Expressive Power of Entity-Linking Frameworks

Douglas Burdick[1], Ronald Fagin[2], Phokion G. Kolaitis[*3], Lucian Popa[4], and Wang-Chiew Tan[5]

1 IBM Almaden Research Center, San Jose, CA, USA
2 IBM Almaden Research Center, San Jose, CA, USA
3 University of California Santa Cruz, Santa Cruz, CA, USA; and IBM Almaden Research Center, San Jose, CA, USA
4 IBM Almaden Research Center, San Jose, CA, USA
5 Recruit Institute of Technology, Mountain View, CA, USA; and University of California Santa Cruz, Santa Cruz, CA, USA

Abstract

We develop a unifying approach to declarative entity linking by introducing the notion of an entity linking framework and an accompanying notion of the certain links in such a framework. In an entity linking framework, logic-based constraints are used to express properties of the desired link relations in terms of source relations and, possibly, in terms of other link relations. The definition of the certain links in such a framework makes use of weighted repairs and consistent answers in inconsistent databases. We demonstrate the modeling capabilities of this approach by showing that numerous concrete entity linking scenarios can be cast as such entity linking frameworks for suitable choices of constraints and weights. By using the certain links as a measure of expressive power, we investigate the relative expressive power of several entity linking frameworks and obtain sharp comparisons. In particular, we show that we gain expressive power if we allow constraints that capture non-recursive collective entity resolution, where link relations may depend on other link relations (and not just on source relations). Moreover, we show that an increase in expressive power also takes place when we allow constraints that incorporate preferences as an additional mechanism for expressing "goodness" of links.

1998 ACM Subject Classification H.2 Database Management

Keywords and phrases Entity linking, entity resolution, constraints, repairs, certain links

Digital Object Identifier 10.4230/LIPIcs.ICDT.2017.10

1 Introduction and Summary of Results

Entity linking is the problem of creating links among records representing real-world entities that are related in certain ways. As an important special case, it includes *entity resolution*, which is the problem of identifying or linking "duplicate" entities. Since the pioneering work of Fellegi and Sunter [11] in 1969, entity linking has been recognized as a fundamental computational problem that has been investigated by several different research communities. While much of the work in this area [8, 10, 15, 18] has focused and continues to focus on the design, implementation, and validation of direct algorithms for entity linking (and, in particular, for entity resolution), recent investigations have developed declarative approaches to entity linking that make it possible to separate the specification of entity linking from its actual implementation (see, for example, [1, 7, 13, 14]).

[*] Part of this work was done while Phokion G. Kolaitis was visiting the Simons Institute for the Theory of Computing.

20th International Conference on Database Theory (ICDT 2017).
Editors: Michael Benedikt and Giorgio Orsi; Article No. 10; pp. 10:1–10:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In [7], we introduced and explored a declarative approach to entity linking that makes use of logical constraints. Our approach differs from earlier declarative approaches because it uses *link-to-source* constraints, instead of *source-to-link* constraints. Source-to-link constraints constitute, in effect, rules for creating links from source data in an operational manner. Our link-to-source constraints spell out conditions that the links must satisfy, independently of how the links will be created, and thus give rise to *solutions* of the declarative entity-linking specification at hand. In [7], we focused on the class of *maximum-value* solutions as "good" solutions for entity linking; intuitively, these are the solutions in which links have maximum "justification" in terms of the constraints and in terms of the source data. Since there can be multiple maximum-value solutions, we introduced the notion of the *certain links*, which, by definition, are the links that appear in every maximum-value solution and, therefore, are the links that should be kept. We then explored the problem of enumerating all maximum-value solutions and the problem of computing the certain links. This investigation was carried out for several different languages expressing link-to-source constraints, including languages that capture *collective entity resolution*, where interdependence between link relations is allowed.

The variety and multitude of entity-linking approaches raise the question of developing methods and tools for comparing such different approaches. A comparative evaluation of the performance of several different direct algorithms for entity resolution (or entity matching) has been carried in [16] and [17]. Up to now, however, no methodology has been developed for comparing, along some axis, different declarative approaches for entity linking. The main aim of this paper is to develop such a methodology that is centered on the notion of the *expressive power* of declarative entity-linking frameworks.

Our first conceptual contribution is to formulate a unifying notion of an *entity-linking framework* and an accompanying notion of the *certain links* in such a framework. This is achieved by bringing into the picture a notion of *weighted repairs* of inconsistent databases, which are a variant of the notion of weighted repairs of inconsistent databases in description logics studied in [9]. The "good" solutions for entity linking are then identified with the maximum weight repairs of inconsistent databases with respect to suitable choices of constraints and weights, while the certain links are defined to be the *consistent answers* of atomic link queries with respect to the maximum weight repairs, that is, those links that are in every maximum weight repair. The inconsistent database whose weighted repairs we consider gives an upper bound or a domain for the candidate links; it could be provided (e.g., handed in from another system), or could be simply based on the Cartesian product of sets of entities (which we do in many of our definitions and proofs[1]).

This general approach gives rise to a single formalism for declarative entity linking in which the constraint language, the sets of constraints allowed, and the weight function that measures the "strength" of the links are parameters of the definition. We demonstrate the modeling capabilities of this formalism by showing that it not only contains as special cases the concrete declarative entity linking scenarios studied in [7], but also can account for new ones, such as entity linking based on maximum cardinality repairs and entity linking with constraints that incorporate preferences.

Our second conceptual contribution is to use the certain links as a measure of the expressive power of an entity linking framework and define what it means for an entity linking framework to *subsume* another entity linking framework. This makes it possible to compare different entity linking frameworks along the axis of their expressive power.

[1] Note that this is conceptual and it does not mean that such a Cartesian product needs to be materialized.

As regards technical results, we first show that, under some mild hypotheses on entity linking frameworks, it is possible to enumerate with polynomial delay all maximum weight repairs and to compute the certain links in polynomial time. This general result contains as special cases several similar results for concrete entity linking scenarios obtained in [7]. Our main technical contribution, however, is to delineate the relative expressive power of different linking frameworks. Specifically, we show that the entity linking framework of the maximum-value solutions considered in [7] and the entity linking framework of maximum cardinality repairs introduced here are of incomparable expressive power, in the sense that neither of the two can subsume the other. We also show that the entity linking framework for collective entity resolution where the constraints allow the link relations to depend on other link relations is strictly more expressive than in the case where constraints do not allow for interdependence among the link relations. This increase in expressive power takes place even when the dependencies among the link relations are non-recursive. Finally, we show that we also gain expressive power by adding preference constraints, which represent an additional, practical mechanism (see HIL [14]) for specifying the "good" links by letting a user explicitly, and declaratively, give priority to some types of links over other types of links. Concretely, we show that there is an entity linking framework with preference constraints that is not subsumed by the entity linking framework of maximum-value solutions (with no preference constraints).

Note that since the expressive power is measured via the certain links, proving that a specific entity linking framework is not subsumed by some other specific entity linking framework is a much more challenging task than simply showing that the constraints defining the first framework are not logically equivalent to those defining the second framework. The proofs of our results about the expressive power of entity linking frameworks involve a combination of special-purpose techniques with techniques from finite model theory. In particular, the proof of the result concerning the expressive power of entity linking frameworks with preference constraints uses a locality theorem that is interesting in its own right.

In summary, the conceptual and technical contributions in this paper provide a unifying approach to declarative entity linking and pave the way for the systematic comparative evaluation of different entity linking frameworks.

2 Weighted Repairs and Consistent Answers

Let $\mathbf{S}$ and $\mathbf{L}$ be two disjoint relational schemas, and let $\mathbf{R} = \mathbf{S} \cup \mathbf{L}$ be the union of these two schemas. If I is an $\mathbf{S}$-instance and J is an $\mathbf{L}$-instance, then $\langle I, J\rangle$ denotes the $\mathbf{R}$-instance that is the union of I and J viewed as sets of facts. Clearly, every $\mathbf{R}$-instance is of the form $\langle I, J\rangle$, where I is an $\mathbf{S}$-instance and J is an $\mathbf{L}$-instance. If I is an $\mathbf{S}$-instance and S is a relation symbol in $\mathbf{S}$, then S^I denotes the relation of I interpreting the relation symbol S; similarly, if J is an $\mathbf{L}$-instance and L is a symbol in $\mathbf{L}$, then L^J denotes the relation of J interpreting the relation symbol L.

▶ **Definition 1.** A *weight function on* $\mathbf{R}$ is a function w that assigns a non-negative weight $w(\langle I, J\rangle, L^J(a_1, \ldots, a_n))$ for every $\mathbf{R}$-instance $\langle I, J\rangle$ and for every fact $L^J(a_1, \ldots, a_n)$ of J, where L is a relation symbol in $\mathbf{L}$. The weight $w(\langle I, J\rangle, L^J(a_1, \ldots, a_n))$ is called the *weight* of the fact $L^J(a_1, \ldots, a_n)$ in $\langle I, J\rangle$.

Note that, even though only facts in relations interpreting $\mathbf{L}$-symbols have weights, the weight of such a fact may depend on the entire $\mathbf{R}$-instance $\langle I, J\rangle$ and not just on J.

In what follows, we will define the notion of a *maximum weight repair* of an $\mathbf{R}$-instance $\langle I, J\rangle$; this notion is inspired by a similar one introduced by Du, Qi, and Shen [9] in the context of knowledge-bases with constraints expressed in description logics.

▶ **Definition 2.** Let Σ be a set of integrity constraints on $\mathbf{R}$, let w be a weight function on $\mathbf{R}$, and let $\langle I, J\rangle$ be an $\mathbf{R}$-instance. A sub-instance $\langle I, J'\rangle$ of $\langle I, J\rangle$ is a *maximum weight repair of* $\langle I, J\rangle$ *with respect to* Σ *and* w if $\langle I, J'\rangle$ has the following properties:

1. $\langle I, J'\rangle$ is consistent, i.e., $\langle I, J'\rangle$ satisfies every constraint in Σ.
2. J' has maximum weight, i.e., if $\langle I, J''\rangle$ is a consistent sub-instance of $\langle I, J\rangle$, then $\Sigma_{f\in J''} w(\langle I, J''\rangle, f) \leq \Sigma_{f\in J'} w(\langle I, J'\rangle, f)$.

In general, the weight function w may also depend on the set Σ of constraints at hand. If Σ and w are understood from the context, then we will simply talk about maximum weight repairs of $\langle I, J\rangle$, instead of maximum weight repairs of $\langle I, J\rangle$ with respect to Σ and w.

Thus, a maximum weight repair of $\langle I, J\rangle$ is a consistent sub-instance $\langle I, J'\rangle$ of $\langle I, J\rangle$ whose total sum of the weights of its $\mathbf{L}$-facts is maximum across all consistent sub-instances $\langle I, J''\rangle$ of $\langle I, J\rangle$. Note that the notion of maximum weight repairs introduced in Definition 2 differs from the standard notion of subset repairs [2] in two ways: first, in the standard notion, the repair takes place with respect to the entire schema or, more precisely, we have there that $\mathbf{S} = \emptyset$ and $\mathbf{R} = \mathbf{L}$; second, in the standard notion, there is no weight function on the facts. Note also that *maximum cardinality subset repairs* [21] are the special case of maximum weight repairs in which $\mathbf{S} = \emptyset$, $\mathbf{R} = \mathbf{L}$, and the weight function assigns weight 1 to each fact. Finally, note that our notion of maximum weight repairs differs also from the notion of maximum weight repairs introduced in [9] in the following way. In [9], the weight of each fact f depends on the inconsistent instance $\langle I, J\rangle$ under consideration, but remains the same on all consistent sub-instances of $\langle I, J\rangle$ containing f. In contrast, in Definition 2, the weight of each fact f may differ from instance to instance; thus, we may have $w(\langle I, J\rangle, f) \neq w(\langle I, J'\rangle, f)$, where $\langle I, J'\rangle$ is a consistent sub-instance of $\langle I, J\rangle$.

Maximum weight repairs give rise to a notion of consistent answers of queries in exactly the same way subset repairs do.

▶ **Definition 3.** Let Σ be a set of integrity constraints on $\mathbf{R}$ and let w be a weight function on $\mathbf{R}$. If q is a query on $\mathbf{R}$, and $\langle I, J\rangle$ is an $\mathbf{R}$-instance, then a tuple $\mathbf{a}$ is a *consistent answer of* q *on* $\langle I, J\rangle$ *with respect to* Σ *and* w if $\mathbf{a} \in q(\langle I, J'\rangle)$, for every maximum weight repair $\langle I, J'\rangle$ of $\langle I, J\rangle$ with respect to Σ and w.

3 Certain Links and Entity-Linking Frameworks

Here, we will focus on maximum weight repairs in declarative scenarios for entity linking, such as the ones considered in [7]. In such scenarios, $\mathbf{S}$ is the schema of *source* relations, while $\mathbf{L}$ is the schema of *link* relations, where each link relation is binary. Relation symbols in $\mathbf{S}$ will be referred to as *source* symbols, while relation symbols in $\mathbf{L}$ will be referred to as *link* symbols. Some source symbols may be interpreted by *built-in* relations, that is, such symbols may have the same interpretation on every allowable source instance. For example, a source symbol may stand for the substring relation between two strings, or it may stand for a user-defined predicate, such as similarity of names. If J is an $\mathbf{L}$-instance and $(a, b) \in L^J$ for some link symbol L in $\mathbf{L}$, then we say that (a, b) is a *link* of L in J. We sometimes write such a link as the fact $L^J(a, b)$, or $L(a, b)$ when J is clear from the context. We may also refer to J as a *link instance.*

▶ **Definition 4.** Let $\mathbf{S}$ be a schema of source symbols, let $\mathbf{L}$ be a schema of link symbols, let Σ be a set of integrity constraints on $\mathbf{R} = \mathbf{S} \cup \mathbf{L}$, and let w be a weight function on $\mathbf{R} = \mathbf{S} \cup \mathbf{L}$. If L is a link symbol in $\mathbf{L}$ and $\langle I, J\rangle$ is an $\mathbf{R}$-instance, then a *certain link of* L *on* $\langle I, J\rangle$ *with respect to* Σ *and* w is a consistent answer of the atomic query $L(x, y)$ on $\langle I, J\rangle$

with respect to Σ and w, i.e., a pair (a, b) such that $(a, b) \in L^{J'}$, for every maximum weight repair $\langle I, J' \rangle$ of $\langle I, J \rangle$ with respect to Σ and w.

We will also use the notation $L(a, b)$ for a certain link (a, b) of L. It will be clear from the context if $L(a, b)$ refers to a certain link or to a link $L^J(a, b)$ for some instance J.

Intuitively, in the above definition, we are given an instance $\langle I, J \rangle$, not necessarily consistent with respect to the set Σ of integrity constraints, where J represents an initial set of link facts. Then, the certain links of L on $\langle I, J \rangle$ represent precisely the subset of L-facts of J that appear in every maximum weight repair of $\langle I, J \rangle$. In this paper, we focus on links that are certain because it is a standard semantics in information integration, including data exchange and incomplete databases. While other alternatives may be considered (e.g., possible links, which are the links that appear in at least one maximum weight repair), we leave such investigation for future work.

Note that Definition 4 is very general and does not make any assumptions about the class of integrity constraints that is allowed in Σ or about the weight function w. We also note that the weight function w is assumed to be defined over instances of $\mathbf{R} = \mathbf{S} \cup \mathbf{L}$, independently of whether these instances are consistent with Σ or not.

The concrete choices for Σ and w will be incorporated into the notion of *entity-linking frameworks*, which we define next, together with the notion of *entity-linking specifications*.

▶ **Definition 5.** Let $\mathbf{S}$ be a schema of source symbols, let $\mathbf{L}$ be a schema of link symbols, and let $\mathbf{R} = \mathbf{S} \cup \mathbf{L}$.

- An *entity-linking framework on* $\mathbf{R}$ is a triple $(\mathcal{L}, \mathcal{S}, \mathcal{W})$ consisting of a logical language $\mathcal{L}$ on $\mathbf{R}$, a collection $\mathcal{S}$ of finite sets of $\mathcal{L}$-formulas, and a collection $\mathcal{W}$ of weight functions such that, for each $\Sigma \in \mathcal{S}$, there is a weight function w_Σ on $\mathbf{R}$.
- If Σ is a member of $\mathcal{S}$ and w_Σ is the associated weight function in $\mathcal{W}$, then we say that the triple $(\mathcal{L}, \Sigma, w_\Sigma)$ is an *entity-linking specification* in the entity-linking framework $(\mathcal{L}, \mathcal{S}, \mathcal{W})$.

Several different logical languages for expressing entity-linking specifications were introduced in [7] and then used to define and study different scenarios for declarative entity linking. Here, we show that all but one of the scenarios considered in [7] (namely, the scenario of maximal solutions) are concrete instances of the notion of an entity-linking framework in Definition 5, by choosing, in each case, the logical language $\mathcal{L}$, the collection $\mathcal{S}$ of finite sets of constraints from $\mathcal{L}$, and the collection $\mathcal{W}$ of weight functions. As we shall see, the weight functions can become progressively more sophisticated. Furthermore, the logical language $\mathcal{L}$ together with the collection $\mathcal{S}$ can become progressively richer.

We first focus on the language $\mathcal{L}_\mathbf{0}$ introduced in [7], consisting of three types of constraints:

- Inclusion dependencies of the form $L[X] \subseteq S[A]$ and $L[Y] \subseteq T[B]$, where L is a link symbol, and S and T are source symbols. We use X and Y to denote the first and the second attribute of L, while A and B denote attributes in relations S and T, respectively. Note that S and T could be the same source symbol.
- Functional dependencies (FDs) $L : X \to Y$ and $L : Y \to X$, where L is a link symbol and X and Y denote the attributes of L.
- Matching constraints of the form:

$$L(x, y) \to \forall \mathbf{u}(\psi(x, y, \mathbf{u}) \to \alpha_1 \vee \ldots \vee \alpha_k), \tag{1}$$

where L is a link symbol, $\psi(x, y, \mathbf{u})$ is a (possibly empty) conjunction of atomic formulas over $\mathbf{S}$ (with the requirement that the universally quantified variables $\mathbf{u}$ must occur in

ψ), and where α_i is of the form $\exists\mathbf{z_i}\ \phi_i(x, y, \mathbf{u}, \mathbf{z_i})$. Each ϕ_i is a conjunction of atomic formulas[2] over $\mathbf{S}$ along with equalities. We assume that the variables in $\mathbf{z_i}$ are disjoint from the variables in ψ and from $\{x, y\}$. Also, note that x and y are universally quantified, but for simplicity of notation we omit their quantifiers.

The intuition behind the use of disjunction in a matching constraint is that it lists all the possible matching conditions $\alpha_1, \ldots, \alpha_k$ for why a link $L(x, y)$ may exist (provided ψ holds). If a link $L(x, y)$ exists, then one or more of those conditions must be true. We do not require a matching constraint to be given for each link; for those links without a matching constraint, the link relation is implicitly defined by the rest of the constraints.

The inclusion dependencies have the important role of specifying the domain of values that can be used to populate a link relation. While in general there could be more than two inclusion dependencies for each link, all the scenarios considered in [7] focused on the case of exactly two inclusion dependencies and also on the case of exactly one matching constraint per link symbol. The next definition captures these requirements by introducing the collection $\mathcal{S}_0$; it also introduces an initial instance $\langle I, I^* \rangle$ that will be used repeatedly in the sequel (intuitively, as a superset for the repairs).

▶ **Definition 6.** Let $\mathbf{S}$ be a schema of source symbols and let $\mathbf{L}$ be a schema of link symbols.

- We write $\mathcal{S}_0$ to denote the collection of all finite sets Σ of $\mathcal{L}_\mathbf{0}$-formulas such that for each link symbol L, the set Σ contains one inclusion dependency on L for each of its attributes, contains zero, one or both functional dependencies on L, and at most one matching constraint on L.
- If I is an $\mathbf{S}$-instance, then we write I^* to denote the $\mathbf{L}$-instance defined as follows: for each link symbol L in $\mathbf{S}$, we have that $L^{I^*} = \pi_A(S^I) \times \pi_B(T^I)$, where A is the attribute of the source symbol S and B is the attribute of the target symbol T for which $\mathcal{L}_\mathbf{0}$ contains the inclusion dependencies $L[X] \subseteq S[A]$ and $L[Y] \subseteq T[B]$.

In the above definition, the instance $\langle I, I^* \rangle$ satisfies the inclusion dependencies of $\mathcal{L}_\mathbf{0}$ on each link symbol, but it need not satisfy the functional dependencies or the matching constraints of $\mathcal{L}_\mathbf{0}$.

While other combinations of constraints may also be meaningful (e.g., more than two inclusion dependencies per link, as mentioned above, or more than one matching constraint per link), the collection $\mathcal{S}_0$ is one of the simplest; it also has a good practical motivation, since it corresponds to entity linking statements in the HIL language [14].

We next give a concrete example taken from [7] of a set Σ of constraints in $\mathcal{S}_0$. We will make use of this example in the sequel.

▶ **Example 7.** In this scenario, we link subsidiaries in one database with parent companies in another database. Consider the following source schema $\mathbf{S}$:

$$\begin{array}{ll} \texttt{Subsid}(\mathit{sid}, \mathit{sname}, \mathit{location}) & \texttt{Company}(\mathit{cid}, \mathit{cname}, \mathit{hdqrt}) \\ & \texttt{Exec}(\mathit{eid}, \mathit{cid}, \mathit{name}, \mathit{title}) \end{array}$$

This source schema includes the relation symbols Subsid from the first database, and Company and Exec from the second database. The link schema $\mathbf{L}$ consists of a single link relation $L(\mathit{sid}, \mathit{cid})$. The following set Σ of constraints can be used to specify declaratively the properties of the link relation in terms of the source relations. First, Σ contains two inclusion

[2] Note that some of these atomic formulas may involve built-in relations.

dependencies $L[\mathit{sid}] \subseteq \texttt{Subsid}[\mathit{sid}]$, $L[\mathit{cid}] \subseteq \texttt{Company}[\mathit{cid}]$, and the functional dependency $L : \mathit{sid} \to \mathit{cid}$. While the inclusion dependencies specify where L is allowed to take values from, the functional dependency gives the additional requirement that the links must be many-to-one from *sid* to *cid* (i.e., every subsidiary must link to at most one parent company). Additionally, Σ includes the matching constraint:

$$\begin{aligned} L(\mathit{sid}, \mathit{cid}) \to \forall sn, loc, cn, hd\ (&\texttt{Subsid}(\mathit{sid}, sn, loc) \wedge \texttt{Company}(\mathit{cid}, cn, hd) \\ &\to (sn \sim cn) \\ &\quad \vee \\ &\quad \exists e, n, t\, (\texttt{Exec}(e, \mathit{cid}, n, t) \wedge \mathit{contains}(t, sn))\,), \end{aligned}$$

which lists all possible reasons as to why a link may exist. Concretely, if a subsidiary id (*sid*) and a company id (*cid*) are linked, then for every binding of `Subsid` and `Company` source tuples where *sid* and *cid* respectively occur, it must be that one of the two matching conditions holds: (1) there is a similarity in the names, as specified by $sn \sim cn$, or (2) there is some executive working for the company and this executive has a title that contains the subsidiary's name.

3.1 Concrete Entity-Linking Frameworks Based on $\mathcal{L}_0$

We are now in a position to define several concrete entity-linking frameworks by instantiating the general concepts introduced above. We first consider three different entity-linking frameworks obtained from $\mathcal{L}_0$ and $\mathcal{S}_0$ by using three different types of weight functions.

▶ **Framework 8.** *The entity-linking framework* $(\mathcal{L}_0, \mathcal{S}_0, \mathcal{W}_\mathbf{1})$ *of maximum cardinality repairs.*

Let $\mathbf{1}$ *be the weight function on* $\mathbf{R}$ *such that* $\mathbf{1}(\langle I, J\rangle, L^J(a, b)) = 1$*, for every* $\mathbf{R}$*-instance* $\langle I, J\rangle$ *and every fact* $L^J(a, b)$*. Consider the entity-linking framework* $(\mathcal{L}_0, \mathcal{S}_0, \mathcal{W}_\mathbf{1})$*, where, for each* $\Sigma \in \mathcal{S}$*, we have that* $w_\Sigma = \mathbf{1}$*.*

A maximum weight repair of $\langle I, I^*\rangle$ *with respect to* Σ *and* $\mathbf{1}$ *is a repair that maximizes the total cardinality of the link facts. We call such repairs* maximum cardinality repairs*.*

It can be verified that if $\langle I, J\rangle$ is such a maximum cardinality repair of $\langle I, I^*\rangle$, then J is a maximal solution for I, as defined in [7]. The converse, however, does not always hold. Like maximal solutions, the notion of maximum cardinality repairs suffers from the deficiency that they give rise to “too few" certain links. This can be seen in the following example from [7].

▶ **Example 9.** Assume the same schemas and constraints as in Example 7. A source instance I for $\mathbf{S}$ is given below as a set of facts:

$\texttt{Subsid}(s_1,$ “Citibank N.A.”, “New York”$)$	$\texttt{Company}(c_1,$ “Citigroup Inc”, “New York”$)$
$\texttt{Subsid}(s_2,$ “CIT Bank”, “Salt Lake City”$)$	$\texttt{Company}(c_2,$ “CIT Group Inc”, “New York”$)$
	$\texttt{Exec}(e_1, c_1,$ “E. McQuade”, “CEO, Citibank N.A.”$)$

In the above, ‘Citigroup Inc” and “CIT Group Inc” are two different parent companies, and “Citibank N.A.” is the name of a true subsidiary of “Citigroup Inc”, while “CIT Bank” is the name of a true subsidiary of “CIT Group Inc”. The goal of entity linking is to identify links such as $L(s_1, c_1)$ and $L(s_2, c_2)$.

It can be seen that, given our set Σ of constraints, there are exactly four maximum cardinality repairs for $\langle I, I^*\rangle$, namely $\langle I, J_i\rangle$, $i = 1, 4$, where the J_i's are as follows:

$$J_1 = \{L(s_1, c_1), L(s_2, c_1)\} \qquad J_2 = \{L(s_1, c_1), L(s_2, c_2)\}$$
$$J_3 = \{L(s_1, c_2), L(s_2, c_1)\} \qquad J_4 = \{L(s_1, c_2), L(s_2, c_2)\}$$

It is assumed here that the name similarity predicate $\sim$ evaluates to true for all pairs of subsidiary name and company name occurring in our instance I (thus, "Citibank N.A." $\sim$ "Citigroup Inc" but also "Citibank N.A." $\sim$ "CIT Group Inc", and so on).

It follows that the set of certain links of L on $\langle I, I^* \rangle$ w.r.t. Σ and $\mathbf{1}$ is empty: there is no link that appears in all four maximum cardinality repairs and, hence, no link qualifies as a certain link. However, some links are clearly stronger than others. In particular, the link $L(s_1, c_1)$ relating "Citibank N.A." to "Citigroup Inc." satisfies both the $\sim$ predicate and the `Exec`-based matching constraint, while the other links satisfy only the $\sim$ predicate. Intuitively, there is evidence that suggests that $L(s_1, c_1)$ is a strong link that should be differentiated from the other links. However, the constant weight function $\mathbf{1}$ does not provide such differentiation.

The above example illustrates the need for more refined notions of weights on links.

▶ **Framework 10.** *The entity-linking framework $(\mathcal{L}_0, \mathcal{S}_0, \mathcal{V}_0)$ of maximum-value solutions.*

For each $\Sigma \in \mathcal{S}_0$, consider the following weight function w_Σ. Given an $\mathbf{R}$-instance $\langle I, J \rangle$ and a fact $L^J(a, b)$, we distinguish the following cases:

1. *If $L(a, b)$ does not satisfy the inclusion dependencies, then $w_\Sigma(\langle I, J \rangle, L^J(a, b)) = 0$. Otherwise:*
   - **(a)** *If Σ contains no matching constraint for L, then $w_\Sigma(\langle I, J \rangle, L^J(a, b)) = 1$.*
   - **(b)** *If Σ contains a matching constraint for L (which, by the definition of $\mathcal{S}_0$, is the only such matching constraint) and if (a, b) does not satisfy the right-hand side of the matching constraint for L, then $w_\Sigma(\langle I, J \rangle, L^J(a, b)) = 0$.*
   - **(c)** *If Σ contains a matching constraint for L and if (a, b) satisfies the right-hand side of the matching constraint for L, then $w_\Sigma(\langle I, J \rangle, L^J(a, b)) = Val(L^J(a, b))$, as defined in Section 5.2 of [7]. The precise definition of Val is as follows.*

     First, let us recall that the matching constraint for L has the form (1). Assume that there is no instantiation $\mathbf{u_0}$ of the vector of universally quantified variables $\mathbf{u}$ such that $I \models \psi(a, b, \mathbf{u}_0)$. This means that the matching constraint for $L(a, b)$ is satisfied for vacuous reasons. As in the earlier case of no matching constraint, we take the value of the link to be 1. In all other cases, we let the value of the link be:

     $$Val(L^J(a, b)) = \min_{\mathbf{u}_0}\Big(\sum_{\alpha_i, \mathbf{z}_0} 1\Big). \tag{2}$$

     In the above, $\mathbf{u}_0$ ranges over all the distinct instantiations of the vector of universally quantified variables $\mathbf{u}$ such that $I \models \psi(a, b, \mathbf{u}_0)$. We take the minimum, over all such $\mathbf{u}_0$, of the strength *with which the source instance I satisfies the disjunction $\alpha_1 \vee \ldots \vee \alpha_k$. This strength is defined as a sum that gives a value of 1 for* every *distinct combination of a disjunct α_i such that I satisfies $\alpha_i(a, b, \mathbf{u}_0)$, and distinct instantiation $\mathbf{z}_0$ of the vector $\mathbf{z}$ of existentially quantified variables of α_i that makes the satisfaction of α_i hold. (Recall that α_i is, in general, of the form $\exists \mathbf{z}\ \phi_i(x, y, \mathbf{u}, \mathbf{z})$.) In the case when α_i is satisfied and the existentially quantified variables are missing, then we count only 1.*

     We can see that, intuitively, the sum in formula (2) calculates the strength of a link by counting the number of satisfied disjuncts together with the evidence (i.e., the number of existential witnesses). Taking the minimum guarantees that we take the weakest strength among all $\mathbf{u}_0$.

We remark that the weights $w_\Sigma(\langle I, J \rangle, L^J(a, b))$ do not actually depend on J.

If we revisit the earlier Example 9, we have that $\text{Val}(L^{J_1}(s_1, c_1)) = 2$, since $L^{J_1}(s_1, c_1)$ satisfies both disjuncts in the matching constraint, while $\text{Val}(L^{J_1}(s_2, c_1)) = 1$. Thus, the total weight of the link instance J_1 is 3. Similarly, the other link instance containing $L(s_1, c_1)$, namely J_2, also has weight 3. The remaining link instances J_3 and J_4 have weight of 2. Hence, $\langle I, J_1\rangle$ and $\langle I, J_2\rangle$ are the two maximum weight repairs of $\langle I, I^*\rangle$ in this example. It follows that there is precisely one certain link of L on $\langle I, I^*\rangle$ w.r.t. Σ and the weight function w_Σ, namely $L(s_1, c_1)$. This is in contrast with the earlier case of maximum cardinality repairs, where we had no certain links.

Consider the above entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$. It is easy to verify that if I is an **S**-instance, then the following statements are equivalent for an **L**-instance J:

1. $\langle I, J\rangle$ is a maximum weight repair of $\langle I, I^*\rangle$ with respect to Σ and w_Σ.
2. J is a maximum-value solution for I with respect to Σ, as defined in [7].

It follows that the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ coincides with the entity-linking scenario given by $\mathcal{L}_\mathbf{0}(\oplus)$ in [7].

▶ **Framework 11.** *The entity-linking frameworks* $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{w})$ *of maximum-value solutions with weighted disjuncts.*

For each matching constraint $L(x, y) \to \forall \mathbf{u}(\psi(x, y, \mathbf{u}) \to \alpha_1 \vee \ldots \vee \alpha_k)$ *of* $\mathcal{L}_\mathbf{0}$ *and for each disjunct* $\alpha_i ::= \exists \mathbf{z}\ \phi_i(x, y, \mathbf{u}, \mathbf{z})$*, let* $w_{\phi_i}(x, y, \mathbf{u}, \mathbf{z})$ *be a function that returns non-negative numbers. Intuitively, with each disjunct that returns true or false, we also have a function that computes a weight for that disjunct. This collection of functions* w_{ϕ_i} *gives rise to a weight function* $\mathcal{V}_\mathbf{w}$ *that is computed as in the case of* $\mathcal{V}_0$ *except that in formula (2) we replace the number 1 by* $w_{\phi_i}(a, b, \mathbf{u}_0, \mathbf{z}_0)$.

Note that each different collection of functions w_{ϕ_i} *gives rise to a different entity-linking framework* $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{w})$*. This family of frameworks captures the entity-linking scenarios given by* $\mathcal{L}_\mathbf{0}(\oplus, \mathbf{w})$*, which, as discussed in [7], is of special interest because of its connection to probabilistic methods for entity resolution, including those based on Markov Logic Networks (MLNs) [22].*

Next, we state a general theorem for enumerating all maximum weight repairs with polynomial delay and for computing the certain links in polynomial time. Several results in [7], including Theorem 5.4, are special cases of this theorem.

▶ **Theorem 12.** *Let* $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{W})$ *be an entity-linking framework such that for each* $\Sigma \in \mathbf{S}_0$*, for each* **S***-instance* I*, for each sub-instance* J *of* I^**, and for each fact* $L^J(a, b)$*, we have that* $w_\Sigma(\langle I, I^*\rangle, L^{I^*}(a, b)) = w_\Sigma(\langle I, J\rangle, L^J(a, b))$*. Then the following statements are true.*

1. *There is a polynomial-delay algorithm that, given an* **S***-instance* I*, enumerates the maximum weight repairs of* $\langle I, I^*\rangle$*.*
2. *There is a polynomial-time algorithm that, given an* **S***-instance* I*, computes the certain links of* $\langle I, I^*\rangle$ *with respect to* Σ *and* w_Σ*.*

Note that the hypothesis of Theorem 12 is satisfied by the preceding three entity-linking frameworks. In particular, in all three frameworks, the weight of a link fact does not depend on the link instance J in which it appears. The proof of Theorem 12 is essentially the same as the proof of Theorem 5.4 in [7], where the problem is reduced to computing and enumerating maximum-weight matchings in undirected weighted bipartite graphs.

3.2 Collective Entity-Linking Frameworks

We now consider a language $\mathcal{L}_\mathbf{c}$ that is richer than $\mathcal{L}_\mathbf{0}$ and allows for link relations to appear in the right-hand side of matching constraints. Thus, the language $\mathcal{L}_\mathbf{c}$ allows us to express what is usually called *collective entity linking* [5], that is, the process of creating or specifying multiple inter-dependent links.

Concretely, in $\mathcal{L}_\mathbf{c}$, the matching constraint for a link symbol L has the same form

$$L(x,y) \rightarrow \forall \mathbf{u}(\psi(x,y,\mathbf{u}) \rightarrow \alpha_1 \vee \ldots \vee \alpha_k)$$

as in $\mathcal{L}_\mathbf{0}$, with the difference that in each disjunct $\alpha_i ::= \exists \mathbf{z}\ \phi_i(x,y,\mathbf{u},\mathbf{z})$, the formula ϕ_i can now be a conjunction of source *and link* atomic formulas, along with equalities. Thus, the matching constraint for L is allowed to refer to other link symbols (possibly, including L itself). As an example, which we give shortly, in $\mathcal{L}_\mathbf{c}$ one can express matching constraints to specify both publication links and venue links, where the matching constraint for publication links may depend on the links between venues, and the matching constraint for venue links may depend on the links between publications.

Based on the language $\mathcal{L}_\mathbf{c}$, we can define two entity-linking frameworks, one that does not allow for recursion among the links, and one that does allow for recursion.

▶ **Framework 13.** *The entity-linking framework* $(\mathcal{L}_\mathbf{c}, \mathcal{S}_\mathbf{1}, \mathcal{V}_\mathbf{1})$ *for recursion-free collective entity linking.*

In this framework, $\mathcal{S}_1$ *is the collection of all finite sets of constraints from* $\mathcal{L}_\mathbf{c}$*, such that for each link symbol* L*, the set* Σ *contains the two inclusion dependencies on* L*, it contains zero, one or two functional dependencies on* L*, and at most one matching constraint on* L*. Additionally, we require that there is no recursion through the links. Thus, for each* Σ *in* $\mathcal{S}_1$*, there is implicitly a hierarchy of link symbols, and a matching constraint for* L *may call only links that are strictly lower in the hierarchy than* L*. Additionally,* $\mathcal{V}_1$ *is the collection of weight functions that associates with each* Σ *in* $\mathcal{S}_1$ *a weight function* w_Σ *defined in the same way as in the entity-linking framework* $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$*.*

▶ **Framework 14.** *The entity-linking framework* $(\mathcal{L}_\mathbf{c}, \mathcal{S}_\mathbf{2}, \mathcal{V}_\mathbf{2})$ *for recursive collective entity linking is defined in the same way as* $(\mathcal{L}_\mathbf{c}, \mathcal{S}_\mathbf{1}, \mathcal{V}_\mathbf{1})$ *except that* $\mathcal{S}_2$ *allows recursion through the links.*

▶ **Example 15.** Consider a bibliographic example from [7], where we link papers (from one database) with articles (from another database), while also linking the corresponding venues. The source schema $\mathbf{S}$ consists of $\texttt{Paper}(pid, title, venue, year)$ and $\texttt{Article}(ano, title, journal, year)$. Here, `pid` is a unique id assigned to `Paper` records, while `venue` could be a conference, a journal, or some other place of publication. The `Article` relation represents publications that appeared in journals, and `ano` is a unique id assigned to such records. The link schema $\mathbf{L}$ consists of two relations: $\texttt{PaperLink}\ (pid, ano)$ and $\texttt{VenueLink}\ (venue, journal)$. The first relation is intended to link paper ids from `Paper` with article numbers from `Article`, when they represent the same publication. The second relation is intended to relate `journal` values that occur in `Article` (e.g., "ACM TODS") to journal values that occur under the `venue` field in `Paper` (e.g., "TODS").

A possible entity linking specification in the framework $(\mathcal{L}_\mathbf{c}, \mathcal{S}_\mathbf{2}, \mathcal{V}_\mathbf{2})$ is $(\mathcal{L}_\mathbf{c}, \Sigma, w_\Sigma)$, where Σ contains the following two matching constraints:

$$\begin{aligned}\texttt{VenueLink}(ven, jou) \rightarrow\ & (ven \sim_1 jou)\\ & \vee \exists pid, t_1, y_1, ano, t_2, y_2\ (\ \texttt{Paper}(pid, t_1, ven, y_1)\\ & \qquad \wedge \texttt{Article}(ano, t_2, jou, y_2)\\ & \qquad \wedge \texttt{PaperLink}(pid, ano)\ )\end{aligned}$$

$$\begin{aligned}&\texttt{PaperLink}(pid, ano) \rightarrow\\ &\quad \forall t_1, ven, y_1, t_2, jou, y_2\ (\ \texttt{Paper}(pid, t_1, ven, y_1) \wedge \texttt{Article}(ano, t_2, jou, y_2)\\ &\qquad \rightarrow ((t_1 \sim_2 t_2)\ \wedge\ (y_1 = y_2))\\ &\qquad \vee ((t_1 \sim_2 t_2)\ \wedge\ \texttt{VenueLink}(ven, jou))\ )\end{aligned}$$

The first constraint specifies that we may link a venue with a journal only if their string values are similar (via some similarity predicate $\sim_1$), or if there are papers and articles that have been published in the respective venue and journal and that are linked via `PaperLink`. The second constraint specifies that we may link a paper with an article only if their titles are similar (via a similarity predicate $\sim_2$) and their years of publication match exactly, or if their titles are similar and their venues of publications are linked via `VenueLink`.

Additionally, Σ includes two functional dependencies on `PaperLink`: $pid \to ano$, $ano \to pid$, to reflect that each paper id in `Paper` must match to at most one article number in `Article`, and vice-versa. We do not require any functional dependencies on `VenueLink`; thus, we could have multiple venue strings in `Paper` matching with a journal string in `Article`, and vice-versa. We also include in Σ the expected inclusion dependencies from the link attributes to the corresponding source attributes (e.g., `PaperLink`$[pid] \subseteq$ `Paper`$[pid]$).

With a simple modification, where we remove the second disjunct in the matching constraint for `PaperLink`, we obtain a different entity linking specification that is in the recursion-free collective entity-linking framework $(\mathcal{L}_{\mathbf{c}}, \mathcal{S}_{\mathbf{1}}, \mathcal{V}_{\mathbf{1}})$.

We point out that the entity-linking framework $(\mathcal{L}_{\mathbf{c}}, \mathcal{S}_{\mathbf{1}}, \mathcal{V}_{\mathbf{1}})$ coincides with the entity-linking scenario given by $\mathcal{L}_{\mathbf{1}}(\oplus)$ in [7], while entity-linking framework $(\mathcal{L}_{\mathbf{c}}, \mathcal{S}_{\mathbf{2}}, \mathcal{V}_{\mathbf{2}})$ coincides with the entity-linking scenario given by $\mathcal{L}_{\mathbf{2}}(\oplus)$ in [7].

For the above two entity-linking frameworks, it is important to note that the weight functions depend on the link instance in a crucial way. In particular, the hypothesis of the preceding Theorem 12, stating that the weight of a link fact only depends on I^* and not on the link instance J, is no longer satisfied. In fact, as shown in [7] (Theorem 7.3), Theorem 12 fails even for $(\mathcal{L}_{\mathbf{c}}, \mathcal{S}_{\mathbf{1}}, \mathcal{V}_{\mathbf{1}})$, unless NP = coNP.

4 Comparing the Expressive Power of Entity-Linking Frameworks

The notion of certain links makes it possible to compare the expressive power of entity-linking frameworks. In the next definition, we first introduce the notion of *certain-link equivalence* between entity-linking specifications. This notion is of interest as a tool to compare entity-linking specifications in a way other than logical equivalence (which may be too strict for entity linking purposes). The second part of the definition then makes use of certain-link equivalence to define a notion of *subsumption* between entity-linking frameworks.

▶ **Definition 16.** Let $\mathbf{S}$ be a schema of source symbols, let $\mathbf{L}$ be a schema of link symbols, let $\mathbf{R} = \mathbf{S} \cup \mathbf{L}$. Assume that $\mathcal{F} = (\mathcal{L}, \mathcal{S}, \mathcal{W})$ and $\mathcal{F}' = (\mathcal{L}', \mathcal{S}', \mathcal{W}')$ are two entity-linking frameworks on $\mathbf{R}$.

- Let $\mathcal{E} = (\mathcal{L}, \Sigma, w_\Sigma)$ be an entity-linking specification in $\mathcal{F}$, and let $\mathcal{E}' = (\mathcal{L}', \Sigma', w_{\Sigma'})$ be an entity-linking specification in $\mathcal{F}'$. We say that *$\mathcal{E}$ and $\mathcal{E}'$ are certain-link equivalent* if for every link symbol L in $\mathbf{L}$ and every $\mathbf{R}$-instance $\langle I, J\rangle$, we have that the certain links of L on $\langle I, J\rangle$ with respect to Σ and w_Σ coincide with the certain links of L on $\langle I, J\rangle$ with respect to Σ' and $w_{\Sigma'}$.
- We say that *$\mathcal{F}$ is subsumed by $\mathcal{F}'$*, denoted $\mathcal{F} \preceq \mathcal{F}'$, if for every entity-linking specification $\mathcal{E}$ of $\mathcal{F}$ there is an entity-linking specification $\mathcal{E}'$ of $\mathcal{F}'$ such that $\mathcal{E}$ and $\mathcal{E}'$ are certain-link equivalent. Otherwise, we say that *$\mathcal{F}$ is not subsumed by $\mathcal{F}'$*, and write $\mathcal{F} \not\preceq \mathcal{F}'$.
- We say that *$\mathcal{F}$ is strictly subsumed by $\mathcal{F}'$* if $\mathcal{F} \preceq \mathcal{F}'$, but $\mathcal{F}' \not\preceq \mathcal{F}$.

We note that a weaker notion of subsumption was considered implicitly in [7] for concrete entity linking scenarios. In this weaker notion, certain-link equivalence holds for repairs of the instance $\langle I, I^*\rangle$ instead of arbitrary instances $\langle I, J\rangle$. In effect, Theorem 6.2 in [7] asserts that *linear* MLNs, an important special case of MLNs, are subsumed under this weaker notion of subsumption by an entity linking framework of maximum-value solutions with weighted disjuncts, where the matching constraints are in the existential fragment $\exists\mathcal{L}_\mathbf{0}$ of the language $\mathcal{L}_\mathbf{0}$.

We note that for of all our subsumption results (Theorems 17, 18, 19, and 23), whenever we prove failure of subsumption, we actually prove it in a stronger sense, by showing that it fails even under the weaker notion.

The next two theorems say that the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ of maximum-value solutions and the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{W}_\mathbf{1})$ of maximum cardinality repairs are incomparable in expressive power, in that neither subsumes the other.

▶ **Theorem 17.** *The entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ of maximum-value solutions is not subsumed by the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{W}_\mathbf{1})$ of maximum cardinality repairs.*

Proof. (Hint) Our entity-linking specification in $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ that is not certain-link equivalent to any entity-linking specification in $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{W}_\mathbf{1})$ has one link symbol L, the matching constraint $L(x, y) \rightarrow R(x, y) \vee S(x, y) \vee T(x, y)$, the FD $L: \ X \rightarrow Y$, and the inclusion dependencies $L[X| \subseteq D$ and $L[Y| \subseteq D$. ◀

▶ **Theorem 18.** *The entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{W}_\mathbf{1})$ of maximum cardinality repairs is not subsumed by the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ of maximum-value solutions.*

Proof. (Hint) Our entity-linking specification in $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{W}_\mathbf{1})$ that is not certain-link equivalent to any entity-linking specification in $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ has one link symbol L, the matching constraint $L(x, y) \rightarrow R(x, y) \vee S(x, y)$, the FD $L: \ X \rightarrow Y$, and the inclusion dependencies $L[X| \subseteq D_1$ and $L[Y| \subseteq D_2$. ◀

By definition, the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ is subsumed by the entity-linking framework $(\mathcal{L}_\mathbf{c}, \mathcal{S}_\mathbf{1}, \mathcal{V}_\mathbf{1})$. The next theorem says that this subsumption is strict. This means that allowing for link relations to appear on the right-hand side of matching constraints gives strictly more expressive power than not allowing this, even when the dependencies among the link relations are non-recursive.

▶ **Theorem 19.** *The entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ of maximum-value solutions is strictly subsumed by the entity-linking framework $(\mathcal{L}_\mathbf{c}, \mathcal{S}_\mathbf{1}, \mathcal{V}_\mathbf{1})$ for recursion-free collective entity linking.*

Proof. (Hint) Our entity-linking specification in $(\mathcal{L}_\mathbf{c}, \mathcal{S}_\mathbf{1}, \mathcal{V}_\mathbf{1})$ that is not certain-link equivalent to any entity-linking specification in $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ has two link symbols L_1 and L_2, ,the matching constraints $L_1(x, y) \rightarrow (S(x, y) \rightarrow (L_2(x, y) \wedge R(x, y))$ and $L_2(x, y) \rightarrow (P(x, y) \rightarrow T(x, y))$ and the inclusion dependencies $L_1[X| \subseteq D$, $L_1[Y| \subseteq D$, $L_2[X] \subseteq D$, and $L_2[Y] \subseteq D$. There are no FDs. ◀

5 Adding Preference Constraints

In this section, we introduce a family of entity-linking frameworks $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{P}_\mathbf{\Pi})$ that is parameterized by a set of Π *preference constraints*. This family of frameworks can be seen as an extension of the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$, where we use a more refined collection of weight functions that also take into account preferences among the link facts.

We first introduce the language of preference constraints from which Π is drawn. The main motivation for such preference constraints is that they allow a user to specify explicitly whether some link facts should be considered stronger than other link facts. Such preference constraints are given independently of, and in addition to, the set Σ of constraints in $\mathcal{S}_0$, and will be used to further differentiate among conflicting links (i.e., pairs of link facts that violate one or both of the functional dependencies on a link relation).

A *preference constraint* has the following general form:

$$L(x,y) \wedge L(x',y') \wedge\ \alpha(x,y) \wedge \neg\alpha(x',y') \rightarrow L(x,y) \geq L(x',y') \tag{3}$$

In the above, L can be any of the link symbols in $\mathbf{L}$ while $\alpha(x,y)$ can be any predicate of the form $\exists\mathbf{z}\ \phi(x,y,\mathbf{z})$, where ϕ is a conjunction of source atomic formulas along with equalities.

▶ **Example 20.** Consider a variation of the earlier Example 7 linking subsidiaries with companies, where the set Σ of constraints is as follows. The functional and inclusion dependencies are as before. However, the matching constraint is simplified, for the purposes of this example, so that it now requires only the similarity of the subsidiary name and company name:

$$\begin{aligned} L(sid, cid) \rightarrow \forall sn, loc, cn, hd\ (\texttt{Subsid}(sid, sn, loc) \wedge \texttt{Company}(cid, cn, hd) \\ \rightarrow (sn \sim cn)\ . \end{aligned}$$

We now consider, additionally, a set Π consisting of a single preference constraint, which uses an Exec-based condition to differentiate among links:

$$\begin{aligned} &L(sid, cid) \wedge L(sid', cid') \\ &\quad \wedge \exists e, n, t, sn, loc\ (\texttt{Exec}(e, cid, n, t) \wedge \texttt{Subsid}(sid, sn, loc)\ \wedge contains(t, sn)\,) \\ &\quad \wedge \neg\exists e, n, t, sn, loc\ (\texttt{Exec}(e, cid', n, t) \wedge \texttt{Subsid}(sid', sn, loc) \wedge contains(t, sn)\,) \\ &\quad \rightarrow L(sid, cid) \geq L(sid', cid') \end{aligned}$$

Thus, whenever we have two links relating a subsidiary with a company, if one of the links satisfies the fact that the company has an executive whose title contains the subsidiary name, while the other link does not satisfy such fact, we prefer the first link over the second link.

Note that a user has the freedom, in general, to choose which conditions to push into the matching constraints of Σ and which ones into the preference constraints of Π. This is manifested, in this example, via the fact that the executive information is used in a preference constraint whereas before it was used as part of a matching constraint.

The notion of a consistent instance when there are preference constraints continues to be the same as that of a consistent instance with respect to an entity-linking specification in $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ where there are no preference constraints. Thus, the set Π of preference constraints plays no role in defining consistent instances under $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{P}_\mathbf{\Pi})$. However, Π plays an important role in defining the weight functions for the links, as we see next.

We are now ready to formally define $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{P}_\mathbf{\Pi})$. First, we recall from Section 3 the instance $\langle I, I^*\rangle$, which for a given source instance I, represents a superset for the repairs that we consider. Thus, I^* represents the domain for all the links that may appear in link relations.

▶ **Framework 21.** *The family of entity-linking frameworks $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{P}_\mathbf{\Pi})$ with preference constraints.*

For every fixed finite set Π of preference constraints, we define an entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{P}_\mathbf{\Pi})$, by assigning to each $\Sigma \in \mathcal{S}_0$ a weight function $w_{\Sigma,\Pi}$ that depends on both Σ and Π. Given an $\mathbf{R}$-instance $\langle I, J\rangle$ and a fact $L^J(a,b)$, we define $w_{\Sigma,\Pi}(\langle I, J\rangle, L^J(a,b))$ to be $w_{\Sigma,\Pi}(\langle I, I^*\rangle, L^{I^*}(a,b))$, which in turn is defined as follows.

For each link symbol L, and source instance I, we first compute a preference relation $\geq_L$ on I^* on conflicting links of L, by evaluating each preference constraint of the form (3) that involves L. Concretely, whenever (x_0, y_0) and (x_0', y_0') are pairs in I^* such that $L(x_0, y_0)$ and $L(x_0', y_0')$ are conflicting (i.e., together violate one or both of the functional dependencies on L), and such that $\alpha(x_0, y_0)$ is true in I but $\alpha(x_0', y_0')$ is not true in I, we set $L(x_0, y_0) \geq_L L(x_0', y_0')$. In general, $\geq_L$ can have cycles. For example, we can have two distinct pairs $l = L(x_0, y_0)$ and $l' = L(x_0', y_0')$ such that $l \geq_L l'$ and $l' \geq_L l$. Such situation may arise when a user gives (at least) two preference constraints for L, the evaluation of which leads to opposite preferences for the particular links.

We then turn $\geq_L$ into an acyclic relation $>_L$ as follows. First, we take the transitive closure $\geq_L^*$ of $\geq_L$. Then, we set $l >_L l'$ whenever $l \geq_L^* l'$ but it is not the case that $l' \geq_L^* l$. Intuitively, $l >_L l'$ means that l is strictly preferred to l'. It can be verified that, for each L, the relation $>_L$ (or rather its inverse $<_L$) forms a strict partial order. We may also drop the subscript L and use the notation $>$ or $(\geq)$ whenever L is understood from the context. We may refer to $>$ as the preference relation.

The weight of a link fact l in I^* is then defined recursively:

$$w_{\Sigma,\Pi}(\langle I, I^*\rangle, l) = w_{\Sigma}(\langle I, I^*\rangle, l) + \sum_{l > l'} w_{\Sigma,\Pi}(\langle I, I^*\rangle, l'),$$

where w_Σ is the weight function associated with Σ in the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ of maximum-value solutions. Thus, the weight of l is obtained by adding up $w_\Sigma(\langle I, I^*\rangle, l)$, which is calculated solely based on Σ as defined for $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$, with the total aggregated weight of all the links that l dominates (via the preference relation $>$). In the special case when there are no preference constraints, the weight of a link l falls back to $w_\Sigma(\langle I, I^*\rangle, l)$. Thus, for each Π, the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{P}_\mathbf{\Pi})$ is an extension of the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$.

Note that, by definition, the weight of a link is relative to $\langle I, I^*\rangle$, on which we evaluated the preference constraints, but independent of any particular sub-instance $\langle I, J\rangle$. Thus, the hypothesis of Theorem 12 holds, by definition.

▶ **Example 22.** Recall the specification in Example 20. First, it is immediate to see that this is an example of an entity-linking specification in the entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{P}_\mathbf{\Pi})$, for the given set Π of preference constraints. Moreover, let us assume the same source instance I as in Example 9. The link $L(s_1, c_1)$ strictly dominates the link $L(s_1, c_2)$ (by the fact that c_1 satisfies the `Exec` condition for s_1 in the preference constraint, while c_2 does not). Since no other strict domination holds, we have that $w_{\Sigma,\Pi}(\langle I, I^*\rangle, L^{I^*}(s_1, c_1)) = 2$, while the weight of any other link is 1. As a consequence, among the four maximal cardinality repairs for $\langle I, I^*\rangle$ that we have seen earlier, we have that $\langle I, J_1\rangle$ and $\langle I, J_2\rangle$ have weight 3, while $\langle I, J_3\rangle$ and $\langle I, J_4\rangle$ have weight 2. Thus, $\langle I, J_1\rangle$ and $\langle I, J_2\rangle$ are the maximum weight repairs with respect to Σ and $w_{\Sigma,\Pi}$. As a result, we also obtain that $L(s_1, c_1)$ is the sole certain link, in this example.

As we noted above, the hypothesis of Theorem 12 holds for $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{P}_\mathbf{\Pi})$ and so we obtain, as a corollary, a polynomial-delay algorithm for the enumeration of maximum weight repairs and a polynomial-time algorithm for the computation of the certain links.

It is clear that every entity-linking framework $(\mathcal{L}_\mathbf{0}, \mathcal{S}_\mathbf{0}, \mathcal{V}_\mathbf{0})$ (Framework 10) can be simulated by using an entity-linking framework involving preferences (Framework 21) by simply taking the set Π of preferences to be empty. The next theorem says that, in fact, we gain expressive power by allowing preference constraints. This is our main technical result.

▶ **Theorem 23.** *There is a finite set Π of preference constraints such that the corresponding framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{P}_{\mathbf{\Pi}})$ is not subsumed by the entity-linking framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$ of maximum-value solutions.*

A key tool in the proof of Theorem 23 is a locality theorem that is interesting in its own right, and that we use multiple times in the proof of Theorem 23. We first need some preliminaries. For each entry a in a fact in an instance I, define $N_0(a)$ to be $\{a\}$. Inductively, define $N_{i+1}(a)$ to consist of $N_i(a)$ along with each c such that there is a' in $N_i(a)$ where a' and c are both entries in some fact in I. Thus $N_r(a)$ consists of those entries of I within distance r of a in the Gaifman graph [20] of I. Let $N_r(a, b)$ be $N_r(a) \cup N_r(b)$. We may refer to $N_r(a, b)$ as an *r-neighborhood.*

▶ **Theorem 24** (Locality Theorem). *Let $\mathcal{E}$ be an entity-linking specification in $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$, with link symbol L. Then there is r, depending only on $\mathcal{E}$, such that for every source instance I, if $I \upharpoonright N_r(a_1, b_1)$ and $I \upharpoonright N_r(a_2, b_2)$ are isomorphic under an isomorphism f with $f(a_1) = a_2$ and $f(b_1) = b_2$, then the weights of the links $L(a_1, b_1)$ and $L(a_2, b_2)$ in $\mathcal{E}$ are the same.*

By $I \upharpoonright N_r(a_i, b_i)$ we mean the usual notion of the restriction of I to the domain $N_r(a_i, b_i)$. The proof of the Locality theorem makes use of the Gaifman locality theorem for first-order logic [12], and extensions of that theorem to logics with counting by Libkin [19]. Our proof of the Locality Theorem depends on a certain uniformity in the choice of r.

Sketch of the proof of Theorem 23. Our entity-linking specification $\mathcal{E}$ in the framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{P}_{\mathbf{\Pi}})$ has one link symbol L, the matching constraint $L(x, y) \rightarrow R(x, y)$, both FDs on L, and the inclusion dependencies $L[X| \subseteq R[X]$ and $L[Y| \subseteq R[Y]$. We define a family of source instance K_r and a set of preference constraints such that we get two long chains $L(0, 1) > L(2, 3) > L(4, 5) > \cdots > L(m, m + 1)$ and $L(0, 1') > L(2', 3') > L(4', 5') > \cdots > L(n', (n+1)')$ of strict preferences, where $m > n$ (so the first chain is longer than the second). It is shown that $L(0, 1)$ has so much weight that it is a certain link for $\mathcal{E}$. However, given an entity-linking specification $\mathcal{E}'$ in the entity-linking framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$ of maximum-value solutions, when we select r based on $\mathcal{E}'$, the source instance $K = K_r$ is designed so that the neighborhoods $K \upharpoonright N_r(0, 1)$ and $K \upharpoonright N_r(0, 1')$ are isomorphic, and so by the Locality Theorem, $L(0, 1)$ and $L(0, 1')$ have the same weight in $\mathcal{E}'$.

Assume, by way of contradiction, that there is an entity-linking specification $\mathcal{E}'$ in the entity-linking framework $(\mathcal{L}_{\mathbf{0}}, \mathcal{S}_{\mathbf{0}}, \mathcal{V}_{\mathbf{0}})$ that is certain-link equivalent to $\mathcal{E}$. By considering an instance with only one fact $R(0, 1)$, we show that $\mathcal{E}'$ has the same inclusion dependencies as $\mathcal{E}$. We show that $\mathcal{E}'$ has both FDs on L with the following argument. Assume first that $\mathcal{E}'$ does not have the FD $L: \ Y \rightarrow X$. Since $L(0, 1')$ has the same weight in $\mathcal{E}'$ as $L(0, 1)$, in particular $L(0, 1')$ satisfies the matching constraint for $\mathcal{E}'$. Now $L(0, 1')$ is not a certain link in $\mathcal{E}'$, since it is not a certain link in $\mathcal{E}$. So let $\langle K, N \rangle$ be a maximum weight repair of $\langle K, K^* \rangle$ that does not contain $L(0, 1')$. Then of course N contains the certain link $L(0, 1)$. Form N' by replacing $L(0, 1)$ in N by $L(0, 1')$. Now N' satisfies the only possible FD $L: \ X \rightarrow Y$, and it satisfies the inclusion dependencies and matching constraint. Furthermore, N' has the same weight as N, since $L(0, 1)$ and $L(0, 1')$ have the same weight, and so $\langle K, N' \rangle$ is a maximum weight repair. But this is a contradiction, since $\langle K, N' \rangle$ is a maximum weight repair that does not contain the certain link $L(0, 1)$. Now define the instance $U(K)$, where (a, b) is a tuple of a relation of K if and only if $(\underline{b}, \underline{a})$ is a tuple of the corresponding relation of $U(K)$, and where $\underline{a}$ and $\underline{b}$ are new values. The proof that the FD $L: \ X \rightarrow Y$ holds for $\mathcal{E}'$ is the same, except rather than replacing the certain link $L(0, 1)$ in a maximum weight repair of $\langle K, K^* \rangle$ by $L(0, 1')$, we instead replace the certain link $L(\underline{1}, \underline{0})$ in a maximum weight repair of $\langle U(K), (U(K))^* \rangle$ by $L(\underline{1}', \underline{0})$.

We explicitly find the set M of certain links for $I = K \cup U(K)$ in $\mathcal{E}$ and prove, using the FDs and inclusion dependencies for $\mathcal{E}'$, that $\langle I, M\rangle$ is the unique maximum weight repair for $\langle I, I^*\rangle$ in $\mathcal{E}'$. Let M' consist precisely of all of the links of $\mathcal{E}$ that are not links in M. We prove, again using the Locality Theorem, that there is a one-to-one correspondence between the links ℓ of M and the links ℓ' of M', where ℓ and ℓ' have the same weight in $\mathcal{E}'$. In particular, each link of M' satisfies the entity-linking specification of $\mathcal{E}'$. Further, since M' also satisfies both FDs and the inclusion dependencies, it follows that $\langle I, M'\rangle$ is a maximum weight repair. But this is a contradiction, since $\langle I, M\rangle$ is the unique maximum weight repair. ◀

6 Concluding Remarks

In this paper, we introduced and explored a unifying approach to entity linking. This approach, which is based on the notion of an entity linking framework and the notion of the certain links in such a framework, provides a single formalism for modeling different entity linking scenarios and for comparing them using the certain links as a measure of their expressive power. To this effect, we defined a notion of *certain-link equivalence* that allows us to compare entity-linking specifications, in a way other than logical equivalence (which may be too strict for entity linking purposes). We then made use of certain-link equivalence to define what it means for an entire entity-linking framework to subsume another one. We established a number of technical results that delineate the comparative expressive power of several concrete entity linking frameworks. Our concrete focus in this paper was on the comparison of the entity linking framework of maximum-value solutions with entity linking frameworks (1) that involve maximum cardinality repairs, (2) that allow recursion-free collective entity linking, and (3) that incorporate preferences among links.

A next step in this investigation is to understand the expressive power of recursive collective entity linking. Specifically, we conjecture that the framework $(\mathcal{L}_{\mathbf{c}}, \mathcal{S}_{\mathbf{2}}, \mathcal{V}_{\mathbf{2}})$ of recursive collective entity linking cannot be subsumed by the framework $(\mathcal{L}_{\mathbf{c}}, \mathcal{S}_{\mathbf{1}}, \mathcal{V}_{\mathbf{1}})$ of non-recursive collective entity linking. Another next step has to do with Markov Logic Networks (MLNs), which were first studied in [22]. As stated earlier, it follows from results in [7] that linear MLNs are subsumed by an entity linking framework of maximum-value solutions with weighted disjuncts, where the constraints are in the existential fragment $\exists\mathcal{L}_{\mathbf{0}}$ of the language $\mathcal{L}_{\mathbf{0}}$. It is an open problem if more general MLNs (i.e., not necessarily linear) can be subsumed by an entity linking framework of maximum-value solutions with weighted disjuncts for some suitable choice of weights and constraints from $\mathcal{L}_{\mathbf{0}}$ or from the more general language $\mathcal{L}_{\mathbf{c}}$.

In a different direction, we note that our unifying approach to entity linking is flexible enough to allow assigning *probabilities* to links in a natural way. Specifically, we can define the probability $Pr(L(a,b))$ of a link $L(a,b)$ to be the number of maximum weight repairs containing $L(a,b)$ divided by the total number of maximum weight repairs. Thus, a link $L(a,b)$ is certain if and only if $Pr(L(a,b)) = 1$. The introduction of probabilities in entity linking frameworks raises several algorithmic questions, including the question of enumerating the links whose probability is above a fixed threshold, say, enumerating all links $L(a,b)$ such that $Pr(L(a,b)) \geq 0.75$. Furthermore, it may be possible to establish tight connections between our approach and other approaches in entity linking and entity resolution, such as Probabilistic Soft Logic (PSL) [3, 4, 6], that derive links with *scores* based on weighted first-order formulas. By utilizing such connections, one may also be able to transfer the formalism of preference constraints, which fits naturally in our declarative approach, into PSL (or into MLN as well). In general, we may obtain more powerful entity linking approaches that combine declarative, logic-based specification with probabilistic reasoning and with explicit user preference constraints.

References

1 Arvind Arasu, Christopher Re, and Dan Suciu. Large-Scale Deduplication with Constraints using Dedupalog. In *ICDE*, pages 952–963, 2009.

2 Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki. Consistent Query Answers in Inconsistent Databases. In *PODS*, pages 68–79, 1999.

3 Stephen H. Bach. *Hinge-Loss Markov Random Fields and Probabilistic Soft Logic: A Scalable Approach to Structured Prediction.* PhD thesis, University of Maryland, 2015.

4 Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. Hinge-Loss Markov Random Fields and Probabilistic Soft Logic. *CoRR*, abs/1505.04406, 2015.

5 Indrajit Bhattacharya and Lise Getoor. Collective Entity Resolution in Relational Data. *TKDD*, 1(1), 2007.

6 Matthias Bröcheler, Lilyana Mihalkova, and Lise Getoor. Probabilistic Similarity Logic. In *UAI 2010, Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, July 8-11, 2010*, pages 73–82, 2010.

7 Douglas Burdick, Ronald Fagin, Phokion G. Kolaitis, Lucian Popa, and Wang-Chiew Tan. A Declarative Framework for Linking Entities. *ACM Trans. Database Syst.*, 41(3):17, 2016. Preliminary version appeared in ICDT, pages 25–43, 2015.

8 Xin Dong, Alon Y. Halevy, and Jayant Madhavan. Reference Reconciliation in Complex Information Spaces. In *SIGMOD*, pages 85–96, 2005.

9 Jianfeng Du, Guilin Qi, and Yi-Dong Shen. Weight-Based Consistent Query Answering over Inconsistent SHIQ Knowledge Bases. *Knowl. Inf. Syst.*, 34(2):335–371, 2013.

10 Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate Record Detection: A Survey. *IEEE TKDE*, 19(1):1–16, 2007.

11 Ivan P. Fellegi and Alan B. Sunter. A Theory for Record Linkage. *J. Am. Statistical Assoc.*, 64(328):1183–1210, 1969.

12 Haim Gaifman. On Local and Non-Local Properties. *Proc. Herbrand Symp. - Logic Colloquium '81*, 1982.

13 Helena Galhardas, Daniela Florescu, Dennis Shasha, Eric Simon, and Cristian-Augustin Saita. Declarative Data Cleaning: Language, Model, and Algorithms. In *VLDB*, pages 371–380, 2001.

14 Mauricio A. Hernández, Georgia Koutrika, Rajasekar Krishnamurthy, Lucian Popa, and Ryan Wisnesky. HIL: A High-Level Scripting Language for Entity Integration. In *EDBT*, pages 549–560, 2013.

15 Mauricio A. Hernández and Salvatore J. Stolfo. The Merge/Purge Problem for Large Databases. In *SIGMOD*, pages 127–138, 1995.

16 Hanna Köpcke and Erhard Rahm. Frameworks for Entity Matching: A Comparison. *Data Knowl. Eng.*, 69(2):197–210, 2010.

17 Hanna Köpcke, Andreas Thor, and Erhard Rahm. Evaluation of Entity Resolution Approaches on Real-World Match Problems. *PVLDB*, 3(1):484–493, 2010.

18 Nick Koudas, Sunita Sarawagi, and Divesh Srivastava. Record Linkage: Similarity Measures and Algorithms. In *SIGMOD*, pages 802–803, 2006.

19 Leonid Libkin. Logics with Counting and Local Properties. *ACM Transactions on Computational Logic*, 1(1):33–59, 2000.

20 Leonid Libkin. *Elements of Finite Model Theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

21 Andrei Lopatenko and Leopoldo E. Bertossi. Complexity of Consistent Query Answering in Databases Under Cardinality-Based and Incremental Repair Semantics. In *ICDT*, pages 179–193, 2007.

22 Matthew Richardson and Pedro Domingos. Markov Logic Networks. *Machine Learning*, 62(1-2):107–136, 2006.