Computing FOAF Co-reference Relations with Rules and Machine Learning[⋆]

Jennifer Sleeman and Tim Finin
University of Maryland, Baltimore County
Baltimore. MD 21250 USA
{jsleem1,finin}@cs.umbc.edu

Abstract. The friend of a friend (FOAF) vocabulary is widely used on the Web to describe 'agents' (people, groups and organizations) and their properties. Since FOAF does not require unique ID for agents, it is not clear when two FOAF instances should be linked as co-referent, i.e., denote the same entity in the world. One approach is to use logical constraints such as the presence of inverse functional properties as evidence that two individuals are the same. Another applies heuristics based on the string similarity of values of FOAF properties such as name and school as evidence for or against co-reference. Performance is limited, however, by many factors: non-semantic string matching, noise, changes in the world, and the lack of more sophisticated graph analytics. We describe a prototype system that takes a set of FOAF agents and identifies subsets that are believed to be co-referent. The system uses logical constraints (e.g., IFPs), strong heuristics (e.g., FOAF agents described in the same file are not co-referent), and an SVM generated classifier. We present initial results using data collected from Swoogle and other sources and describe plans for additional analysis.

Keywords: FOAF, machine learning, linked data

1 Introduction

The FOAF (Friend of a Friend) vocabulary has been one of the most widely used ontologies since the beginning of the Semantic Web. It defines classes and properties for describing entities (people, organizations and groups), their attributes, and their relations. FOAF's popularity is evident in the social networking sites that publish profiles using FOAF, the number of RDF documents using the FOAF namespace, the number of foaf:Agent instances or the volume of RDF triples using FOAF terms [8, 20]. Its widespread use can be explained by the common need to publish, find and reason with basic data on people and organizations and also the lightweight and practical design of the vocabulary.

One of the principles underlying the Semantic Web is that it enables us to give concepts and individuals URIs that serve as unique identifiers, removing much of the ambiguity that comes with using human language and representation systems that are not designed to be distributed and open. The FOAF ontology

[⋆] Partial support for this research was provided by National Science Foundation award 0910838 and the Johns Hopkins University Human Language Technology Center of Excellence.

allows one to create a foaf:Agent instance that indeed represents a single, unique entity and to describe its properties and relations.

What FOAF does not require is a property that represents a globally unique identifier (GUID) that can be used to recognize when two `foaf:Agent` individuals are co-referent, i.e., refer to the same individual whether real or fictional. It's considered good practice to associate a URI with each FOAF instance, which is a global ID, but not necessarily a unique one. It is common to find scores of FOAF descriptions on the Web for the same entity, each with a different URI.

This is a problem that is common to other representation systems, including human language, database address records, and even official government records. An entity's name is often a useful property for distinguishing it from other entities, but there are many people named `Michael Jordan` and several companies known as `Apple`. Given two such descriptions and depending on the context and task at hand, we may find enough evidence to conclude that the entity mentions are or are not `co-referent`. If there is enough supporting evidence to conclude that they are, we can integrate the information from the two descriptions.

There are many potential problems in integrating information from two sources once we've decided that they provide information on the same entity. One source may be known to be unreliable. The properties may be subjective. Sources may differ in their beliefs even about objective properties. The descriptions may include dynamic properties and refer to their values at different points in time.

In the rest of this paper we discuss the problem of deciding when RDF descriptions of two FOAF agents are likely to be co-referent. One common approach is to use the presence of FOAF properties declared to be inverse functions [28, 14, 17, 18] (e.g., foaf:mbox) as a validation that two individuals are the same. A second [27] applies heuristics based on the string similarity of values of FOAF properties such as name and school as evidence for or against co-reference.

We describe an 'ensemble' approach which uses both a rules-based model and a vector model and clustering to group sets of co-referent pairs. The vector model is part of a supervised machine learning method which uses features defined over pairs of FOAF individuals to produce a classifier for identifying co-referent FOAF instances. The rules-based model uses logical rules such as inverse functional properties and other identifying properties such as `owl:sameAs` and `owl:differentFrom`.

This approach accounts for spare data, in that when inverse functional properties or other properties that provide evidence of co-reference such as `owl:sameAs` are not present, the classifier can then be used to provide a prediction of co-reference. However, when these properties are present, taking advantage of their presence is practical. A machine learning approach is used to account for issues that arise when using a heuristic string similarity approach. Common properties are not always present in both profiles and noise in the data can obfuscate the fact that two profiles represent the same person. Noise and anomalies in the data can feasibly be captured in a classification model.

```
:coref a owl:TransitiveProperty.
:coref a owl:SymmetricProperty.
owl:sameAs rdfs:subPropertyOf :coref.
:notCoref a owl:SymmetricProperty.
owl:differentFrom rdfs:subPropertyOf :notCoref.
{?a :notCoref ?b. ?b :coref ?c.}  => {?a :notCoref ?c}
{?a foaf:knows ?b.}  => {?a :notCoref ?b}
```

Fig. 1: These statements and rules define the :coref and :notCoref properties that we use instead of owl:sameAs.

2 owl:sameAs Considered Dangerous

In our earlier work, we used `owl:sameAs` to model the semantics of our processing. That is, when we concluded that two FOAF agents referred to the same individual, we asserted that they were equivalent using `owl:sameAs`. The use of `sameAs`, however, can cause problems, when integrating information as discussed by Ding et al [7] and others [26, 25, 15]. We are currently using weaker predicates, `coref` and `notCoref` to represent that two instances are or are not thought to be coreferential. For coreferential instances, we can merge their descriptions for some, but not all, uses.

We may decide that two FOAF agents refer to the same object in the world, but their descriptions might be incompatible and asserting that they are equivalent with `sameAs` could lead to contradictions. One source of problems is that the descriptions could have both been true but at different times. Simply concatenating their triples is dangerous. Ding et al. present an example of two FOAF profiles for Li Ding. One hosted at Stanford was accurate when it was published several years ago, but some facts have changed since then. A more recent FOAF profile indicates that he is now working at RPI and holds a job title of Research Scientist. Each profile uses a unique URI to identify the person Li Ding, and it is reasonable to declare the two URIs refer to the same person. However, if we connect the two URIs using `sameAs`, an OWL reasoner can infer that Li Ding holds the position of Research Scientist at Stanford, which has never been the case. Another problem is that sources might differ in their beliefs about the world, such as the birthplace of the 44th president of the United States.

Figure 1 gives some axioms in N3 for the `coref` and `notCoref` properties. The `coref` property is transitive and symmetric and has `sameAs` as a sub-property. `notCoref` is symmetric, but not transitive and has `owl:differentFrom` as a sub-property. The first rule states that if two instances, a and b, are not coreferent, then every instance coreferent with a is `:notCoref` with every instance coreferent with b. The second rule, which is really a heuristic, states that if a knows b, then they are assumed to be distinct individuals and thus `:notCoref`. Note that `owl:sameAs` implies `coref` and `owl:differentFrom` implies `notCoref`, so reasoners that can derive `sameAs` and `differentFrom` properties will also contribute to computing coreference relations.

3 Background and Related Work

The problem of identifying co-reference entities is common in many contexts, including databases, bibliographic collections, and Semantic Web graphs. The earliest applications dealt with linking records for people in databases of significant life events, such as birth, marriage, and death records [10]. Fellegi and Sunter's 1969 paper [12] provided an early formal model for matching database records which represented identical persons, objects or events. Elmagarmid et al. [11] is a recent survey of record linking in database systems.

Name disambiguation in bibliographic databases has also been studied. Citations are rich with names – for people, departments, institutions, journals, conferences, publishers etc. Complicating the matching process is the fact that these are often abbreviated using many inconsistent forms. Han et al. [16] describe supervised learning approaches for name disambiguation in citations used in the CiteSeer system. Singla and Domingos [29] apply machine learning and probabilistic logic to a similar problem.

The traditional approach on the Semantic Web is the process of 'smushing' [21] FOAF instances – combining the profiles. 'Smushing' FOAF profiles can bring together information from various sources that are determined to represent the same 'person'. One can choose to rely solely on the presence of owl:sameAs, as this property is meant to link individuals [2]. However, relying on this property alone to 'smush' data is not effective, as its presence is not always found and it can also be represented inaccurately. There are multiple techniques used to both identify co-referent FOAF profiles and that perform some type of 'smushing' [28, 27, 18].

OWL's InverseFunctional property class (IFP) can help provide a way to recognize co-referent profiles, but it does not offer a complete solution. The FOAF vocabulary defines, for example, `foaf:homepage` and `foaf:mbox` to be inverse functions, providing strong evidence that two FOAF agents are co-referent if they share an identical value for either of those properties. However, with the popularity of social networking sites that support FOAF extraction of profiles, extracted FOAF profiles do not always include such an inverse functional property and sometimes these properties can be misused.

This was discovered by previous work [28, 18, 19]. For example, [19] describes how a list of FOAF profiles produced by exporters with empty `foaf:mbox` values all contained duplicate `foaf:mbox_sha1sum` values. In [18], it was discovered that a large portion of FOAF profiles which contained `foaf:weblog` property, contained a duplicate value for each profile, representing the community web logs. In [28], `foaf:weblog` and `foaf:homepage` in particular were found to be representing collective sites. In our previous work [30], we have found `foaf:homepage` values representing community web sites and we have also found FOAF profile with the absence of inverse functional properties altogether. It is not uncommon for several people to share a common value that is contained within an inverse functional property.

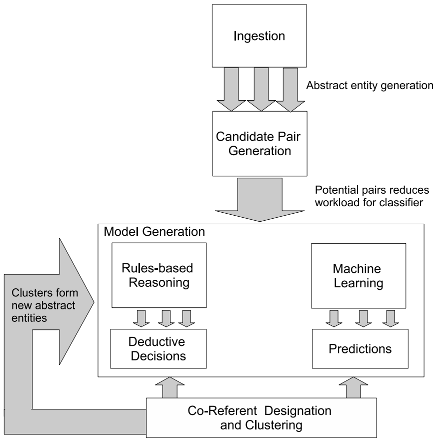

Ingestion
Abstract entity generation
Candidate Pair Generation
Potential pairs reduces workload for classifier
Model Generation
Rules-based Reasoning
Machine Learning
Deductive Decisions
Predictions
Clusters form new abstract entities
Co-Referent Designation and Clustering

Fig. 2: Co-Referent System Architecture

4 Architecture and Methodology

Our overall approach is similar to the ones used in other co-reference problems, such as [31, 24]. Given a collection of FOAF instances to compare, we would like to cluster them into sets that we believe refer to the same person in the world. This process is divided into five stages: (i) generating candidate pairs, (ii) generating the rules-based model, (iii) classification, (iv) designating pairs as co-referent or not, and (v) creating clusters. Figure 2 shows a high level architecture of our system. Entities are extracted from FOAF profiles and are the basis for the system. Entities can also be represented as a cluster of previously evaluated entities. Co-reference is determined by evaluation of both rules-based and vector models.

4.1 Methodology

We parse FOAF profiles by extracting triples from the associated URL and build an entity table based on extracted persons. For each entity found in the FOAF profile, a new entity is created in the database. When an entity is defined by a `foaf:knows` graph, our system uses any referring URL for that entity and attempts to parse FOAF data described by that URL. In a number of examples, we acquire more information about an entity by retrieving their FOAF profile. Our ensemble approach builds both a rules-based model that consists of results generated by logical rules and a vector classification model. If co-reference cannot be determined by the rules-based model, the prediction established by the vector model is then used. Co-referent pairs are part of larger clusters that are also used in the system to potentially discover other co-referent pairs. When we cluster our entities we use results from the rules-based model as a possible entity elimination from the cluster due to a logical result that indicates a pair within the cluster is known not to be co-referent.

```
{?p a owl:IFP. ?a ?p ?x. ?b ?p ?x) => {?a :coref ?b}
{?p a owl:FP . ?a ?p ?x. ?a ?p ?y.) => { ?x :coref ?y }
{?a foaf:knows ?b. ?a foaf:knows ?c. ?b neq ?c} => {?b :notCoref ?c}
```

Fig. 3: These N3 logic rules use FP and IFP properties and a special heuristic to draw conclusions about co-reference relations.

FOAF profiles were obtained from URLs extracted from Swoogle [6]. When retrieving documents based on the Swoogle listing, an attempt is made to retrieve the latest version of the document and if the latest version is no longer accessible we retrieve the cached version from the Swoogle database. We also used URLS extracted from tests conducted in previous work [30].

4.2 Candidate Designation

Given a potentially large collection of N FOAF instances we could proceed by testing each of the $O(N^2)$ possible pairs to see which are co-referent. Since the vast majority of the pairs will not be matched and the co-reference test will be relatively expensive, we start by filtering the possible pairs to produce a smaller set of candidates using a simple string matching heuristic test for each pair. The result is a set of FOAF instance pairs that can be used for both training and generating test sets that will be run through the classifier in step two.

A potential match is calculated based on common properties. For each pair of FOAF instances, an exact match is attempted for each property. If the exact match returns a false match for all properties then a partial match is attempted. If no partial matches exist, we attempt a simple cross-property match. For each type of match potential, if any single property pair returns true, we include the pair as a candidate. By performing this step we reduce the number of potential matches per URL which has improved total running time.

4.3 Rules-based Model

Our rules-based component consists of rules that can conclude that a pair of FOAF instances are co-referent or not-coreferent. Of course, for most pairs, neither conclusion can be drawn. If the rules conclude that both are true, then this inconsistency results in neither conclusion being used.

Some of the rules implement the semantics of OWL given the axioms in Figure 1, i.e., `owl:sameAs` implies `:coref` and `owl:differentFrom` implies `:notcoref`. This is supported by rules that use functional and inverse function properties, as shown in Figure 3.

We also use a heuristic rule that all individuals in a FOAF agent's `knows network` are assumed to be distinct, i.e., not co-referent. Note that the rule is applied to a graph that is extracted from a single document without prior processing and thus applies to explicitly asserted foaf:knows relations.

4.4 Vector Model

The co-referent classifier predicts whether two FOAF instances are co-referent. We use a Support Vector Machine (SVM) for classifying our data. We used SVMLight [22] with a linear kernel and standard parameters to build our model for classification. The classification model and predictions are captured for co-referent processing. Potential pairs are evaluated using a number of features based on FOAF property comparisons.

Feature Set Property-specific features include the following types:

- Inverse functional properties as a boolean feature
- Two different distance measures of properties common to both instances
- More complex distance measures, which might include unpacking semantic information (e.g., the geographical distance between two geotags) and resolving entity mentions (e.g., Baltimore) to linked data nodes
- Partial analysis of the graphs centered on the instances, such as the immediate (one-hop) social networks formed by `foaf:knows` properties

Distance metrics were calculated using the `Levenshtein distance` [23] method and Dice's Coefficient [5].

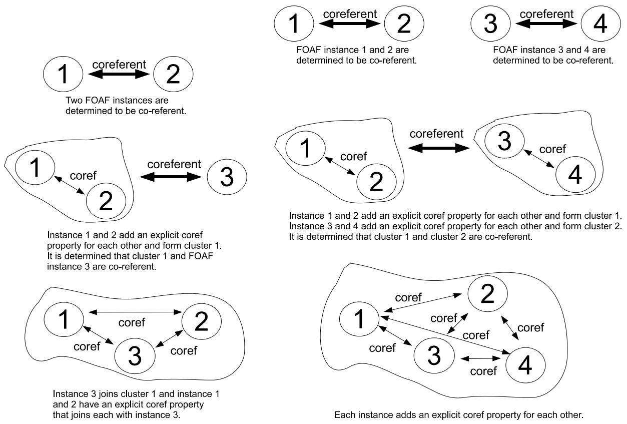
$$Dice'sCoefficient : \frac{(2 * Numberofcharacterbigramsinbothstrings)}{(Numberofbigramsinstring1 + Numberofbigramsinstring2)} \tag{1}$$

Simple Property Matching Distance. A simple property match is when a single property matches within the two FOAF instances being evaluated. For example, `foaf:name` matches in both instances.

Partial Property Matching Distance. In some cases, a property has a subpart which represents uniqueness that can be used as a distinguishing string to be matched to a subpart of the same property in a different instance. For example, part of the `foaf:weblog` property offers a partial match.

Cross-Property Matching Distance. In some cases, either a full property or a subpart of a property can be used to match a different property in another FOAF instance. In some of our gathered FOAF instances we discovered properties that were commonly cross-matched. For example, a `foaf:name` string part would correspond to a `foaf:nick`.

Training We automatically label training and tests using a heuristic. Training and tests must be manually inspected and evaluated for correctness. What we have seen is over a 70% accuracy in the automatic labeling heuristic. This has reduced the time it takes us to generate our training and test sets.



Two FOAF instances are determined to be co-referent.

Instance 1 and 2 add an explicit coref property for each other and form cluster 1. It is determined that cluster 1 and FOAF instance 3 are co-referent.

Instance 3 joins cluster 1 and instance 1 and 2 have an explicit coref property that joins each with instance 3.

FOAF instance 1 and 2 are determined to be co-referent.

FOAF instance 3 and 4 are determined to be co-referent.

Instance 1 and 2 add an explicit coref property for each other and form cluster 1.
Instance 3 and 4 add an explicit coref property for each other and form cluster 2.
It is determined that cluster 1 and cluster 2 are co-referent.

Each instance adds an explicit coref property for each other.

Fig. 4: Instance matching can result in new cluster formation and cluster matching can result in merged clusters.

4.5 Clustering

Once co-referent pairs are designated we cluster our pairs in such a way that the cluster is a representation of all instances of a particular FOAF agent. Clusters can grow over time as the amount of information used during the pair evaluation increases. We use a greedy process for clustering foaf individuals that begins by putting each in a singleton coreference set. A merging process continues as long as two candidate sets are judged to be similar enough to be merged into a new one that replaces its ancestors and stops when there are no pairs that can be merged. In this figure, the four foaf individuals end up in two coreference sets. When a FOAF pair is designated as co-referent this forms a cluster. As clusters begin to form in the system with multiple iterations, the co-referent pairing can be in the form of a FOAF instance and a cluster, two FOAF instances, and two FOAF clusters as depicted in Figure 4.

5 Experiments and Results

We ran two experiments, the first resulting in about 50,000 triples with over 500 entity mentions. Approximately 600 classes were used for training. The second had about 250,000 triples with over 3500 entity mentions. The classification training set consisted of over 1800 classes. For experiment two the distribution of URLs is conveyed in Figure 5. We conducted a 10-fold cross-validation with results shown in Table 2.

5.1 Results

For experiment two we show in Table 1 that we only saw the inverse functional property rule result in a number of positive co-referent cases. The majority

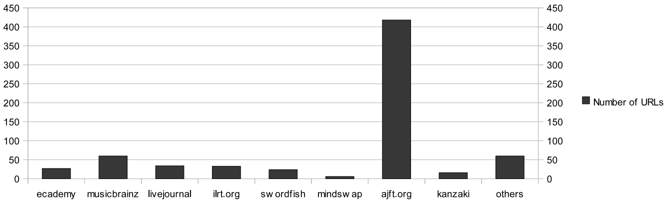


Fig. 5: Experiment Two URL Distributions

Table 1: Rules-based Model Results

E2 Number of pairs	Rule	Rule Conclusion
9138326	differentFrom	Undetermined
47184	inverse functional	Undetermined
2402	inverse functional	Co-referent
8687410	knows graph	Undetermined
9138326	sameAs	Undetermined
1047874	knows graph	Not Co-referent

Table 2: 10-Fold Cross-Validation Classification Test

	TP Rate	FP Rate	Precision	Recall	F-Measure
E1	0.933	0.267	0.93	0.933	0.93
E2	0.959	0.128	0.958	0.959	0.958

of the rules resulted in an undetermined state. As expected, the `foaf:knows` rule returned a number of pairs that resulted in a non-co-referent state. For experiment one 900 pairs were designated as a non-match and the majority of the rules returned an undetermined result. Table 2 shows that our classification step is likely predicting accurately co-referent and non-co-referent pairs.

During our E2 clustering phase, the first phase of clustering resulted in a 90% accuracy. The error occurred in pairs that should have been clustered but were not. A second round of clustering did not yield any new relationship pairs among instances but cluster to cluster pairing did occur.

5.2 Evaluation

The results of the above experiments are encouraging however since our new approach actually retrieves additional FOAF profiles based on members defined within the `foaf:knows` graphs, we quickly reach large numbers of 'entities' with the average number of entities known by a person between 50-100 people. This can produce a selection of entities that is tightly linked which can have two effects on the system. It can reduce diversity of analyzed data and it can produce a number of entities that are likely representing the same person. Future iterations will include larger, more diverse sets of data with a diversity filter used to select pairs that span a number of domains. Our two experiments explored using different url distributions, however, 10-fold cross-validation results were close in measures. When choosing to add new classes to a cluster a threshold is used as a way to reduce errors. This threshold will require additional testing to determine an appropriate setting.

6 Conclusions and Future Work

We have described an approach to predicting coreferent pairs of FOAF instances that uses a small set of rules and a classifier developed by supervised machine learning process. The descriptions of coreferent pairs are merged to create a new description that is then re-evaluated to find additional descriptions judged to be coreferent.

We have been working with FOAF data as an instance of a larger problem: automatically linking RDF instances based on their descriptions. Making headway on this problem will allow us to more easily add data to the growing linked data cloud [3]. Our machine learning approach, system architecture and many of our techniques should also apply to non-FOAF data equally well.

The FOAF co-referent problem described here is also a common problem in non-FOAF domains. Our approach can be abstracted and applied to other domains. In particular, instance matching [13] among ontologies is a domain that could benefit from such a co-referent solution. Entity clustering [1] is also another domain which could benefit from our co-referent solution.

Future work will include exploiting additional properties within the instance that are not of the FOAF vocabulary (e.g., sioc [4]) and using these properties to provide additional evidence as to whether a pair of FOAF instances represent the same individual or not. A portion of our collected FOAF documents had non-FOAF vocabularies that offered additional information such as 'author'. By exploiting these additional properties, we could increase accuracy particularly when a FOAF property is absent and the non-FOAF property offers the same meaning.

As highlighted in our introduction, inverse functional properties can be implemented incorrectly. We account for this type inaccuracy in our classification method, however we also plan to account for this in our rules-based model in future revisions of the system.

Many properties asserted about FOAF instances have string values that refer to entities. Examples from the core FOAF vocabulary are `foaf:Organization` and `foaf:fundedBy`. We would like to recognize that two strings refer to the same entity when their values are different but known aliases or alternate names. Luckily, for many entities, it is easy to generate lists of known aliases drawing on resources such as Gazetteers, Wikipedia and Freebase. We have developed lists of known aliases for organizations and places from data extracted from Wikipedia and Freebase, including aliases for about 270K places and 50K organizations. The current system does not yet exploit these lists but we plan to do so in the next version, probably as an additional string matching metric, e.g., the two instances have a property whose values differ but are in members of a known set of aliases.

References

1. Artiles, J., Gonzalo, J., Sekine, S.: Weps 2 evaluation campaign: Overview of the web people search clustering task. In: 18th Int. World Wide Web Conf. Madrid, Spain (April 2009)
2. Bechhofer, S., Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P., L.Stein: Owl web ontology language reference w3c recommendation 10 february 2004. http://www.w3.org/TR/owl-ref/ (2004)
3. Bizer, C.: The emerging web of linked data. IEEE Intelligent Systems 24(5), 87–92 (2009)
4. Bojārs, U., Breslin, J., Peristeras, V., Tummarello, G., Decker, S.: Interlinking the social web with semantics. IEEE Intelligent Systems 23(3), 29–40 (2008)
5. Christen, P.: A comparison of personal name matching: Techniques and practical issues. In: Proceedings of the Second International Workshop on Mining Complex Data. IEEE (2006)
6. Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V.C., , Sachs, J.: Swoogle: A search and metadata engine for the semantic web. In: Proc. 13th ACM Conf. on Information and Knowledge Management (2004)
7. Ding, L., Shinavier, J., Finin, T., McGuinness, D.L.: OWL:sameAs and linked data: an empirical study. In: Proc. 2nd Web Science Conf. (April 2010)
8. Ding, L., Zhou, L., Finin, T., Joshi, A.: How the Semantic Web is Being Used:An Analysis of FOAF Documents. In: Proc. 38th Int. Conf. on System Sciences (January 2005)
9. Dredze, M., McNamee, P., Rao, D., Gerber, A., Finin, T.: Entity disambiguation for knowledge base population. In: Proc. 23rd Int. Conf. on Computational Linguistics (August 2010)
10. Dunn, H.: Record linkage. American Journal of Public Health 36(12), 1412 (1946)
11. Elmagarmid, A., Ipeirotis, P., Verykios, V.: Duplicate record detection: A survey. IEEE Transactions on knowledge and data engineering pp. 1–16 (2007)
12. Fellegi, I., Sunter, A.: A theory for record linkage. Journal of the American Statistical Association 64(328), 1183–1210 (1969)
13. Ferrara, A., Lorusso, D., Montanelli, S., Varese, G.: Towards a benchmark for instance matching. In: Int. Workshop on Ontology Matching, volume 431, 2008 (2008)
14. Golbeck, J., Rothstein, M.: Linking social networks on the web with FOAF. In: Proc. 17th Int. World Wide Web Conf. (April 2008)
15. Halpin, H., Hayes, P.J.: When owl:sameas isnt the same: An analysis of identity links on the Semantic Web. In: Proc. 2010 Int. Workshop on Linked Data on the Web (April 2010)
16. Han, H., Giles, L., Zha, H., Li, C., Tsioutsiouliklis, K.: Two supervised learning approaches for name disambiguation in author citations. In: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries. pp. 296–305 (2004)
17. Harth, A., Gassert, H.: On searching and displaying RDF data from the web. In: Proc. Demos and Posters, 2nd European Semantic Web Conf. Heraklion, GR (May 2005)

18. Hogan, A., Harth, A., Decker, S.: Performing object consolidation on the semantic web data graph. In: Proc. I3: Identity, Identifiers, Identification. Workshop at 16th Int. World Wide Web Conf. (February 2007)
19. The god entity. http://blog.aidanhogan.com/2008/10/god-entity.html (2008), accessed January 2010
20. Billion triple challenge 2009 dataset. http://vmlion25.deri.ie/ (2009), accessed November 2010
21. Foaf-project.org definition of smushing. http://wiki.foaf-project.org/w/Smushing (2010), accessed January 2010
22. Joachims, T.: SVMLight: Support Vector Machine (1999), university of Dortmund, http://svmlight.joachims.org/
23. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10, 707–710 (1966)
24. Mayfield, J., Alexander, D., B, B.D., Eisner, J., Elsayed, T., Finin, T.: Cross-document coreference resolution: A key technology for learning by reading. In: Proc. AAAI Spring Symposium on Learning by Reading and Learning to Read. AAAI (March 2009)
25. McCusker, J., McGuinness, D.L.: owl:sameas considered harmful to provenance. In: Proc. ISCB Conf. on Semantics in Healthcare and Life Sciences (Feburary 2010)
26. Passant, A.: :me owl:sameas flickr:33669349@n00. In: Proc. 1st Int. Workshop on Linked Data on the Web (April 2008)
27. Price, S., Rawles, S., Flach, P.: Estimating whether partial FOAF descriptions describe the same individual. In: Proc. Workshop on Friend of a Friend, Social Networking and the Semantic Web (September 2004)
28. Shi, L., Berrueta, D., Fernandez, S., Polo, L., Fernandez, S.: Smushing rdf instances: are alice and bob the same open source developer? In: Proc. 3rd Expert Finder workshop on Personal Identification and Collaborations: Knowledge Mediation and Extraction, 7th Int. Semantic Web Conf. (November 2008)
29. Singla, P., Domingos, P.: Entity resolution with markov logic. In: 6th Int. Conf. on Data Mining, 2006. ICDM'06. pp. 572–582 (2006)
30. Sleeman, J., Finin, T.: A Machine Learning Approach to Linking FOAF Instances. In: Spring Symposium on Linked Data Meets AI. AAAI (January 2010)
31. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Silk - a link discovery framework for the web of data. In: Proc. 2nd Workshop on Linked Data on the Web. Madrid, Spain (April 2009)