Learning Co-reference Relations for FOAF Instances*

Jennifer Sleeman and Tim Finin
University of Maryland, Baltimore County
Baltimore. MD 21250 USA

Abstract. FOAF is widely used on the Web to describe people, groups and organizations and their properties. Since FOAF does not require unique IDs, it is often unclear when two FOAF instances are co-referent, i.e., denote the same entity in the world. We describe a prototype system that identifies sets of co-referent FOAF instances using logical constraints (e.g., IFPs), strong heuristics (e.g., FOAF agents described in the same file are not co-referent), and a Support Vector Machine (SVM) generated classifier.

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. 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.

A traditional approach for identifying co-reference entities on the Semantic Web is the process of smushing [4] FOAF instances combining the information from various sources that are determined to represent the same person. One can choose to rely solely on the presence of owl:sameAs, however, 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 as mentioned in our previous work [5]. We describe a hybrid approach for deciding when RDF descriptions of two FOAF agents are likely to be co-referent that combines rules and an SVM-based classifier.

While the `owl:sameAs` relation is typically used to assert that two FOAF instances refer to the same individual, this can lead to unwarranted and problematic inferences [2, 3]. For this reason, we use the 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. Figure 1 gives some axioms in N3 for the `coref` and `notCoref` properties. The `coref` property is transitive and symmetric and has

* Partial support for this research was provided by NSF award 0910838 and the Johns Hopkins University Human Language Technology Center of Excellence.

```
:coref a owl:TransitiveProperty, 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: Definitions of the :coref and :notCoref properties uses instead of owl:sameAs.

`owl: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, 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.

2 Approach

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 a 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. We describe our approach below as defined in [6].

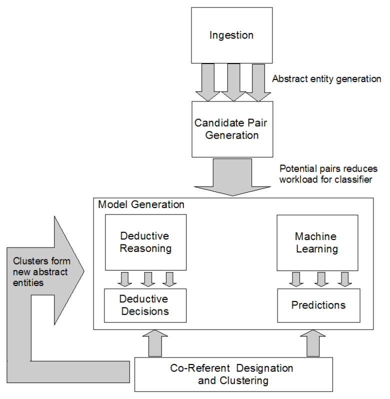
Ingestion
Abstract entity generation
Candidate Pair Generation
Potential pairs reduces workload for classifier
Model Generation
Deductive Reasoning
Machine Learning
Deductive Decisions
Predictions
Clusters form new abstract entities
Co-Referent Designation and Clustering

Fig. 2: Our system architecture starts by selecting pairs of foaf instances to compare and applies both rule-based reasoning and an SVM-based classifier to determine likely co-reference relations that are then use to form clusters.

Pair Generation. With a potentially large collection of 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.

Rule-based Model and Classification. Filtered pairs are evaluated by rules that provide a result that indicates whether the pair is or is not co-referent. Rules can include properties such as owl:sameAs, property attributes such as

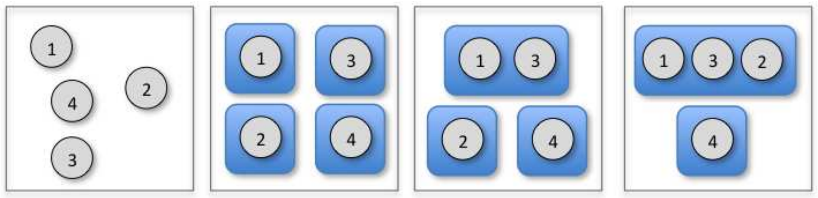

Fig. 3: After assigning each foaf individual to a singleton coreference set, we use a greedy algorithm to merge the sets that are judged to be similar.

inverse functional properties and rules deduced from the data itself. For example, persons defined by a persons knows graph are likely not to be co-referent with that person. The co-referent classifier takes a pair of FOAF instances and decides if they are co-referent or not. Property-specific features used by the classifier include distance measures of properties common to both instances, inverse functional properties, more complex distance measures, which might include unpacking semantic information (e.g., the geographical distance between to geotags) and resolving entity mentions (e.g., Baltimore) to linked data nodes, and partial analysis of the graphs centered on the instances, such as the immediate (one-hop) social networks formed by foaf:knows properties. Refer to [6] for a more detailed discussion.

Co-referent Designation. The results of the classifier are used by the co-referent process when determining co-referent pairs. The rules-based model is used to determine if a pair of FOAF instances are co-referent. If the results of the rules-based model are indeterminate, then the prediction generated by the classifier is used. Pairs that are designated as co-referent form a new cluster. Clusters are then processed by the system in addition to pairs.

Clustering. New co-referent pairs can be considered as a graph where the nodes are the original set of FOAF instances and an edge exists between two nodes that are a candidate pair as determined by the co-referent processing. Co-referent triples become part of larger cluster groups and are used for future pair matching.

Figure 3 depicts the greedy process for four foaf individuals. We begin 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.

3 Evaluation

We ran two experiments, the first experiment resulted in about 50,000 triples with over 500 entity mentions. We applied the deductive rules which resulted in 900 pairs that were designated as a non-match and the majority was undetermined. The classification portion of this process consisted of 600 pairs used for training and 3 tests consisting of 200 pairs each. The third test also included a single post-clustering run.

	True Positive Rate	False Positive Rate	Precision	Recall	F-Measure
E1	0.933	0.267	0.930	0.933	0.930
E2	0.959	0.128	0.958	0.959	0.958

Table 1: The results of a 10-fold cross-validation classification test show good results fir both precision and recall.

Our latest experiment contained about 250,000 triples with over 3500 entity mentions. We applied the rules which resulted in positive co-referent cases based on the inverse functional property. The majority 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. The classification training set consisted of over 1800 classes. We conducted a 10-fold cross-validation with results conveyed in table 1. Table 1 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. Additional tests and evaluations are outlined in [6].

4 Conclusions and Future Work

We have described an approach to predicting coreferent pairs of FOAF instances that uses a small set of rules, a classifier developed by supervised machine learning process and clustering co-referent pairs. We have been working with FOAF data as an instance of a larger problem: automatically linking RDF instances based on their descriptions.

References

1. Artiles, J., Gonzalo, J., Sekine, S.: Weps 2 evaluation campaign: Overview of the web people search clustering task. In: 18th WWW Conf. Madrid (2009)
2. 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)
3. Halpin, H., Hayes, P., McCusker, J., McGuinness, D., Thompson, H.: When owl:sameas isn't the same: An analysis of identity in linked data. In: Proc. 9th Int. Semantic Web Conf. (2010)
4. Foaf-project.org definition of smushing. http://wiki.foaf-project.org/w/Smushing (2010), accessed January 2010
5. Sleeman, J., Finin, T.: A Machine Learning Approach to Linking FOAF Instances. In: Spring Symposium on Linked Data Meets AI. AAAI (January 2010)
6. Sleeman, J., Finin, T.: Computing FOAF Co-reference Relations with Deduction and Machine Learning. In: Proceedings of the Third International Workshop on Social Data on the Web (November 2010)