Structural Constraints for Multipartite Entity Resolution with Markov Logic Network

Tengyuan Ye[*]
College of Computer Science & Technology
Zhejiang University
ytyuan1993@gmail.com

Hady W. Lauw
School of Information Systems
Singapore Management University
hadywlauw@smu.edu.sg

ABSTRACT

Multipartite entity resolution seeks to match entity mentions across several collections. An entity mention is presumed unique within a collection, and thus could match at most one entity mention in each of the other collections. In addition to domain-specific features considered in entity resolution, there are a number of domain-invariant structural contraints that apply in this scenario, including one-to-one assignment as well as cross-collection transitivity. We propose a principled solution to the multipartite entity resolution problem, building on the foundation of Markov Logic Network (MLN) that combines probabilistic graphical model and first-order logic. We describe how the domain-invariant structural constraints could be expressed appropriately in terms of Markov logic, flexibly allowing joint modeling with domain-specific features. Experiments on two real-life datasets, each spanning four collections, show the utility of this approach and validate the contributions of various MLN components.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering; H.2.8 [**Database Applications**]: Data mining

Keywords

entity resolution; markov logic network; structural constraints;

1. INTRODUCTION

Often the same entity, be it a person, document, or object, is mentioned in several separate collections, and there is a need to identify that several mentions refer to the same entity. For instance, when two health organizations merge, the medical records of patients need to be integrated. While shopping for a camera, one may wish to compare prices on online shops, such as BestBuy, Newegg, or Amazon.

[*]Work done while at Singapore Management University.

CIKM'15, October 19–23, 2015, Melbourne, Australia.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3794-6/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2806416.2806590.

This problem, formalized by [3], is known by several terms including entity resolution [2, 9], deduplication, or reference reconciliation. Most works focus on either the similarity measure between two mentions [10, 7, 6], or the algorithmic framework to match the mentions (e.g., graph matching [5]).

This paper focuses on structural constraints facing the entity resolution problem. Real-world scenarios have varying structures. In one scenario (deduplication), a single collection may contain duplicate elements to be resolved. In a different scenario (referred to as multipartite resolution), two or more collections need to be integrated into a unified collection, by resolving which entity from one collection corresponds to which entity from another collection.

Focusing on the multipartite resolution scenario, we assume that each collection is free of duplicates, because each collection belongs to a party (e.g., an online shop, or a hospital) with an interest in maintaining a high-quality collection, or because the collection has been previously cleaned up through deduplication (a related, yet distinct research problem). Thus, when integrating two collections, there is a natural constraint that a mention from one collection could match at most one mention from the other. One possible approach is graph matching [5], whereby a one-to-one assignment may be obtained from maximum weight bipartite matching. However, for three or more collections [15], the 3-dimensional matching problem is NP-hard [1].

Here, we take the approach of incorporating structural constraints into entity resolution with Markov Logic Network (MLN) [12]. An MLN-based formulation has several advantages. First, the proposed domain-invariant structural constraints can be combined with previous MLN predicates for entity resolution based on domain-specific features [14] or constraints [13]. Second, it is "extendable", as MLN is an open framework that allows joining the entity resolution task with other mutually-reinforcing tasks such as segmentation [11]. Third, MLN is probabilistic, whereby the importance of different constraints can be learned from data.

Contributions. *First*, we describe a series of domain-invariant structural constraints for MLN-based bipartite entity resolution (see Section 3), and investigate how the constraints extend to multipartite structures (see Section 4). *Second*, we validate the effectiveness of these constraints on real-life datasets of phones and cameras (see Section 5).

2. OVERVIEW

Problem Statement. As input, we are given a set of N collections $\{X_1, X_2, \ldots, X_N\}$. Each collection X_i consists of M_i entity mentions, where each mention in X_i is associated with some features. Our focus in this work is on structural constraints, and not on the feature set or similarity measure. For simplicity, and without loss of generality, for subsequent discussions we assume there is a single feature, which is the name of the entity, and we use Jaccard similarity [6].

We associate two mentions $x \in X_i$ and $y \in X_j$ from different collections with a binary variable $\texttt{Match}(x, y)$, whose value is 1 if x and y "match", i.e., referring to the same entity, and is 0 otherwise. Our objective is to determine the states of these variables for all pairs of entity mentions. If x and x' belong to the same collection X_i, trivially $\texttt{Match}(x, x') = 0$, i.e., each collection is assumed to be free of duplicates.

Markov Logic Network (MLN). Our approach is to express structural constraints for entity resolution as first-order logic formulae. Each formula may use four types of symbols: constants, variables, functions, and predicates. Constants are objects in the domain of interest, which in this case are the entity mentions. Variables, e.g., x, range over objects in the domain. Predicates represent relations among objects, e.g., $\texttt{Similar}(x, y)$ is a predicate that indicates that x and y are similar in feature, whereas $\texttt{Match}(x, y)$ represents that x and y refer to the same entity. The formulae for the structural constraints are specified in Sections 3 and 4.

Some constraints are *hard* constraints, i.e., they must be satisfied. Others are *probabilistic* constraints, i.e., solutions that satisfy them are more likely to be correct. The first-order logic formulae are weighted, with higher weight indicating higher probabilities. To model these weights in a principled manner, as well as to learn them from data, we employ Markov Logic Network (MLN) [12], which combines probabilistic graphical model and first-order logic. An MLN is a set of weighted first-order logic formulae (each formula f_t is associated with weight w_t). ϕ is a possible solution in the space of possible solutions Φ. The probability $P(\Phi = \phi)$ is expressed in Equation 1, where Z is the partition function and $f_t(\phi)$ is 1 if the formula f_t holds in ϕ and 0 otherwise.

$$P(\Phi = \phi) = \frac{1}{Z} \exp\left(\sum_t w_t f_t(\phi)\right) \tag{1}$$

In the inference stage, we use the learnt weights to estimate the probability of $\texttt{Match}(x, y)$ for every pair of entity mentions from any two different collections.

3. BIPARTITE ENTITY RESOLUTION

In this section, we begin with the first-order logic formulae for constraints in bipartite entity resolution.

Similarity Constraint. There are various features used to match entities in entity resolutions. Some are domain-specific [14], which are not our focus here. Instead, we use a domain-invariant constraint specifying that matching entities share some level of similarity. We introduce the predicate $\texttt{Similar}(x, y)$ when x and y has a similarity above a certain threshold (we try different values based on Jaccard and settle on 0.5 for experiments). The basic formula links `Similar` (which is observed) to `Match` (to be determined).

There are several possible instantiations. The first instantiation (Equation 2) indicates that similar pairs should match, which may be too stringent when polysemy is an issue, i.e., many similar entity mentions do not match in real life. The second instantiation (Equation 3) indicates that matching pairs should be similar, which is more appropriate for the electronic products datasets that we experiment with, where different products may share similar features (e.g., iPhone 5 models of different memory sizes).

$$\texttt{Similar}(x, y) \Rightarrow \texttt{Match}(x, y) \tag{2}$$

$$\texttt{Match}(x, y) \Rightarrow \texttt{Similar}(x, y) \tag{3}$$

Cardinality Constraint. Due to the assumption that each collection is duplicate-free, an entity mention $x \in X_i$ cannot match more than one mention within another collection X_j. Otherwise, the two entity mentions $y, y' \in X_j$ matched to x would effectively be themselves matched. This "at most one" rule is expressed in Equation 4, where full stop indicates that it is an inviolable hard rule. In some scenarios, we need an "at least one" rule (Equation 5), if every mention within a collection must match at least one mention in another collection. If there is a bijection or one-to-one match between two collections of the same size, we model it by using both at-most-one and at-least-one rules simultaneously.

$$\texttt{Match}(x, y) \wedge \texttt{Match}(x, y') \wedge (y = y'). \tag{4}$$

$$\forall x \in X_i,\ \exists y \in X_j,\ \texttt{Match}(x, y). \tag{5}$$

Preference Constraint. While the similarity constraint helps to ensure that matching pairs have some level of similarity, it may result in false positives. If a mention x is similar to two different mentions y and y', there should be a way for x to favor a match with the *more* similar mention. We therefore introduce another predicate $\texttt{Prefer}(x, y, y')$, which indicates that x is more similar to y than to y'.

We then link `Prefer` to `Match`. The following first-order logic formula is inspired by the Stable Marriage Problem [4], where the objective is to arrive at a matching of "men" and "women", such that no unmatched couple would rather be with each other than with their respective partners. Expressed in first-order logic, Equation 6 makes it more likely that if x is matched to y, and x' to y', we would not have the case where x prefers y' to y, and y' prefers x to x'.

$$\begin{aligned} &\texttt{Match}(x, y) \wedge \texttt{Match}(x', y') \Rightarrow \\ &!(\texttt{Prefer}(x, y', y) \wedge \texttt{Prefer}(y', x, x')) \end{aligned} \tag{6}$$

Global Constraint. The preference constraint above induces a local optimization by preferring one matching pair over another. There is no guarantee that the matching is optimized in a global way for all pairs. Complementarily, this is the strength of graph matching [5]. In graph matching, every mention is a vertex in a bipartite graph. Each edge between two mentions is weighted by the similarity value. Employing maximum weight bipartite matching obtains the matching with the highest sum of similarities. Due to the flexibility of MLN, it is feasible to be informed by this global solution. We introduce another predicate `MaxWeight`(x, y) that indicates that (x, y) is part of the maximum weight bipartite matching solution. For generality, we also model the greedy version [5], which iteratively constructs the matching solution by selecting the highest-weighted edges first, as the predicate `Greedy`(x, y). It is feasible to factor these results into the MLN formulation, as shown in Equations 7 and 8.

$$\texttt{MaxWeight}(x, y) \Rightarrow \texttt{Match}(x, y) \tag{7}$$

$$\texttt{Greedy}(x, y) \Rightarrow \texttt{Match}(x, y) \tag{8}$$

4. MULTIPARTITE ENTITY RESOLUTION

When there are $N > 2$ collections to be integrated, the problem turns into multipartite entity resolution. On one hand, the multipartite scenario can be decomposed into its $N(N-1)/2$ constituent bipartite scenarios, which enables the re-use of constraints defined in Section 3. On the other hand, modeling the problem jointly has the benefit of potentially allowing the bipartite matches to correct one another.

Cross-Collection Transitivity. The basic mechanism to join the bipartite matches is through cross-collection transitivity. Suppose $x \in X_i$ matches $y \in X_j$, and in turn y matches $z \in X_k$, it is probable that x also matches z, as expressed in Equation 9. This essentially creates two "pathways" for x to match z. One is direct, through the bipartite constraints defined in Section 3. The other is indirect, through transitivity via y. However, these two pathways may not always agree, which may result in errors.

$$\texttt{Match}(x, y) \wedge \texttt{Match}(y, z) \Rightarrow \texttt{Match}(x, z) \qquad (9)$$

To combat this, we accumulate even more evidence from a greater number of collections. For instance, for four collections involving $v \in X_l$, Equation 10 is a more efficient rule that allows accumulation of evidence by combining two cross-collection transitivities. This rule has a notion of "majority". There are now three pathways, the direct pathway to $\texttt{Match}(x, z)$ based on constraints defined in Section 3, and two indirect pathways via y, i.e., $\texttt{Match}(x, y) \wedge \texttt{Match}(y, z)$, as well as via v, i.e., $\texttt{Match}(x, v) \wedge \texttt{Match}(v, z)$. If the direct pathway is incorrect, and the two indirect pathways are correct and in agreement, the latter two (intuitively forming a "majority") could have a corrective effect on the former.

$$\begin{aligned}&\texttt{Match}(x, y) \wedge \texttt{Match}(y, z) \wedge \\ &\texttt{Match}(x, v) \wedge \texttt{Match}(v, z) \quad \Rightarrow \texttt{Match}(x, z) \qquad (10)\end{aligned}$$

Label vs. Feature Transitivity. In the above, we rely on the target label $\texttt{Match}(x, y)$ to model cross-collection transitivity. This is not ideal for MLN, because the target label is not grounded, and is to be learned. As a result, it induces too much dependency among the four collections, and may propagate errors. To break these dependencies, we propose to ground the transitivities on observed features. One way is to express these transitivities in terms of global constraints, as shown in Equation 11 (using `MaxWeight` predicate) and Equation 12 (using `Greedy` predicate).

$$\begin{aligned}&\texttt{MaxWeight}(x, y) \wedge \texttt{MaxWeight}(y, z) \wedge \\ &\texttt{MaxWeight}(x, v) \wedge \texttt{MaxWeight}(v, z) \quad \Rightarrow \texttt{Match}(x, z) \qquad (11)\end{aligned}$$

$$\begin{aligned}&\texttt{Greedy}(x, y) \wedge \texttt{Greedy}(y, z) \wedge \\ &\texttt{Greedy}(x, v) \wedge \texttt{Greedy}(v, z) \quad \Rightarrow \texttt{Match}(x, z) \qquad (12)\end{aligned}$$

Constraint	MLN								
	#1	#2	#3	#4	#5	#6	#7	#8	#9
Similarity	✓	✓	✓	✓	✓	✓	✓	✓	✓
Cardinality		✓	✓	✓	✓	✓	✓	✓	✓
Preference			✓	✓	✓	✓	✓	✓	✓
Global (MaxWeight)				✓		✓		✓	
Global (Greedy)					✓		✓		✓
Cross-Collection Transitivity (MaxWeight)						✓		✓	
Cross-Collection Transitivity (Greedy)							✓		✓
Domain-Specific (SameModel)								✓	✓

Table 1: MLN's with Various Components

5. EXPERIMENTS

The primary objective is to study the effectiveness of the MLN approach, and the contributions of various MLN rules.

Datasets. Existing datasets for entity resolution comprise at most two collections. As our interest is in the multipartite scenario, we construct two new real-life datasets comprising four collections each. *Cameras* consists of the names of 80 cameras that existed on all four Web sites studied (Overstock, Newegg, BestBuy, and Amazon). *Phones* consists of the names of 70 mobile phones obtained from the same four Web sites. We use 30 entities for training, and the rest for testing. The size of the data is restricted by the need to manually label the matching across four collections.

Comparative Methods. Our main focus is on the comparison of various MLN constraints. Table 1 lists the nine MLN's to be studied, indicating the constraints included by each MLN. For learning and inference, we build the MLN solutions on the Alchemy [8] library. In addition, we will compare to two graph matching approaches, namely maximum weight bipartite matching, as well as its greedy version.

Metric. MLN outputs the probability that two mentions match. One possible metric is the likelihood of true matches. However, to put our approach on the same footing as graph matching that outputs binary outcomes, we discretize the probabilities to 1 (matched) or 0 (not matched) by assigning each entity mention to the highest probability match (taking into account any conflict). We then measure the accuracy of the pairs according to the ground truth labels.

Bipartite Entity Resolution. First, we look at the progression of structural constraints in the bipartite scenario (see Section 3). Table 2 shows the accuracies for *Cameras*, while Table 3 shows the accuracies for *Phones*. For four collections, there are six bipartite scenarios. Each row shows accuracies for each bipartite scenario. The last row is the overall accuracies, averaging the first six rows. MLN#1 that only has similarity constraint performs poorly because the one-to-one constraint is not enforced. By adding cardinality constraint, MLN#2 improves in accuracy significantly. The greatest boost in accuracy comes from adding preference constraint as in MLN#3, because the relative ranking of similarities helps to arrive at better pairings. Adding global constraints, through factoring in the MaxWeight (MLN#4) or Greedy (MLN#5), results in only a small increment in accuracies. These show that the progression of structural constraints result in better performances across both datasets.

Multipartite Entity Resolution. In multipartite settings, it is possible to gain further accuracies by modeling all the collections jointly, rather than by decomposing them into the constituent bipartite settings. Table 4 compares accuracy of bipartite versus multipartite modeling. Only the average accuracies are shown due to space limitation. Whether MaxWeight or Greedy is used in the global and cross-collection transitivity constraints, multipartite mod-

	MLN#1	MLN#2	MLN#3	MLN#4	MLN#5
Overstock-Newegg	18.0%	18.0%	90.0%	94.0%	90.0%
Overstock-BestBuy	4.0%	10.0%	88.0%	88.0%	88.0%
Overstock-Amazon	8.0%	30.0%	92.0%	94.0%	92.0%
Newegg-BestBuy	16.0%	26.0%	78.0%	96.0%	76.0%
Newegg-Amazon	6.0%	16.0%	72.0%	70.0%	72.0%
BestBuy-Amazon	12.0%	16.0%	72.0%	72.0%	72.0%
Average	10.7%	19.3%	82.0%	85.7%	81.7%

Table 2: Bipartite Entity Resolution - Cameras

	MLN#1	MLN#2	MLN#3	MLN#4	MLN#5
Overstock-Newegg	2.5%	12.5%	100.0%	95.0%	100.0%
Overstock-BestBuy	10.0%	10.0%	82.5%	82.5%	85.0%
Overstock-Amazon	12.5%	20.0%	85.0%	90.0%	82.5%
Newegg-BestBuy	12.5%	7.5%	95.0%	95.0%	95.0%
Newegg-Amazon	7.5%	35.0%	95.0%	100.0%	100.0%
BestBuy-Amazon	10.0%	12.5%	92.5%	100.0%	92.5%
Average	9.2%	16.3%	91.7%	93.8%	92.5%

Table 3: Bipartite Entity Resolution - Phones

elling helps to improve the accuracies, e.g., from 85.7% to 90.0% for *Cameras* and from 93.8% to 96.7% for *Phones*, in the case of MaxWeight-based constraints.

Comparison to Graph Matching. Our focus is on MLN-based approaches. Previously, when MaxWeight and Greedy are used, they are factored into the MLNs. For completeness, we include a comparison to graph matching on its own. For bipartite matching, we use MaxWeight (MW2) and Greedy (G2) directly. For multipartite matching (MaxWeight is NP-hard), we rely on Greedy for three (G3) and four (G4) sources respectively, as defined in [15]. Table 5 shows that MLN#6 and MLN#7 are quite similar to (in some cases slightly better than) graph matching. Importantly, MLN has not resulted in a drop in accuracy, while providing benefits such as extendability and probabilities.

Extendability via Domain-Specific Feature. Our focus here is on domain-invariant rules. However, to illustrate the potential extendability of MLN, we show how the addition of one new domain-specific rule may improve the accuracies. The rule is $\texttt{SameModel}(x, y) \Rightarrow \texttt{Match}(x, y)$. It uses a new predicate `SameModel`, specific to the domain of consumer electronics (cameras, phones), indicating whether two mentions share similar model and brand. Table 6 shows that by adding this rule, MLN#8 outperforms MLN#6, and similarly MLN#9 outperforms MLN#7. This extendability shows the value of MLN-based approach, which allows additional rules, constraints, or even tasks to be modeled jointly.

6. CONCLUSION

We explore a framework for expressing structural constraints for multipartite entity resolution using MLN. Experiments on real-life datasets indicate the promise of this approach, showing how the various structural constraints contribute to the overall effectiveness. As future work, we plan to investigate further how factoring in domain-specific features and constraints, in addition to the proposed domain-invariant structural constraints, can result in more flexible and yet more accurate entity resolution through MLN.

	with MaxWeight		with Greedy	
	Bipartite MLN#4	Multipartite MLN#6	Bipartite MLN#5	Multipartite MLN#7
Cameras	85.7%	90.0%	81.7%	82.0%
Phones	93.8%	96.7%	92.5%	93.3%

Table 4: Multipartite Entity Resolution

	with MaxWeight		with Greedy			
	MLN#6	MW2	MLN#7	G2	G3	G4
Cameras	90.0%	87.3%	82.0%	81.7%	81.5%	85.0%
Phones	96.7%	95.8%	93.3%	92.5%	91.9%	92.5%

Table 5: Comparison to Graph Matching

	with MaxWeight		with Greedy	
	without MLN#6	with domain-specific MLN#8	without MLN#7	with domain-specific MLN#9
Cameras	90.0%	90.7%	82.0%	83.0%
Phones	96.7%	98.3%	93.3%	93.3%

Table 6: Adding Domain-Specific Feature

7. REFERENCES

[1] Y. Crama, A. G. Oerlemans, and F. C. Spieksma. *Approximation algorithms for three-dimensional assignment problems with triangle inequalities.* Springer, 1996.

[2] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *TKDE*, 2007.

[3] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *JASA*, 1969.

[4] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *American mathematical monthly*, 1962.

[5] J. Gemmell, B. I. P. Rubinstein, and A. K. Chandra. Improving entity resolution with global constraints. *arXiv preprint arXiv:1108.6016*, 2011.

[6] P. Jaccard. *Etude comparative de la distribution florale dans une portion des Alpes et du Jura.* Impr. Corbaz, 1901.

[7] M.A. Jaro. Probabilistic linkage of large public health data files. *Statistics in Medicine*, 1995.

[8] S. Kok, P. Singla, M. Richardson, P. Domingos, M. Sumner, H. Poon, and D. Lowd. The Alchemy system for statistical relational AI. *University of Washington, Seattle*, 2005.

[9] H. Köpcke and E. Rahm. Frameworks for entity matching: A comparison. *DKE*, 2010.

[10] A. E. Monge and C. Elkan. An efficient domain-independent algorithm for detecting approximately duplicate database records. In *SIGMOD Workshop on Data Mining and Knowledge Discovery*, 1997.

[11] H. Poon and P. Domingos. Joint inference in information extraction. In *AAAI*, 2007.

[12] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 2006.

[13] W. Shen, X. Li, and A. Doan. Constraint-based entity matching. In *AAAI*, 2005.

[14] P. Singla and P. Domingos. Entity resolution with markov logic. In *ICDM*, 2006.

[15] D. Zhang, B. I. P. Rubinstein, and J. Gemmell. Principled graph matching algorithms for integrating multiple data sources. *TKDE*, preprint, 2015.