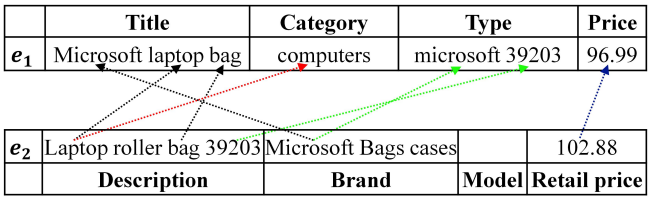


Figure 1: Demonstrations of heterogeneous and dirty entity records, and cross-attribute token alignment between them.

The above entity homogeneity assumption, unfortunately, often suffers from schema heterogeneity and dirty data problems when applied to real-world scenarios:

**Schema heterogeneity.** In many ER problems, entities usually come from different sources and are represented using heterogeneous schemas, i.e., they have different attributes. For instance, the first product $e_1$ in Figure 1 is described using attributes $\{title, category, type, price\}$, and the second one $e_2$ uses different attributes $\{description, brand, model, retailprice\}$. To enable aligned attribute-based approaches to these heterogeneous entity records, an additional schema matching step is needed to align attributes [Rahm and Bernstein, 2001; Bilke and Naumann, 2005]. However, due to the complex relationships between heterogeneous attributes, schema matching is not a trivial task. For instance, there are three types of attribute alignments in Figure 1: **a) 1-to-1:** *price* in $e_1$ corresponds to *retail price* in $e_2$; **b) 1-to-N:** *title* in $e_1$ corresponds to *description* and *brand* in $e_2$; **c) N-to-N:** *category* and *type* in $e_1$ together correspond to *brand* and *model* in $e_2$. Furthermore, because schema matching and entity matching are conducted independently, it often results in error propagation and causes difficulty in the global optimization of ER systems.

* Corresponding author.

**Dirty data.** In real-world datasets, it is very common that attribute values may be redundant, noisy, missing, misplaced, or misspelled – this paper refers it as *the dirty data problem*. For instance, in Figure 1, model "39203" of the second product entity $e_2$ is misplaced in its title. It is obvious that aligned attribute-based ER approaches can not effectively solve dirty entity resolution problem – by restricting comparison only between aligned attributes, they can not capture evidence across different attributes (e.g., similarity of product model between the entity records in Figure 1), and different attributes can not share and reinforce with each other.

Some recently proposed deep ER models can be directly applied to such heterogeneous and dirty ER scenarios by treating a whole entity record as a sequence of tokens, e.g., DeepER [Ebraheem *et al.*, 2018] and DeepMatcher [Mudgal *et al.*, 2018], but they completely ignore entity structure information which is usually critical for ER task. Seq2SeqMatcher [Nie *et al.*, 2019] is a work specially designed for heterogeneous ER, whose motivation is quite similar to ours. However, it only models structure information by learning an embedding vector for each attribute, and mainly applies them to the token-level matching, therefore also can not make full use of hierarchical structure informations of entities.

To address the above two challenges, this paper proposes a hierarchical matching network (HierMatcher), which can jointly model entity matching at three levels (*token, attribute,* and *entity*) in a unified neural framework. At the token level, we construct a cross-attribute token alignment module. By selecting comparison objects for all tokens across all attributes, it can effectively address the schema heterogeneity and the misplaced-type dirty data problems. At the attribute level, we design an attribute-aware attention mechanism, which can learn to identify important information for different attributes, therefore can effectively resolve the redundant-type and noisy-type dirty data problems. Furthermore, by obtaining matching evidence level by level, i.e., aggregating comparison results from token level to attribute level, and then to entity level, our model can fully take advantage of hierarchical structure information of entities. Finally, because all alignment, comparison and aggregation components in our framework are learnable, our model can be globally optimized in an end-to-end way, which prevents the error propagation problem and the local optimum problem.

We evaluate our approach on ten datasets. Experimental results show that, by adaptively selecting cross-attribute matching objects for tokens and effectively identifying important information of each attribute, our method significantly outperforms previous methods on all three kinds of datasets (homogeneous, heterogeneous and dirty).

## 2 Entity Resolution via Hierarchical Matching Network

This section describes how to resolve heterogeneous entities via our hierarchical matching network model. We first introduce our framework, then present the cross-attribute token alignment module used for selecting the most similar tokens from the other entity, and then describe the attribute-aware attention mechanism used for identifying important information of each attribute. Finally, we describe the entity level matching of our model.

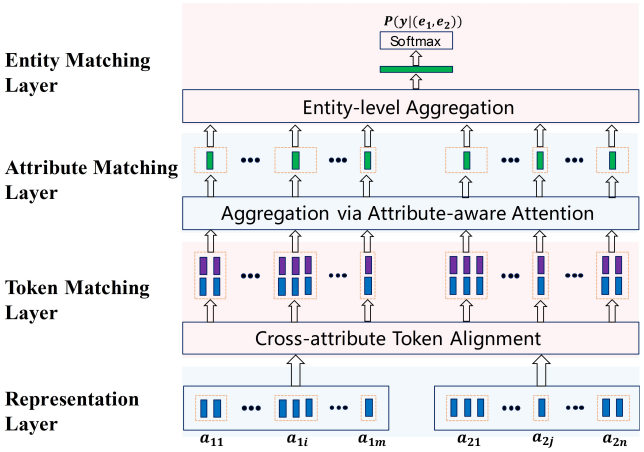


Figure 2: Framework of our hierarchical matching network (HierMatcher) for heterogeneous entity resolution.

Formally, given two entity records $e = \{< A_{11}, a_{11} >, \ldots, < A_{1m}, a_{1m} >\}$ and $e^{'} = \{< A_{21}, a_{21} >, \ldots, < A_{2n}, a_{2n} >\}$ respectively from two data sources $E$ and $E^{'}$, with heterogeneous attributes $\{A_{11}, A_{12}, \ldots, A_{1m}\}$ and $\{A_{21}, A_{22}, \ldots, A_{2n}\}$, where each attribute value $a_{ij}$ is a token sequence, 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 can be very large (e.g., Amazon contains millions of products), a complete ER system usually performs an additional blocking step to find candidate pairs, which has been extensively studied and well addressed [Christen, 2011; Kenig and Gal, 2013; Papadakis *et al.*, 2016]. Like previous entity matching studies [Mudgal *et al.*, 2018; Fu *et al.*, 2019], this paper focuses on entity matching and will not describe blocking.

### 2.1 Hierarchical Entity Matching Framework

Our neural entity matching framework is shown in Figure 2. Given two heterogeneous entity records, we first encode all attribute values via a bi-directional RNN (representation layer), then align and compare all tokens via a cross-attribute token alignment module (token matching layer), and then perform attribute level comparisons employing an attribute-aware attention mechanism (attribute matching layer), finally aggregate all attribute level comparison results to get entity level matching evidence and make the final ER decisions (entity matching layer). In following we describe our framework layer by layer.

**Representation Layer.** In this layer, we take each attribute value as a token sequence and get an embedding vector and a contextual vector for each token. Formally, given an attribute value $a$ with tokens $w_t, t \in [1, T]$, where $T$ is the token number of $a$, we first represent each token as a pre-trained embedding vector $\boldsymbol{x}_t \in \mathbb{R}^d$, then utilize a bi-directional GRU (BiGRU) [Cho *et al.*, 2014] to incorporate their contextual information. We denote hidden state at the $t$th time-step of the BiGRU as $\boldsymbol{h}_t \in \mathbb{R}^{2u}$, where $u$ is the hidden size, and take it as contextual vector representation of $w_t$.

**Token Matching Layer.** Given an attribute value, this layer first aligns all tokens it contains using a cross-attribute alignment module, then compare each token with its aligned token from the other entity. In this way, alignment of each token may come from any attribute of the other entity. Therefore, schema homogeneity constraint existed in previous aligned attribute-based approaches can be broken here. Details of this layer will be described in Section 2.2.

**Attribute Matching Layer.** This layer aggregates all token level comparison results of each value to generate its attribute level matching evidence using an attribute-aware attention mechanism, which can adaptively identify important information for each attribute. Details of this layer will be described in Section 2.3.

**Entity matching Layer.** In this layer, we first concatenate all attribute level comparison results of e and $e^{'}$ into a comparison vector, and make final decisions based on it 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. Details of this layer will be described in Section 2.4.

**Model Learning.** Given a training set D that 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 $y_i$ output by our model.

We can see that, the proposed framework models all token level alignment, hierarchical comparison and aggregation as learnable neural layers in a single neural network. The main advantage of our framework is that all components can be learned end-to-end, allowing for easy global optimization of the ER framework.

## 2.2 Token Matching via Cross-Attribute Alignment

In heterogeneous and dirty entity records, corresponding information of one attribute may be scattered in multiple attributes of the other entity. For example, the *brand* attribute of $e_2$ in Figure 1 consists of two types of information: product category and brand information, which are correspondingly contained in the *category* attribute and *type* attribute of $e_1$. It means that, to obtain an accurate comparison object of an attribute value, we must look over the whole entity, instead of only one specific attribute. To this end, we design a cross-attribute token alignment module, which uses a global selection mechanism to select the most similar token from the other entity for each token.

Take the $t$th token $w_{1it}$ of the $i$th attribute value in entity record $e$ as an example, let its contextual representation be $\boldsymbol{h}_{1it} \in \mathbb{R}^{2u}$, and let contextual representation matrix of record $e^{'}$ be $\boldsymbol{H}_2 \in \mathbb{R}^{2u \times Q}$, where $Q$ is the total number of all tokens in $e^{'}$ and therefore each column of $\boldsymbol{H}_2$ represents contextual representation of a token. Via an element-wise absolute difference comparison operation, we first compare $w_{1it}$ with all tokens in entity $e^{'}$ to get a compare matrix $\boldsymbol{C}_{1it} \in \mathbb{R}^{2u \times Q}$:

$$\boldsymbol{C}_{1it} = |\boldsymbol{h}_{1it} \bigotimes \boldsymbol{e}_Q - \boldsymbol{H}_2| \tag{1}$$

where the outer product $(. \bigotimes \boldsymbol{e}_Q)$ produces a matrix by repeating the vector on the left for $Q$ times. Then we feed the compare matrix through a two-layer HighwayNet [Srivastava *et al.*, 2015], and transform the outputted matrix $\boldsymbol{G}_{1it} \in \mathbb{R}^{2u \times Q}$ to an attention vector $\boldsymbol{v}_{1it} \in \mathbb{R}^{Q}$ using a linear layer followed by a softmax function:

$$\boldsymbol{G}_{1it} = HighwayNet(\boldsymbol{C}_{1it}) \tag{2}$$

$$\boldsymbol{v}_{1it} = softmax(\boldsymbol{w}\boldsymbol{G}_{1it} + b) \tag{3}$$

where $\boldsymbol{w} \in \mathbb{R}^{2u}$ and $b \in \mathbb{R}^1$ are parameters to be learned. After that, we transform the attention vector to a one-hot selection vector $\boldsymbol{s}_{1it} \in \mathbb{R}^{Q}$ by an element-wise function $f$ which assigns 1 to $\boldsymbol{s}_{1it}[l]$ if $\boldsymbol{v}_{1it}[l] == max(\boldsymbol{v}_{1it})$, otherwise 0:

$$\boldsymbol{s}_{1it} = f(\boldsymbol{v}_{1it}) \tag{4}$$

Finally, we can use the global selection vector $\boldsymbol{s}_{1it}$ to pick out the aligned token for $w_{1it}$, and take its final token level matching result as:

$$\boldsymbol{r}_{1it} = \boldsymbol{C}_{1it}\boldsymbol{s}_{1it}^{\mathsf{T}} \tag{5}$$

We can see that, no matter how complex the relationships between attributes are, or whether values are misplaced, by using the above cross-attribute token alignment module, we can adaptively get suitable matching object for each attribute, therefore can effectively address the schema heterogeneity and misplaced-type dirty data problems.

## 2.3 Attribute Matching via Attribute-aware Attention

Given results output by the token matching layer, we can match each attribute via effectively aggregating comparison results of all its tokens. One main challenge for attribute matching is that, the importance of different tokens in the same attribute are usually different, therefore their comparison results should be assigned different weights when being aggregated. Take the *brand* value of $e_2$ in Figure 1 as an example, its token "Microsoft" describing brand information is much more important than the other two ("Bags" and "cases"), which are used to describe the product category information. Furthermore, the dirty data problem may make some attribute values contain amounts of misplaced, redundant or noisy information, therefore how to denoise such attribute values is critical for entity resolution.

Inspired by the sentence level attention in [Yang *et al.*, 2016], we design an attribute-aware attention mechanism to address the above challenge. Our proposed attribute-aware attention mechanism can effectively identify important information for each attribute value. Specifically, we learn a context vector for each attribute, and model the importance of all tokens as their similarities with the context vector. One such

| Type | Dataset | Size | Attribute |
|---|---|---|---|
| Homogeneous | *Walmart-Amazon*$_1$ | 10,242 | 5 |
| | *Amazon-Google* | 11,460 | 3 |
| | *DBLP-ACM*$_1$n | 12,363 | 4 |
| | *DBLP-Scholar*$_1$ | 28,707 | 4 |
| Dirty | *Walmart-Amazon*$_2$ | 10,242 | 5 |
| | *DBLP-ACM*$_2$ | 12,363 | 4 |
| | *DBLP-Scholar*$_2$ | 28,707 | 4 |
| Heterogeneous | *Walmart-Amazon*$_3$ | 10,242 | <4, 5> |
| | *Walmart-Amazon*$_4$ | 10,242 | <4, 4> |
| | *Walmart-Amazon*$_5$ | 10,242 | <4, 4> |

Table 1: Statistics of all the datasets used in our experiments.

context vector can be seen as a high-level representation of a query "what is the informative token in this attribute".

Formally, given an attribute value $a_{1i}$ of entity record $e$ with $T$ tokens. On basis of its token level comparison results output by the previous token matching layer, we compute the attribute level comparison result of $a_{1i}$ as a weighted sum of comparison results of all its tokens:

$$\boldsymbol{r}_{1i} = \sum_{t=1}^{T} \alpha_{1it} \boldsymbol{r}_{1it} \tag{6}$$

where $\boldsymbol{r}_{1it}$ is the comparison vector of the $t$th token $w_{1it}$, and $\alpha_{1it}$ is an attention score representing the importance of $w_{1it}$. In this paper, $\alpha_{1it}$ is obtained as:

$$\alpha_{1it} = \frac{exp(\boldsymbol{p}_{it}^{\mathsf{T}} \boldsymbol{h}_{1it})}{\sum_{t=1}^{T} exp(\boldsymbol{p}_{it}^{\mathsf{T}} \boldsymbol{h}_{1it})} \tag{7}$$

where $\boldsymbol{p}_{it} \in \mathbb{R}^{2u}$ is the context vector of attribute $A_{1i}$, it is randomly initialized and jointly learned during training.

We can see that, by learning to assign more weights to important tokens, the proposed attention mechanism can reduce the influence of redundant and noisy information, therefore can effectively relieve the dirty data problem.

### 2.4 Entity matching Layer

Because heterogeneous entity records consist of different attributes, we need to aggregate attribute level comparison evidence from both entity records to make final ER decisions. Formally, to determine whether entity records $e$ and $e^{'}$ match, in this layer, we first concatenate all their attribute level comparison results as a $2u(m+n)$ dimension evidence vector:

$$\boldsymbol{r} = [\boldsymbol{r}_{11}; \boldsymbol{r}_{12}; ...; \boldsymbol{r}_{1m}; \boldsymbol{r}_{21}; \boldsymbol{r}_{22} ...; \boldsymbol{r}_{2n}] \tag{8}$$

then feed $\boldsymbol{r}$ into a two-layer fully-connected ReLU HighwayNet followed by a softmax layer, which outputs the matching probability $P(y = 1|e, e^{'})$.

## 3 Experiments

In this section, we evaluate our method and compare it with previous methods. Our code is freely available online[1].

[1]https://github.com/cipnlu/EntityMatcher

### 3.1 Datasets

We conduct experiments on ten datasets of three types, whose statistics are shown in Table 1:

- **Four homogeneous datasets:** *Walmart-Amazon*$_1$, *Amazon-Google*, *DBLP-ACM*$_1$, and *DBLP-Scholar*$_1$, which are commonly used real-world datasets. Because this paper focuses on entity matching, we use their after-blocking versions provided by Mudgal *et al.* [2018].
- **Three dirty datasets:** *Walmart-Amazon*$_2$, *DBLP-ACM*$_2$, and *DBLP-Scholar*$_2$, which are individually derived from *Walmart-Amazon*$_1$, *DBLP-ACM*$_1$, and *DBLP-Scholar*$_1$ by randomly moving the value of each attribute to attribute title in the same tuple with 50% probability. These datasets are also provided by Mudgal *et al.* [2018].
- **Three heterogeneous datasets:** *Walmart-Amazon*$_3$, *Walmart-Amazon*$_4$, and *Walmart-Amazon*$_5$, which are all pseudo datasets derived from *Walmart-Amazon*$_1$ using different attribute merging operations. Specifically, for *Walmart-Amazon*$_3$, we merge *brand* and *model* attributes of entities from Walmart to get a new attribute *brand-model*. For *Walmart-Amazon*$_4$, we further merge the *category* and *brand* of entities from Amazon to get a new attribute *category-brand* on basis of *Walmart-Amazon*$_3$. For *Walmart-Amazon*$_5$, we merge the *category* and *model* of entities from Amazon to get a new attribute *category-model* on basis of *Walmart-Amazon*$_3$. After the above operations, there are respectively 1-to-2, 2-to-2 and 2-to-2 attribute corresponding relations in the three datasets.

### 3.2 Baselines

Four baselines are used in our experiments:

- **Magellan:** A state-of-the-art non-deep learning ER baseline proposed by Konda *et al.* [2016]. In this paper, We use its performance on all homogeneous and dirty datasets reported by Mudgal *et al.* [2018].
- **DeepMatcher:** A deep ER framework proposed by Mudgal *et al.* [2018], which consists of three modules: attribute embedding, attribute aimilarity representation and classifier. For the homogeneous and dirty datasets, we use its performance from Mudgal *et al.* [2018]. For the heterogeneous datasets, We use the performance from its implement in the open-source deepmatcher Python package.
- **MPM:** A deep ER model used for resolving attribute aligned entities proposed by Fu *et al.* [2019], which can adaptively select optimal similarity measures for heterogeneous attributes in an end-to-end way.
- **Seq2SeqMatcher:** A deep model proposed for heterogeneous ER tasks [Nie *et al.*, 2019], which models ER as a token-level sequence-to-sequence matching task. We re-implement this model and use it for *Amazon-Google* and the three heterogeneous datasets in our experiments.

| Type | Dataset | F1 score | | | | | | $\Delta F1$ | $\Delta F1'$ |
|---|---|---|---|---|---|---|---|---|---|
| | | Magellan | DeepMatcher | MPM | Seq2SeqMatcher | HierMatcher-ave | HierMatcher | | |
| Homogeneous | *Walmart-Amazon*$_1$ | 71.9 | 67.6 | 73.6 | 78.2 | 77.1 | **81.6** | +8.0 | +3.4 |
| | *Amazon-Google* | 49.1 | 69.3 | 70.7 | 61.2 | 70.0 | **74.9** | +4.2 | +13.7 |
| | *DBLP-ACM*$_1$ | 98.4 | 98.4 | - | **98.9** | 98.1 | 98.8 | +0.4 | -0.1 |
| | *DBLP-Scholar*$_1$ | 92.3 | 94.7 | - | **95.3** | 94.9 | **95.3** | +0.6 | +0.0 |
| Dirty | *Walmart-Amazon*$_2$ | 37.4 | 53.8 | - | 68.3 | 61.4 | **68.5** | +14.7 | +0.2 |
| | *DBLP-ACM*$_2$ | 91.9 | 98.1 | - | **98.4** | 97.3 | 98.1 | +0.0 | -0.3 |
| | *DBLP-Scholar*$_2$ | 82.5 | 93.8 | - | 94.1 | 93.9 | **94.5** | +0.7 | +0.4 |
| Heterogeneous | *Walmart-Amazon*$_3$(1-n) | - | 67.1 | - | 75.6 | 67.5 | **80.7** | +13.6 | +5.1 |
| | *Walmart-Amazon*$_4$(n-n) | - | 63.4 | - | 74.7 | 67.6 | **81.4** | +18.0 | +6.7 |
| | *Walmart-Amazon*$_5$(n-n) | - | 66.5 | - | 74.4 | 69.0 | **81.0** | +14.5 | +6.6 |

Table 2: The results of our systems and baselines on all three types of datasets. $\Delta F1$ denotes F1 gains of our HierMatcher system compared with all the aligned attribute-based baselines (Magellan, DeepMatcher and MPM). And $\Delta F1'$ denotes F1 gains of HierMatcher compared with Seq2SeqMatcher, which is also specially designed for resolving heterogeneous and dirty entities.

### 3.3 System Settings

We use two model settings in our experiments:

- **HierMatcher-ave:** A variant of our model, which does not distinguish the importance of different tokens when conducting the attribute level comparison. Given all token level comparison vectors of an attribute, it simply performs an element-wise average operation to get the final attribute comparison result.
- **HierMatcher:** The full entity matching model proposed in this paper, which uses the attribute-aware attention mechanism to identify important information for each attribute. Given all token level comparison results, it performs a weight-sum operation to get the final attribute level matching evidence.

For our systems, we use the pre-trained FastText 300-dimensional word embedding [Bojanowski *et al.*, 2017] and fix the embeddings during training. The hidden size of each GRU layer is set 150. For model learning, we use the same 60%/20%/20% train/dev/test split as in [Mudgal *et al.*, 2018], and use Adam algorithm for optimization. Following previous studies, we evaluate all systems using precision (P), recall (R), and F1 score, and F1 is used as the primary measure.

### 3.4 Overall Results

The performance of our models and all baselines are shown in Table 1. From the table we can see that:

- By performing hierarchical cross-attribute entity matching, our model can effectively solve the schema heterogeneity and dirty data problems in ER. Specifically, compared with the state-of-the-art aligned attribute-based baselines (Magellan, DeepMatcher and MPM), our best system HierMatcher achieves 15.4 and 5.1 average F1 score improvements correspondingly on the heterogeneous and the dirty datasets. Even compared with Seq2SeqMatcher model, which is also specially designed for heterogeneous ER, HierMatcher achieves 6.1 and 0.1 average F1 score improvements correspondingly on the heterogeneous and the dirty datasets. This is mainly because, via cross-attribute token alignment, our model can adaptively obtain the optimal matching object for each attribute value, even if it is misplaced or broken into multiple parts, which are respectively distributed in different attributes. Furthermore, by employing a three-level hierarchical matching strategy, our model can make full use of entity structure information.
- Due to the prevalence of dirty data (redundant, noisy, etc.) in real-world applications, effectively identifying important information for each attribute is critical for entity resolution. Compared to the full model HierMatcher, our HierMatcher-ave system gets large performance declines on all three types of datasets. Specifically, because of treating all tokens equally when aggregating their comparison results, HierMatcher-ave gets 2.6, 2.8, and 13.0 average F1 score reductions correspondingly on the homogeneous, dirty, and heterogeneous datasets. We can see that, the performance declines on the dirty and the heterogeneous datasets are larger than on the homogeneous datasets, this is because their entity records are dirtier. Especially for the three heterogeneous datasets, tokens in their merged attribute values may seriously interfere with each other, therefore can result in large performance decline. The above results further demonstrate the necessity of our attribute-aware attention mechanism in entity resolution tasks.
- Our matching model is quite robust in multiple types of entity resolution scenarios. We can see that, even without using explicit attribute alignment information, our model can significantly outperform all baselines on the two challenging homogeneous datasets – *Walmart-Amazon*$_1$ and *Amazon-Google*. Specifically, correspondingly on them, HierMatcher system achieves 8.0 and 4.2 F1 improvements compared with all aligned attribute-based baselines, and achieves 3.4 and 13.7 F1 improvements compared with Seq2SeqMatcher. We believe this is because attribute alignment information of these datasets can be adaptively learned by our model. Besides, there are also some dirty records existed in them, i.e., records with missing, misplaced or noisy values, which can be better resolved by our model. This effectively demonstrates the universality of our approach.

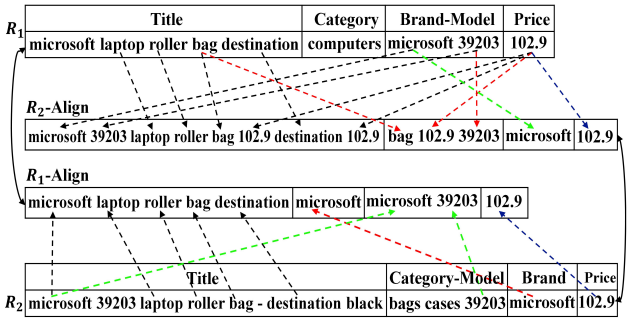


Figure 3: Examples of alignment results output by the cross-attribute alignment module. $R_1$ and $R_2$ are a pair of records to be compared, and $R_1$-*Align* and $R_2$-*Align* are their alignment results.

### 3.5 Detailed Analysis

**Effects of the cross-attribute token alignment.** In order to illustrate the effect of our cross-attribute token alignment module, we take a pair of entity records $< R_1, R_2 >$ from heterogeneous dataset *Walmart-Amazon*$_5$ as an example and display their alignment results in Figure 3. We can see that, although counterparts of some attribute values in $R_1$ are scattered in multiple attributes of the record $R_2$, and there are different attribute corresponding relations (1-1, 1-2, 2-2) between the two records, our cross-attribute alignment module can learn to adaptively get suitable matching object for each attribute value. For example, for the key attribute *brand-model* in $R_1$, our model correctly selects "microsoft" from the *title* of $R_2$ as the alignment of its token used to describe brand information, and selects "39203" from *category-model* of $R_2$ as the alignment of its token used to describe product model information. The above results further demonstrate the effectiveness of our cross-attribute token alignment module.

**Effects of the attribute-aware attention mechanism.** To illustrate the effect of our attribute-aware attention mechanism used in the attribute matching layer, we display attention scores of all tokens in an entity record pair $< R_1, R_2 >$ (the same as the one in Figure 3) from *Walmart-Amazon*$_5$ in Figure 4. We can see that, by using the attention mechanism, our model can learn to assign higher weights to more important tokens. For example, in the value of *category-model*, the highest attention (0.95) is assign to the token "39203" representing the model of the product, and attention scores of the two tokens "bags cases" describing product category information are quite close to zero (0.03 and 0.02), this is because product model information in this dataset is much more discriminating for ER task than product category information. This example further demonstrates the effect of our attribute-aware attention mechanism in reducing influences of unimportant tokens on attribute level matching results.

| $R_1$ | Title | Category | Brand-Model | Price |
|---|---|---|---|---|
| | microsoft laptop roller bag destination | computers | microsoft 39203 | 102.9 |
| Att | 0.05 0.03 0.05 0.15 0.72 | 1.0 | 0.09 0.91 | 1.0 |

| $R_2$ | Title | Category-Model | Brand | Price |
|---|---|---|---|---|
| | microsoft 39203 laptop roller bag - destination black | bags cases 39203 | microsoft | 102.9 |
| Att | 0.04 0.62 0.11 0.13 0.04 0.01 0.02 0.03 | 0.03 0.02 0.95 | 1.0 | 1.0 |

Figure 4: Examples of importance weights output by our attribute-aware attention mechanism. *Att* denotes attention of each token.

## 4 Related Work

Entity resolution (ER) aims at detecting entity records which describe the same real-world object from given datasets. It has been extensively studied since the 1950s [Newcombe *et al.*, 1959], thus a variety of methods for solving the ER problem have been proposed [Doan and Halevy, 2005; Koudas *et al.*, 2006]. The existing ER approaches can be roughly divided into two categories: rule-based, and machine learning-based. Rule-based approaches resolve entity record pairs using matching rules given by domain experts [Hernández and Stolfo, 1995] or automatically learned from labeled examples [Chaudhuri *et al.*, 2007; Wang *et al.*, 2011; Singh *et al.*, 2017]. Machine learning (ML)-based approaches usually treat entity resolution as a classification problem [Fellegi and Sunter, 1969]. Traditional ML approaches include SVM-based models [Bilenko and Mooney, 2003], Markov logic-based methods [Singla and Domingos, 2006], active learning-based solutions [Sarawagi and Bhamidipaty, 2002], etc. Recently, some deep learning-base methods were also proposed for ER. One main advantage of such approaches is that they can better capture semantic similarity between textual attributes, and can efficiently reduce human cost in ER pipeline [Ebraheem *et al.*, 2018; Mudgal *et al.*, 2018; Fu *et al.*, 2019; Nie *et al.*, 2019].

Most of existing approaches focus on resolving homogeneous entity records with aligned attributes. When applied to real-world scenarios, such approaches usually suffer from schema heterogeneity and dirty data problems. To address these problems, some methods have been proposed, which includes traditional ML-based approaches [Lin *et al.*, 2019] and recent DL-based models [Ebraheem *et al.*, 2018; Mudgal *et al.*, 2018; Nie *et al.*, 2019]. These approaches either can not handle complex attribute corresponding relations, or can not make full use of entity structure information. Compared with previous studies, by designing a 3-level hierarchical matching framework, our model can not only directly resolve heterogeneous and dirty entities, but also take advantage of hierarchical structure information of entities.

## 5 Conclusions

This paper proposes a hierarchical matching network for entity resolution, which uses a cross-attribute token alignment module to adaptively compare heterogeneous entities, and employs an attribute-aware attention mechanism to denoise dirty attribute values. Experimental results show that our method outperforms previous methods on multiple types of datasets. In future work, we plan to further take the importance of different attributes into consideration when aggregating their comparison results to make the final ER decisions.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants no. U1936207 and 61772505, the National Key Research and Development Program of China under Grant No.2017YFB1002104, and Beijing Academy of Artificial Intelligence (BAAI2019QN0502).

## References


[Ailon *et al.*, 2008] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM*, 55(5):23, 2008.

[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*, 18(1):255–276, 2009.

[Bhattacharya and Getoor, 2007] Indrajit Bhattacharya and Lise Getoor. Collective entity resolution in relational data. *Journal of the TKDD*, 1(1):5, 2007.

[Bilenko and Mooney, 2003] Mikhail Bilenko and Raymond J Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ACM SIGKDD*, pages 39–48. ACM, 2003.

[Bilke and Naumann, 2005] Alexander Bilke and Felix Naumann. Schema matching using duplicates. In *Proceedings of the ICDE*, pages 69–80. IEEE, 2005.

[Bojanowski *et al.*, 2017] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *TACL*, 5:135–146, 2017.

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

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of EMNLP*, pages 1724–1734, 2014.

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

[Doan and Halevy, 2005] AnHai Doan and Alon Y Halevy. Semantic integration research in the database community: A brief survey. *AI magazine*, 26(1):83–83, 2005.

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

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

[Fu *et al.*, 2019] Cheng Fu, Xianpei Han, Le Sun, Bo Chen, Wei Zhang, Suhui Wu, and Hao Kong. End-to-end multi-perspective matching for entity resolution. In *Proceedings of the IJCAI*, pages 4961–4967. AAAI Press, 2019.

[Hernández and Stolfo, 1995] Mauricio A Hernández and Salvatore J Stolfo. The merge/purge problem for large databases. In *ACM Sigmod Record*, volume 24, pages 127–138. ACM, 1995.

[Kenig and Gal, 2013] Batya Kenig and Avigdor Gal. Mfi-blocks: An effective blocking algorithm for entity resolution. *Information Systems*, 38(6):908–926, 2013.

[Konda *et al.*, 2016] Pradap Konda, Sanjib Das, Paul Suganthan GC, AnHai Doan, Adel Ardalan, Jeffrey R Ballard, Han Li, Fatemah Panahi, Haojun Zhang, Jeff Naughton, et al. Magellan: Toward building entity matching management systems. *PVLDB*, 9(12):1197–1208, 2016.

[Kong *et al.*, 2016] Chao Kong, Ming Gao, Chen Xu, Weining Qian, and Aoying Zhou. Entity matching across multiple heterogeneous data sources. In *Proceedings of the DASFAA*, pages 133–146. Springer, 2016.

[Koudas *et al.*, 2006] Nick Koudas, Sunita Sarawagi, and Divesh Srivastava. Record linkage: similarity measures and algorithms. In *Proceedings of the ACM SIGMOD*, pages 802–803. ACM, 2006.

[Lin *et al.*, 2019] Yiming Lin, Hongzhi Wang, Jianzhong Li, and Hong Gao. Efficient entity resolution on heterogeneous records. *Journal of the TKDE*, 2019.

[Mudgal *et al.*, 2018] Sidharth Mudgal, 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 *Proceedings of the ICDM*, pages 19–34. ACM, 2018.

[Newcombe *et al.*, 1959] Howard B Newcombe, James M Kennedy, SJ Axford, and Allison P James. Automatic linkage of vital records. *Science*, 130(3381):954–959, 1959.

[Nie *et al.*, 2019] Hao Nie, Xianpei Han, Ben He, Le Sun, Bo Chen, Wei Zhang, Suhui Wu, and Hao Kong. Deep sequence-to-sequence entity matching for heterogeneous entity resolution. In *Proceedings of CIKM*, pages 629–638, 2019.

[Papadakis *et al.*, 2016] George Papadakis, Jonathan Svirsky, Avigdor Gal, and Themis Palpanas. Comparative analysis of approximate blocking techniques for entity resolution. *PVLDB*, 9(9):684–695, 2016.

[Rahm and Bernstein, 2001] Erhard Rahm and Philip A Bernstein. A survey of approaches to automatic schema matching. *the VLDB Journal*, 10(4):334–350, 2001.

[Sarawagi and Bhamidipaty, 2002] Sunita Sarawagi and Anuradha Bhamidipaty. Interactive deduplication using active learning. In *Proceedings of the ACM SIGKDD*, pages 269–278. ACM, 2002.

[Sehgal *et al.*, 2006] Vivek Sehgal, Lise Getoor, and Peter D Viechnicki. Entity resolution in geospatial data integration. In *Proceedings of ACM-GIS*, pages 83–90, 2006.

[Singh *et al.*, 2017] Rohit Singh, Venkata Vamsikrishna Meduri, Ahmed Elmagarmid, Samuel Madden, Paolo Papotti, Jorge-Arnulfo Quiané-Ruiz, Armando Solar-Lezama, and Nan Tang. Synthesizing entity matching rules by examples. *PVLDB*, 11(2):189–202, 2017.

[Singla and Domingos, 2006] Parag Singla and Pedro Domingos. Entity resolution with markov logic. In *Proceedings of the ICDM*, pages 572–582. IEEE, 2006.

[Srivastava *et al.*, 2015] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

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

[Yang *et al.*, 2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the NAACL-HLT*, pages 1480–1489, 2016.