

Ground Truth Inference for Weakly Supervised Entity Matching

RENZHI WU, ALEXANDER BENDECK, XUCHU, YEYE HE^{Original [1]} —Mingli Tan, Yongda Li, Ruichao Dai^{summary}

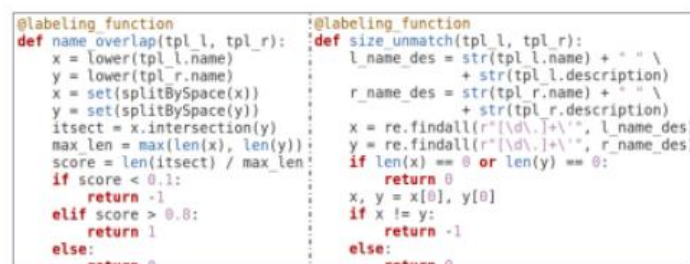
Abstract

The authors first emphasize the purpose of the paper and define what EM is, and briefly explain that traditional data models use strongly supervised machine learning, but the disadvantage is also obvious that large data sets are required. It is impossible to spend a lot of manpower to label data in actual work or life. Therefore, the author innovatively proposes the use of label function to match the relationship between data sets. In this paper, the author completes entity matching by using simple and powerful labeling tasks based on weak supervision, and focuses on introducing the concept of transitivity unique to EM and some comparison algorithms. Finally, experimental data are used to prove the advantages of the author's model, and the conclusion is that the author's labeling model results are 9% higher than the average value of existing best methods. Below we will be divided into several parts for a brief introduction and summary.

1 INTRODUCTION

First, the author explains what is entity matching? Entity matching is the process of processing data from two data sources to determine whether it is the same entity in the real world through various algorithms. In addition, EM has many practical functions, such as matching commodity list, building knowledge graph, and data integration. For data mining, a large number of data sets are initially required for data analysis, so that more meaningful data sets can be obtained through data matching of complicated data in the real world to facilitate more comprehensive data analysis. An idea proposed by the author here is to let the user write the tag function, because it is better for the user to understand the meaning of the data set and the direction of analysis.

For the tag function, the author specifically cited two examples, and explained the meaning of the tag function, as shown in the following figure.



```
@labeling_function
def name_overlap(tpl_l, tpl_r):
    x = lower(tpl_l.name)
    y = lower(tpl_r.name)
    x = set(splitBySpace(x))
    y = set(splitBySpace(y))
    itsect = x.intersection(y)
    max_len = max(len(x), len(y))
    score = len(itsect) / max_len
    if score < 0.1:
        return -1
    elif score > 0.8:
        return 1
    else:
        return 0

:@labeling_function
def size_unmatch(tpl_l, tpl_r):
    l_name_des = str(tpl_l.name) + " " + \
        + str(tpl_l.description)
    r_name_des = str(tpl_r.name) + " " + \
        + str(tpl_r.description)
    x = re.findall(r"[\d\.\.]+\.", l_name_des)
    y = re.findall(r"[\d\.\.]+\.", r_name_des)
    if len(x) == 0 or len(y) == 0:
        return 0
    x, y = x[0], y[0]
    if x != y:
        return -1
    else:
        return 0
```

Fig.1.Twouser-writtenLFsfortheabt-buydataset[1].

Here will explain the meaning of the data label, the essence is a python function, the left is to calculate the number of overlapping of two characters, and finally calculate the score to obtain the result of matching, the right is to calculate the number of mismatch between two characters, and return the result of matching through the final matching length.

The author also explains the need to create a label matrix, by using multiple different label functions between each pair of tuples to obtain the results, and finally compare these label results, and finally obtain the value of GT, which is the result of whether the tuple matches. The details are shown in the following figure:

	LF1	LF2	LF3	...	Naive Inferred Label	GT
(t ₁ , t ₂)	1	1	1	...	1	1
(t ₁ , t ₃)	1	1	-1	...	1	-1
(t ₁ , t ₄)	1	1	0	...	1	1
...
(t ₂ , t ₃)	0	-1	-1	...	-1	-1
(t ₂ , t ₄)	0	0	-1	...	-1	1

Fig.2.Exampleofalabelingmatrix[1]

Here the author also raises a question. Whether models can be labeled in a smarter way, such as random forests, avoiding the use of overly complex existing algorithms. After thinking about it, the author obtained a positive answer by inquiring documents or other ways, and explained the camera in the third chapter to get a better performance in a simple way. Here we can learn that the purpose is to solve the problem, so the simpler the better, and learn to ask yourself more, more to search, and finally find a conclusion or even an innovative idea.

2 PROBLEM DEFINITION

The author mainly proposed the use of tag functions to obtain matching results, but there may be noise or interdependence between these tag functions. Therefore, in order to improve the accuracy of matching and meet the transitivity requirements of entity matching, the author needs to design and apply a tag model to better combine the accuracy of tag functions. And introduce transitivity in matching.

3 PROPOSED LABELING MODEL

In this section, the author will introduce in detail the general form of the label model proposed by the author using the weakly supervised approach.

3.1 A Generic View of Labeling Models

Firstly, the author introduces a formula expression to abstract the proposed label model. Firstly, each label model can be regarded as a truth value inference method, which contains label matrix X and prediction vector. The author then wanted to express the problem as a mathematical relationship. Represents the result of the model with parameter Θ , which can be expressed as: since the data set is independent before, the prediction results are also independent, so the method is abstracted at last, the parameter Θ is obtained by the classification method, the problem is transformed into solving a classifier, and the parameter Θ is obtained, so that all label points can be predicted. The authors propose to use the expectation maximization algorithm to learn the Θ parameter and convert the objective function to solve the negative log-likelihood function, as shown in (1) below

$$L(\theta, X, \gamma) = - \sum_{i=1}^N \left(\gamma_i \log(g(x_i, \theta)) + (1 - \gamma_i) \log(1 - g(x_i, \theta)) \right) \quad (1)$$

Here, the author explains in detail how to obtain parameters through known data results. First, problem transformation, minimizing formula (1) is equivalent to maximizing label matrix X, and then solving the value of formula (1) through maximum expectation algorithm, which is divided into four steps. The first step is to obtain an initial value through majority voting, and then M steps are to learn model parameters. Step E updates through the fourth value, and step 4 continues 2 and 3 until the algorithm converges. At the same time, the author also suggests that the model can be instantiated by many other algorithms.

3.2 The SIMPLE Algorithm

The above mentioned g can be any classifier, but g can not be too large, resulting in excessive algorithm complexity or hypothesis space, and g can not be too small, because we want to capture the interaction and independence between different features. Here the author starts to think again. Based on the requirements of the appeal to g , the author asks whether similar results can be obtained by using a general classifier. Then the author gives a positive answer that yes, better results can be obtained by using random forests and limiting the capacity. Then the author explains why random forest is used. Firstly, the method based on tree structure can fully reflect the interrelationship between different features, which is also the need of our model. Secondly, tree-based methods perform well in structured data, and random forests happen to be tree-based. We can easily adjust the capacity of the classifier by setting parameters, and use cross-validation to get parameter values, and explain the role of M and E steps in model training.

The second problem is that there is a lot of class imbalance for EM data sets. Using the most advanced algorithm SMOTE to solve the class imbalance problem, and also applied to the M and E steps, the author gives a pseudo-code to refer to, as follows

Algorithm 1: SIMPLE

Input: Labeling matrix X
Output: Estimated soft labels γ

```

1  $\gamma \leftarrow$  majority vote on  $X$ 
2 while Not Converged do
3   M Step
4     Obtain hard labels  $\hat{y}$  by binarize the soft labels  $\gamma$ .
5     Make the classes balanced:  $X', \hat{y}' = \text{SMOTE}(X, \hat{y})$ 
6     Select random forest parameters  $d_{\max}$  and  $\text{ccp\_alpha}$  with cross validation on data
        $(X', \hat{y}')$ 
7      $\text{RandomForestClassifier.fit}(X', \hat{y}')$ 
8   E Step
9      $\gamma \leftarrow \text{RandomForestClassifier.predict\_proba}(X)$ 
10 end
11 return  $\gamma$ 

```

The author also discusses the complexity of the algorithm, the time complexity is $O(N \log(N))$, where the number of tuples in the candidate set and the space complexity is $O(N)$.

Here the author compares his own model with the above model, in several ways, for example, the above pseudo-code is semi-supervised, while the authors are unsupervised. Another example is that the above pseudo-code is performed on the training data, while his proposed code is performed on all data sets. It can be seen from this that the advantage of the author's algorithm is still very strong.

4 Matching method

4.1 Transitivity as a constraint on matching probability

Because for any tuple t_i, t_j, t_k , transitive constraints can be expressed as the following inequality: $\gamma^{(i,j)} \times \gamma^{(i,k)} \leq \gamma^{(j,k)}$, Bring it into the maximum expectation algorithm

$L(\theta, X, \gamma) = -\sum_{i=1}^N (\gamma_i \log(g(x_i, \theta)) + (1 - \gamma_i) \log(1 - g(x_i, \theta)))$ In the formula, we get $L(\theta, X, \gamma) =$

$$\sum_{(i,j)} -\gamma^{(i,j)} \log \frac{g(x^{(i,j)}, \theta)}{\gamma^{(i,j)}} - (1 - \gamma^{(i,j)}) \log \frac{1 - g(x^{(i,j)}, \theta)}{1 - \gamma^{(i,j)}} \dots (3)$$

In the calculation of this correlation M-step remain unchanged from the original The E-step is calculated through the projection heuristic γ , Put the E-step in $g(x^{(i,j)}, \theta)$ replace with $\gamma^{*(i,j)}$, to get the best match γ is $\gamma^{**} = \argmin_{\gamma \in Q} F(\theta, X, \gamma) \dots (4)$

The relationship between constrained solutions and unconstrained solutions:
Suppose there is a function h , Let $\gamma^{**} = h(\gamma^*)$, And γ^* depends on $\gamma^{*(i,j)} = g(x^{(i,j)}, \theta)$ to obtain, Then in step E, we can use $\gamma^{*(i,j)}$ replace $g(x^{(i,j)}, \theta)$, Bring it into equation 3, so that the objective function only has two variables $\gamma^{*(i,j)}$ and $\gamma^{(i,j)}$, And because there are constraints in equation 4, we can get the constrained solution γ^{**} relies only on unconstrained solutions γ^*

4.2 Double-meter EM transfer

(1) Constraint solution when there is no duplication in the left table

Assume that the tuples of the left table are grouped (t^l_i, t^l_j) , the probability of matching with the table on the right is non-zero, and we can get $\gamma^{**}(r_k, l_i) \gamma^{**}(r_k, l_j) \leq \gamma^{**}(l_i, l_j) = 0$, So this assumption does not hold, i.e. for any tuple in the right table t_{r_k} , There is only one tuple in the left table that has a non-zero probability of matching t_{r_k} . The matching probability of is non-zero.

Therefore, the algorithm to get γ^{**} according to γ^* is as follows: for each tuple in the right table, find the left tuple with the largest matching probability, and set the matching probability of all other left tuples to zero. Its time complexity is $O(M_l N \log(N))$, And in the case where a table is duplicate-free, it is the optimal method. This approach is suboptimal/greedy when both tables are unique.

(2) Constraint solution when both tables are non-duplicate

The reasoning in the case of single table without duplication can be extended to two-way, that is, each tuple in the left table has a non-zero matching probability for a tuple in the right table, and each tuple in the right table has a non-zero matching probability. The non-zero matching probability of a tuple in the left table. Therefore we want to keep the minimum $(|L|, |R|)$, and set everything else to 0.

This is essentially an allocation problem, and there is currently an effective algorithm to solve it - the LAPJV algorithm, whose time complexity is $O(M \min(N_l, N_r))$

(3) No repeated testing

See 8,1 appendix

4.3 Single table EM transfer

As in 4.1 the author recommends training a model offline to approximate h such that

$\gamma^{**} = h(\gamma^*)$, randomly generated γ^* , and solved γ^{**} , to train this model ml that approximates h . The normal idea is to get all possible values in the γ^* domain and solve numerically for each value, then save the result in a dictionary this way, we get the numerical representation of the function h as a dictionary. Then when reasoning, for each value γ^* , we can find the corresponding in the dictionary γ^{**} .

The ml method proposed by the author has been optimized. By fixing the input and output dimensions, get a random subset of the γ^* domain to compress the map. The authors first train a model to approximate h for fixed-dimension γ^* (e.g., a 1024-dimensional γ is enough to represent up to 32 tuple pairs, Because there are $32 \times 32 = 1024$ matching probabilities). It is then carefully decomposed γ^* into subcomponents (i.e., clusters) of maximum size 1024 and the model is applied to each subcomponent.

Randomly generate a probability matrix of data matching probability matrix with a size of 32×32 , each probability matrix corresponds to a 1024-dimensional vector γ^* . Each probability matrix corresponds to a training data point. By replacing $g(x^{(i,j)}, \theta)$ in $F(\theta, X, \gamma)$ with $\gamma^{*(i,j)}$, we get:

$$\gamma^{**} = \arg \min_{\gamma \in Q} \sum_{(i,j)} -\gamma^{(i,j)} \log \frac{\gamma^{*(i,j)}}{\gamma^{(i,j)}} - (1 - \gamma^{(i,j)}) \log \frac{1 - \gamma^{*(i,j)}}{1 - \gamma^{(i,j)}} \dots (5)$$

Express the large sum on the right as $h_1(\gamma^*, \gamma)$, so $\gamma^{**} = \arg \min_{\gamma \in Q} h_1(\gamma^*, \gamma)$, At the same time, in order to

explain the transitivity constraint Q, the author adds an additional transitivity loss to the objective function.

The total number of transitivity violations for all triples is:

$$l_{\text{transitivity}}(\gamma) = \sum_{i,j,k} \text{Relu}(\gamma^{(i,j)} \gamma^{(i,k)} - \gamma^{(j,k)}) \dots (6)$$

The constrained solution γ^{**} can be obtained by minimizing the following expression with respect to

$$\gamma: \text{Loss}(\gamma^*, \gamma) = \alpha l_{\text{transitivity}}(\gamma) + h_1(\gamma^*, \gamma) \dots (7)$$

For each γ^* , multiple different solutions for γ^{**} are obtained, where each solution comes from a different optimizer. Then we choose the solution with the smallest loss. thereby getting the result.

Output size reduced

The authors simplify the task to predicting only a single value by exploiting the symmetry of the task.

Let h represent a model that accepts an input matrix and predicts a single value in the red cell, where the original value is γ_{ij}^{**} . Let $S_{ij}^{kl}(\gamma^*)$ mean exchanging t_k and t_i and exchanging t_i and t_j . Then, $\forall i, j$, We have $\gamma_{ij}^{**} = h(S_{ij}^{kl}(\gamma^*))$. This way, we only need to train a model that predicts one value, which is much easier.

Exchange-immutable model architecture

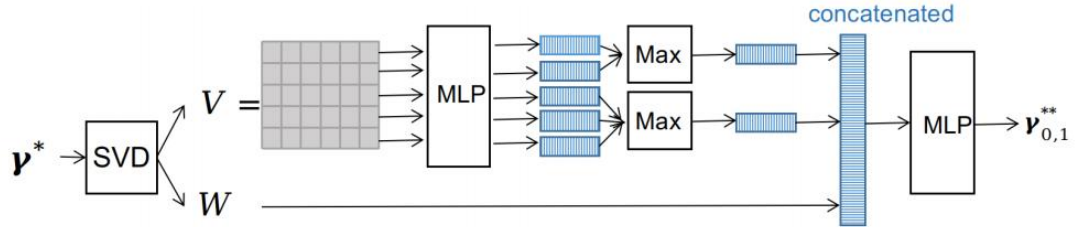
$$h(\gamma^*) = h(S_{0,1}^{1,0}(\gamma^*))$$

$$h(\gamma^*) = h(S_{ij}^{kl}(\gamma^*)), \forall i, j, k, l \text{ that } \{0,1\} \cap \{i, j, k, l\} \dots (8)$$

In order to prove that a single value is feasible, the author wants to bring constraint 8 into the calculation model to achieve exchange invariance. Let γ_{mat}^* represent γ^* in matrix form. The exchange operation $S_{ij}^{kl}(\gamma^*)$ can also be written in matrix form as $(P_i^k P_j^l) \gamma_{\text{mat}}^* (P_i^k P_j^l)^T$. Then according to the author's core idea, singular value decomposition γ is used to decompose the input matrix into an eigenvector matrix V and an eigenvalue matrix $\gamma_{\text{mat}}^* = V W V^T$ to obtain the following formula:

$$(P_i^k P_j^l) \gamma_{\text{mat}}^* (P_i^k P_j^l)^T = (P_i^k P_j^l) V W V^T (P_i^k P_j^l)^T = (P_i^k P_j^l V) W (V P_i^k P_j^l)^T = V' W V'^T \dots (9)$$

This resulted in the final architecture of a model architecture shown in the figure below:



For double-table EM, when one table is non-duplicate, the time complexity is $O(M_1 N \log(N))$; When there are no duplicates in the two tables, the time complexity is $O(M_1 N (\log(N) + \min(N_l, N_r)))$; When no table is duplicate-free, then transitivity is not used and the time complexity is $O(M_1 N \log(N))$. For single table EM, the time complexity is $O(M_1 N \log(N))$. In all cases, the space complexity is $O(N)$

5 EXPERIMENTS

First, the author proposes five methods to evaluate its model. First, the overall performance is compared with other algorithms. 2 is the difference between the algorithm combined transitivity and the previous algorithm; 3 is data migration, the performance of traditional methods and weak supervision behavior in data migration. 4 is the sensitivity of the data set to different label functions. 5 is the truth inference for general tasks, how it works in general weakly supervised algorithms. The following various experimental data will answer these questions of the author one by one.

5.1 Experimental Setup

First, the data set analyzed by the author comes from the single table and multi-table benchmark data set of the realistic data in EM's works, such as the Leipzig DB Group [2], the Magellan project [3] and the Alaska benchmark [4]. The authors will use only the 10 most common data attributes in this dataset, and fill in NA if there are missing attributes. Experiments are carried out using other most advanced algorithms and the algorithms proposed by the authors. Such as MV, D&S, EBCC and other algorithms.

5.2 Labeling Function Development

Different data labels can be developed for existing data sets, but previous data sets with the same data attributes can share data labels, which can greatly save the development time of data labels.

5.3 Overall Performance

By comparing the data results of different algorithms, the overall data performance of different algorithms is explained, and the reasons for the data performance are given. No matter from any Angle, it can be proved that the effect of the algorithm proposed by the author is much higher than the average, and there is a little gap between it and very individual data. However, the algorithm corresponding to this data does not perform well for other data and relies heavily on the characteristics of the data set, but the algorithm proposed by the author is universal.

5.4 Handling Transitivity Constraint

Experiments were conducted to compare different methods to solve transitivity, namely no transitivity, Simplex -EM, ZeroER trans and Postprocess. Among them, the one proposed by the author has the best performance after using transitivity, with an increase of 9% over F1 score.

5.5 Controlled Study of Transitivity

5.5.1 Exploration of violation transitivity in real-world data sets

First, the author proposes that there are data sets that violate transitivity in the real world. Examples are given to illustrate the performance of transitivity violation in single tables and multiple tables, and the reasons for the occurrence of transitivity violation are discussed. Only 4% of the data set is transitive, which is a small percentage, and then 40% of the violations are due to incorrect matching and 60% are due to incomplete matching of labels.

5.5.2 Change the number of transitivity violations

Through experiments, the number of transitivity is changed by variable x . The larger x is, the more the number of transitivity violations is. Then the three methods are compared. From the point of view of increasing the value of x , the overall score is decreasing, but the method proposed by the author has a higher score than the other two methods, indicating that the author's performance is optimal even if there is a match that violates transitivity.

5.6 Adaptation to Data Shift

In this section of data migration, which exists in the real world, the authors study the use of labeling functions to compare with traditional manual labeling by comparing patterns between data sets. The method proposed by the author is more convenient, even if the data is migrated, as long as the data is the

same or similar attributes, the repeatability of the label function is high, and the final effect is much higher than that of manual labeling.

5.7 Sensitivity Analysis

The author studies the sensitivity of the method proposed by the author to the label function when the label function is controlled by threshold x and the number of label functions is gradually reduced. The results show that with the reduction of the proportion of label function, the final score will indeed be reduced, but it is equivalent to other methods, and the proportion of score reduction is much smaller and better than other algorithms. It is proved that the reduction of label function will reduce the score, but only when the proportion is about 40%, the score will be significantly reduced, and other scores will not be much reduced.

5.8 Truth Inference on General Tasks

The author compared other methods from the overall data, mainly from the F1 and acc dimensions, and found that the method proposed by the author is not only simple, but also 3% higher than the average of the best baseline, which is far better than other existing methods, which have very complex models.

6 RELATED WORK

Entity matching

Although the supervised algorithm performs well in entity matching, it needs a lot of manpower to label data. Transfer learning can also be applied to existing source data or trained language models, and its robustness will deteriorate when the target data set changes, as can be seen from the experimental data. Different from the previous existing methods, for entity matching, weak supervision is adopted here, and the data label is generated programmatically, which saves a lot of manpower, and the matching effect is better. Especially when there is no annotated data, a new entity matching method is provided.

Truth Inference. Truth inference

The general truths that exist are designed for general tasks. The Simplex -EM method proposed by the author is specially designed for EM. It combines the unique transitivity of EM, and has better performance than the existing general truth value inference methods, lower time complexity, easier to understand, lower model complexity, and no need to manually label the data set based on weak supervision. Most of the existing methods need manual annotation, and the models are complex with various assumptions. In contrast, our approach requires no assumptions and is generic for all datasets, yielding better results on both general and EM tasks.

7 CONCLUSION

In this paper, a new EM matching method is proposed innovatively. By combining different LF functions to apply to weakly supervised data sets, it does not need data labeling and saves a lot of manpower. This matching method is proposed for the first time, it is simple and efficient, and this method is applied to EM, adding the unique transitivity of EM. The experimental data show that our method performs better than the existing method in 10 weakly supervised data sets from different angles by comparing the existing methods from various angles.

To solve the EM problem, the author first compared the existing methods and found the defects of the existing methods, such as the complexity of the model and the need for a lot of manpower to label the data. Based on this, a new innovative method is proposed, and the unique transitivity of EM is combined

with the credibility and dexterity of the method from a theoretical point of view, and then several groups of experimental data are compared with different existing methods and from different angles, and it is found that the existing method performs better. From this paper, we learn that in view of the shortcomings of existing methods, we constantly ask ourselves whether we can do better in what aspects and from what angles. Then I searched a lot of literature to find out the feasibility of this problem. Finally, I thought from the experimental point of view, which aspects can better demonstrate the superiority of this method, and everything was based on data and feedback conclusions from data results. This kind of thinking is what we need to learn, whether it is any topic or any difficulty, it is to think about the defects in the existing methods, and make good use of the existing theoretical basis, and constantly improve and progress little by little. The last is to learn the idea of data demonstration, everything must have the data to speak, the author a large length of experimental data, different angles to demonstrate the method of data superiority, from the perspective of data is very convincing, but also have a global thinking, the Angle should be complete, can not only from a favorable perspective to explain, so it is not credible.

REFERENCES

- [1] Renzhi Wu, Alexander Bendeck, Xu Chu, and Yeye He. 2023. Ground Truth Inference for Weakly Supervised Entity Matching. *Proc. ACM Manag. Data* 1, 1, Article 32 (May 2023), 28 pages. <https://doi.org/10.1145/3588712>
- [2] Benchmark datasets for entity resolution.
https://dbs.uni-leipzig.de/research/projects/object_matching/benchmark_datasets_for_entity_resolution.
- [3] Sanjib Das, AnHai Doan, Paul Suganthan G. C., Chaitanya Gokhale, Pradap Konda, Yash Govind, and Derek Paulsen. [n.d.]. The Magellan Data Repository.
<https://sites.google.com/site/anhaidgroup/useful-stuff/the-magellan-datarepository>
- [4] Valter Crescenzi, Andrea De Angelis, Donatella Firmani, Maurizio Mazzei, Paolo Merialdo, Federico Piai, and Divesh Srivastava. 2021. Alaska: A Flexible Benchmark for Data Integration Tasks. *arXiv preprint arXiv:2101.11259* (2021).

