Entity Type Recognition for Heterogeneous Semantic Graphs

Jennifer Sleeman, Tim Finin, Anupam Joshi

■ *We describe an approach for identifying fine-grained entity types in heterogeneous data graphs that is effective for unstructured data or when the underlying ontologies or semantic schemas are unknown. Identifying fine-grained entity types, rather than a few high-level types, supports coreference resolution in heterogeneous graphs by reducing the number of possible coreference relations that must be considered. Big data problems that involve integrating data from multiple sources can benefit from our approach when the data's ontologies are unknown, inaccessible, or semantically trivial. For such cases, we use supervised machine learning to map entity attributes and relations to a known set of attributes and relations from appropriate background knowledge bases to predict instance entity types. We evaluated this approach in experiments on data from DBpedia, Freebase, and Arnetminer using DBpedia as the background knowledge base.*

Big data is often characterized as data exhibiting the four *v*'s: volume, velocity, variety, and veracity (McAfee and Brynjolfsson 2012). Annotating data elements with semantic representations can help manage two of them: variety and veracity. Often what this entails is the integration of data from different sources whose schemas are unknown, largely syntactic or very weak, and impossible or difficult to integrate. Automatically linking data to common semantic models enhances integration and interoperability, especially if the semantic models support reasoning. Semantic annotations can also help ensure veracity by detecting violations of semantic constraints and allowing the application of semantically grounded statistical models.

Often attempts to solve big data integration problems are addressed by means of schema mappings, record linkage, and data fusion (Dong and Srivastava 2013). In this regard, coreference resolution becomes a necessity, and often traditional approaches are not designed to solve these types of integration problems because they do not account for integrating data from multiple, and often schemaless, sources.

 ISSN 0738-4602

Finding entity mentions and identifying their types are important steps in many data analysis tasks including processing structured tables and logs, semistructured graphs, and unstructured text. The results directly support subsequent tasks, such as record linkage, question answering, coreference resolution, and ontology mapping. For example, identifying medical conditions based on symptoms requires integrating medical records from a particular medical practice, known medical conditions from a trusted medical knowledge base, and possibly results from a question-answering system, all of which may or may not have some underlying ontological structure. Entity resolution can support the identification of medical conditions by identifying entities and matching entities that are likely to be coreferent. In this example medical conditions are the entities and their associated symptoms are properties. Identifying a match between a new instance, that is, the patient's list of symptoms and a known medical condition, is an example of identifying whether they corefer to each other.

Most natural language analysis systems use linguistic evidence and context to identify entity mention strings and predict their type, typically chosen from a relatively small number of high-level possibilities, such as person, place, and organization, perhaps augmented with additional application-specific types. For many forms of structured and semistructured data (for example, tables, logs, XML, JSON), schemas may be unavailable or overly simple and semantically weak. When working with semantic graphs, if an ontology is present, the ontology may explicitly define the entity types. However, in situations where semantic graphs are not defined ontologically or when the data itself does not sufficiently use the ontology, the types are harder to identify. Linguistic analysis cannot be used in this case since it relies upon the structure of the sentence to understand the components of the sentence, which is not present when data is represented as semantic graphs or a similar representation.

When performing coreference resolution over RDF data or a similar formalism, the entity types can be explicitly given in a familiar ontology and their properties understood, enabling systems to reason about instance equality (Ferrara et al. 2008, Seddiqui and Aono 2010, Araujo et al. 2011). When this is not the case, that is, when the ontologies are not accessible or not understood or several nonaligned ontologies are used, direct reasoning about instance equality is difficult, if not impossible. We believe that this situation will be common in many big data applications, where semantic annotations may be relatively simple and where entities and their schemas can have very different representations.

Contribution

In this work we address identifying fine-grained entity types as a prefilter for algorithms that determine which entities in a given heterogeneous data set corefer. For example, in the medical domain, for the condition cancer, we would not only identify high-level cancer types such as carcinoma, sarcoma, leukemia, lymphoma, and myeloma, and central nervous system cancers.[1] Rather we would also identify more fine-grained types such as breast cancer and bladder cancer. Our preliminary experiments show a one-level-deep identification and our ongoing work will include experiments that show identification of types at various levels.

Among linguistic-based entity recognition approaches, most current research does not address fine-grained entity type identification. However, it is often useful to have a more fine-grained understanding of entity types to support efforts in heterogeneous data integration. We use a linear approach for compatibility with big data architectures. With large knowledge bases there could exist thousands of entity types; it would be inefficient and unnecessary to evaluate an instance against each entity type. By means of information theory and high potential predicate filtering, we associate each new instance with a set of high potential candidate entity types, resulting in a significant reduction in the number of classifications. The results of our mapping approach from a single instance to a set of entity types allows us to cluster candidate coreferent instances by entity types.

Background

Semantic graphs are graph-based representations that typically are represented as triples. The resource description framework (RDF) is commonly used to describe resources on the web and provides a graph-based representation (Beckett 2004, Brickley and Guha 2004).

An RDF graph is a set of triples, each of which has a subject, predicate, and object. For example, the DBpedia resource Monaco, Monaco would be the subject, an attribute such as areaTotal would be the predicate, and a literal value 1.98 for areaTotal would be the object. A triple T is represented by a subject s, a predicate p, and an object o, such that $T(s, p, o)$, where o is a node, s is a node, and p defines the relationship between s and o by a URI. Given that a node can be a URI identifying the node, a literal, or blank, the following definitions apply: $s \in (\text{URI} \cup \text{Blank})$,$p \in (\text{URI})$ and $o \in (\text{URI} \cup \text{Blank} \cup \text{Literal})$ (Beckett 2004, Brickley and Guha 2004).

Linked open data (LOD) (Bizer 2009) enables one to make data publicly available and linked to known data sets. LOD attempts to address this problem of integrating heterogeneous data, which is an inherent

problem for big data (Bizer et al. 2012). However, linking to known data sets is a challenge, particularly when the data is heterogeneous. For example, one data set could represent an attribute using numeric values, whereas another might use string representations. In work by Nikolov and colleagues (Nikolov et al. 2009; Nikolov, Uren, and Motta 2010), they discuss the heterogeneity problem, how it relates to coreference resolution, and the need for LOD automatic entity linking. Araujo and colleagues (2011) also reference the need for an entity mapping solution that is domain independent.

Ontologies

An ontology can be thought of as a schema and provides a definition of the data, similar to an entity-relationship model (Euzenat and Shvaiko 2007). It includes a vocabulary of terms with specific meaning (Gomez-Perez, Fernandez-Lopez, and Corcho 2004). Ontologies play a critical role in the semantic web (Berners-Lee, Hendler, and Lassila 2001) and can be used to describe a domain. Typically ontologies use OWL (Bechhofer et al. 2004) or other languages as a representation. They define classes, instances, attributes, and relations (Euzenat and Shvaiko 2007). Often one will find instance data that is described by multiple ontologies in addition to RDF. Ontologies can be privately defined and not accessible to the general public or publicly defined. It is common to be exposed to instances described by ontologies that cannot be accessed and that require an alternative method to understand the data.

Comprehensive Knowledge Bases

The development of a comprehensive, general-purpose knowledge base has been a goal of AI researchers dating back to the CYC project (Lenat, Prakash, and Shepherd 1985) in the early 1980s. In the past five years, two important open knowledge bases come close to realizing CYC's vision: DBpedia and Freebase. DBpedia is a structured representation of Wikipedia (Auer et al. 2007). The DBpedia knowledge base provides classification for 3.22 million objects, which mainly consist of people, locations, organizations, diseases, species, and creative works.[2] Freebase is also a large, structured knowledge base (Bollacker et al. 2008) with a considerably larger number of topics than DBpedia.

Coreference Resolution

Coreference resolution is the task of determining which instances in a collection represent the same real-world entities. It tends to have an $O(n^2)$ complexity since each instance needs to be evaluated with every other instance. Various techniques have been developed to reduce the number of instances (McCallum, Nigam, and Ungar 2000; Mayfield et al. 2009; Sleeman and Finin 2010a; Rao, McNamee, and Dredze 2010; Singh et al. 2011; Uryupina et al. 2011; Song and Heflin 2011). In our previous work (Sleeman and Finin 2010b, 2010a) we also used filtering to reduce the number of candidates for the coreference resolution algorithm. We often processed data using ontologies that were not publicly available. Without an understanding of the ontologies used, it is often challenging to process data that uses those ontologies and could negatively affect accuracy.

Problem Definition

Definition 1.

> Given a set of instances *INST,* extracted from a set of heterogeneous sources SRC, that are not ontologically defined and not grammatically represented in a sentence, for each instance $inst_1 \ldots inst_m \in INST$, we wish to associate a set of entity types $ET_1 \ldots ET_m$.

Recognizing semantic graph entities is related to information-extraction entity recognition, the process of recognizing entities and their type (for example, a person, location, or organization) (Ratinov and Roth 2009, Nadeau and Sekine 2007). However, it does not require the entities to be grammatically defined in a sentence structure and it entails the recognition of fine-grained entities that would be harder to obtain from a typical information-extraction system.

Fundamental to our work is understanding the attributes and relations defined by the instance data. By classifying the attributes and relations, we relate unknown attributes and relations to known attributes and relations. We use this as a means for predicting entity types among heterogeneous semantic graphs.

A similar problem arises in work related to database interoperability (Nottleman and Straccia 2007; Berlin and Motro 2002; Do, Melnik, and Rahm 2003) and ontology matching (Albagli, Ben-Eliyahu-Zohary, and Shimony 2012; Mitra, Noy, and Jaiswal 2005). In both, integrating heterogeneous data drawn from different repositories with different schemas is difficult simply because it is hard to establish that an attribute or relation in one schema is the same (or nearly the same) as an attribute or relation in another (Jaiswal, Miller, and Mitra 2010).

LOD (Bizer 2009) has specifically addressed the issue of linking heterogeneous structured data in RDF to enable interoperability. In order to add an RDF data set to a LOD collection, we represent the information as RDF and then link its elements (classes, properties, and individuals) to known elements elsewhere in the collection. Though the LOD cloud collection has grown significantly, the total number of linked data sets is still relatively small (about 300)[3] and the degree of interlinking often modest. Given the amount of data both available online and not available online, this number indicates that most repositories are still not linked to significant LOD collections and it is likely that these repositories use

custom schemas.

In many popular knowledge bases such as DBpedia, entity types are not always present in the data, even with a sufficiently defined ontology present. Recent research by Paulheim and Bizer (2013) found DBpedia types were only 63 percent complete with 2.7 million missing types.

Shvaiko and Euzenat (2008) described challenges in ontology matching where one such challenge is missing context. Couple the absence of context with opaquely defined attributes and often ontologies are hard to align.

Related Work

The recent work by Paulheim and Bizer (2013) tackles the problem of identifying entity types absent in instance data by inferring types based on existing type definitions. They assign type probabilities to indicate the likelihood of the assertion and use these as weights to establish which relations provide the best evidence for the type assertion. Their approach differs from ours in that they use link analysis to develop their model whereas we do consider the links between graphs but we do not rely upon this alone. Rather we build a dictionarylike structure that we then try to map to evaluated instances. Paulheim and Bizer work with ontologies where type definitions exist; we specifically address the issue of identifying types when the ontologies are either not present or insufficiently defined.

Nikolov, Uren, and Motta (2010) describe the problem of mapping heterogeneous data where often "existing repositories use their own schemas." They discuss how this makes coreference resolution difficult, since similarity evaluation is harder to perform when attribute mappings are unclear. They take advantage of linked data and knowledge of relationships between instances to support schema-level mappings. However, if a repository is not linked to an appropriate LOD collection, this method is not feasible. We address this issue of custom schemas and their impact on coreference resolution by mapping attributes to a known set of attributes for various entity types.

Early work by Berlin and Motro (2002) addressed the problem of database mapping using machine learning. Their Automatch system used machine learning to build a classifier for schema matching using domain experts to map attributes to a common dictionary. The approach performed well, achieving performance exceeding 70 percent measured as the harmonic mean of the soundness and the completeness of the matching process. We build on this idea, using the dictionary mapping concept generated from DBpedia through a process guided by information gain.

Work by Reeve and Han (2005) provides a survey related to semantic annotation that is more closely related to our work. They describe and benchmark methods designed for unstructured text complemented with the output of information-extraction tools to construct mappings. This differs from our approach in that we start from the graphs themselves without the raw text and information-extraction data and metadata. This is a key distinction since using the graphs alone can b more limiting. The benchmark compared various annotation tools using annotation recall and annotation precision, which we also will use to measure our entity typing performance.

Recent research by Suchanek, Abiteboul, and Senellart (2012) describes their approach, PARIS, for aligning ontologies. This work uses string equality and normalization measures and also takes the approach of only using positive evidence. Again our goal was to be domain independent, such that one could use a data set to build the dictionary of types they wish to recognize then apply our mapping process to map to these dictionaries. We use techniques more akin to traditional named entity recognition to perform the task. This distinguishes our work from much of the ontology mapping research.

Bootstrapping to a Well-Defined Knowledge Base

In order to assign entity types to new entity instances, we use a known knowledge base and build a model of this information. By bootstrapping to a known knowledge base, we ground unknown instance data to a known definition. For example, using the medical domain, if we wanted to identify different types of leukemia, our bootstrapping knowledge would have entities, properties, and relations defined that represent leukemia. If we process data that entails medical information regarding many types of cancers, we would map to our leukemia knowledge base to try to identify specific leukemia cancers. Since there are different types of leukemia, we would attempt to identify the unknown cancers with the types defined in our leukemia knowledge base.

Definition 2. Unknown to Known Type Map

Given a set of known entity types ET extracted from a well-defined knowledge base KB, we create a bootstrapped system that looks to identify unknown entity types based on ET. Each type $et_1 \ldots et_n \in ET$ is defined based on a set of predicates $EP_1 \ldots EP_n$, that is, attributes and relations.

We bootstrap to a large well-defined knowledge base to define our set of entity types. In our experiments we used DBpedia but our approach is flexible enough to work with any well-defined knowledge base. We used the DBpedia ontology itself to build a model of the entity types. The model includes equivalence relationships, hierarchical relationships, and pattern similarity relationships. We use this model

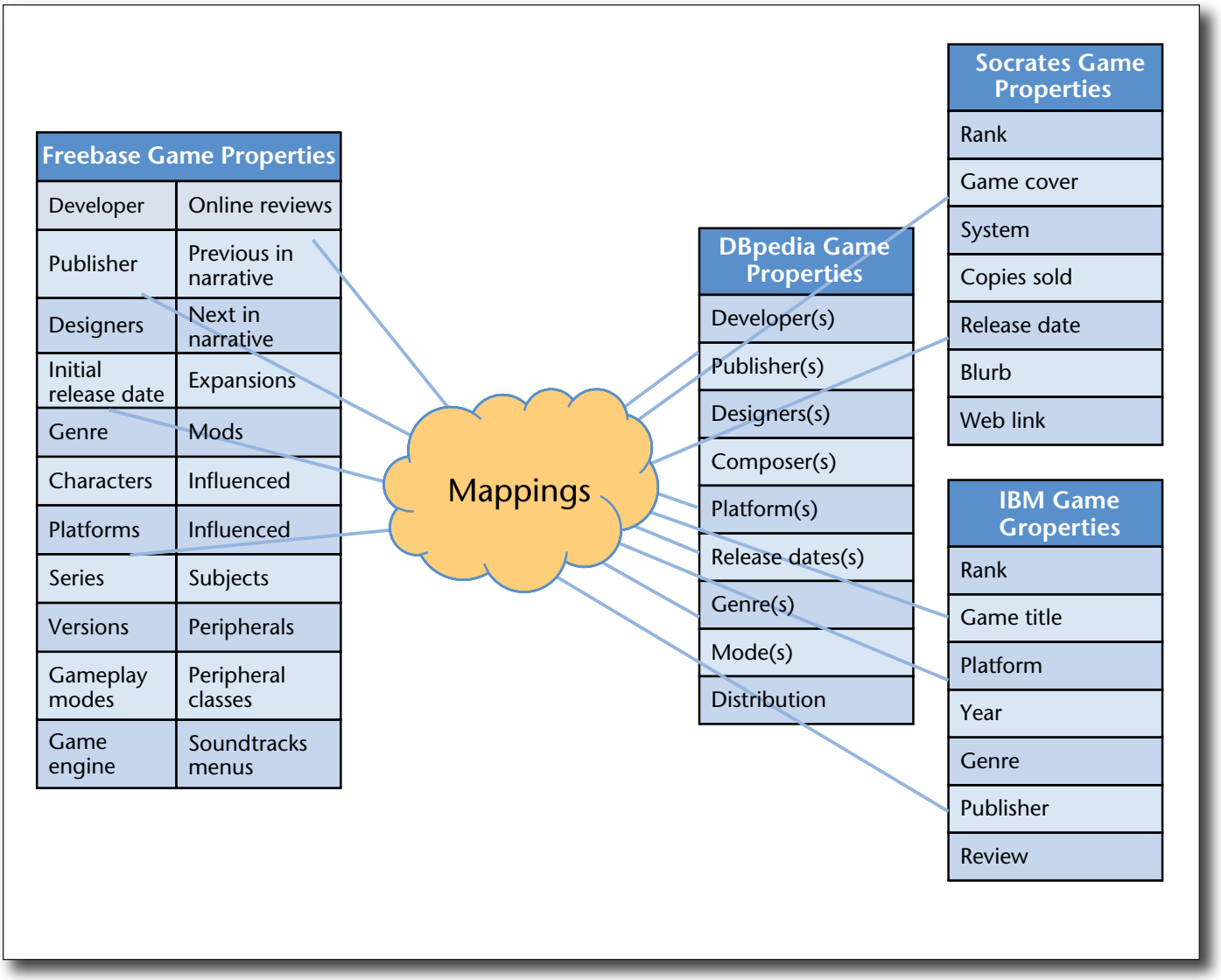

Figure 1. Mapping Entity Types Between Data Sources.

during evaluation to expand the number of potential entity types for a candidate and to aide in the prediction of entity types when an entity is not classified as any of the known types. Entities defined in DBpedia typically are associated with a number of entity types. This information allows us to infer types when information is not present.

Predicate Filtering

Fundamental to our work is mapping predicates associated with known types to predicates of unknown types. Figure 1 shows an example of mapping between video game definitions. However, evaluating a set of instance predicates with all possible predicates in the knowledge base is costly and unnecessary. Based on the model we define during bootstrapping, when we process a new instance we evaluate its predicates with high potential predicates in our model.

Definition 3. High Potential Predicates

Given an entity type knowledge base KB, with a set of entity types ET, where each type $et_1 \ldots et_n \in ET$ is defined by $EP_1 \ldots EP_n$. From this overall distribution of predicates, we define a subset of predicates *H P E P* and link *H P E P* to *ET*.

Definition 4. Instance to High Potential Predicate Link

Given a set of instances *INST,* each instance $inst_1 \ldots inst_m \in INST$ is defined by a set of predicates $IP_1 \ldots IP_m$, which is then evaluated against *H P E P*. Each $inst_1 \ldots inst_m \in INST$ is then linked with a set of high potential candidate entity types $CET_1 \ldots CET_m$.

Instance Predicate	High Potential Mappings
http://creativecommons.org/ns#attributionName	http://xmlns.com/foaf/0.1/name
http://rdf.freebase.com/ns/people.person.profession	http://dbpedia.org/ontology/president
	http://dbpedia.org/ontology/profession
http://rdf.freebase.com/ns/award.award_winner.awards_won	http://dbpedia.org/ontology/award
http://rdf.freebase.com/ns/people.person.spouse_s	http://dbpedia.org/ontology/spouse
http://rdf.freebase.com/ns/people.person.ethnicity	http://dbpedia.org/ontology/colour
http://rdf.freebase.com/ns/music.artist.track_contributions	http://dbpedia.org/ontology/genre
http://rdf.freebase.com/ns/people.person.gender	http://dbpedia.org/ontology/gender
http://rdf.freebase.com/ns/award.award_nominee.award_nominations	http://dbpedia.org/ontology/country
	http://dbpedia.org/ontology/award
http://rdf.freebase.com/ns/people.person.place_of_birth	http://dbpedia.org/ontology/position
	http://dbpedia.org/ontology/grades
	http://dbpedia.org/ontology/office
	http://dbpedia.org/ontology/president
	http://dbpedia.org/ontology/ranking
http://rdf.freebase.com/ns/type.object.name	http://xmlns.com/foaf/0.1/name
http://rdf.freebase.com/ns/people.person.nationality	http://dbpedia.org/ontology/country
	http://dbpedia.org/ontology/colour

Figure 2. Mapping Instance Predicates.

For example, a new instance with a set of predicates will result in the mappings in figure 2 based on predicate filter and predicate mapping.

Entity types associated with the mapped predicates are then candidates for type matching. We are able to evaluate a smaller selection of entity types without evaluating each new instance with every other instance. Ongoing work will quantitatively show the impact of our method on computation time. We use this approach as a prefilter to coreference resolution, reducing the number of instances that need to be evaluated without incurring a cost that is equal to the n^2 computation time cost of the coreference resolution algorithm. This prefiltering is beneficial to the coreference resolution algorithm because it partitions the instances into smaller clusters such that the instances within a cluster have a higher likelihood of being coreferent.

The evaluation of each instance with each potential entity type candidate results in the features used to build a supervised classification model. We perform the same mapping approach for unlabeled test data and classify these instances using the supervised model.

Feature Reduction and Information Gain

Information gain is one of the measures used to define the HPEP. It is also used when evaluating an instance with a candidate entity type. In order to create the mappings defined in figure 2, we start with a set of predicates that have high information gain. Using information gain we filter predicates that are to be mapped to instance attributes. Ongoing research will evaluate the effects of information gain thresholding as a means to filter predicates.

Given our set of types S and their set of predicates P, we calculate information gain and associate a weight for each predicate $\in P$.

$$Gain(S,P) = Entropy(S) - \sum_{v \in Values(P)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (1)$$

Where p is the probability of the value x_i.

$$Entropy(S) = -\sum_{i}^{N} p(x_i) log_2 p(x_i) \quad (2)$$

Mapping

The concept of mapping to a common set of attributes is similar to database mapping and ontology alignment research (Nottleman and Straccia 2007; Berlin and Motro 2002; Mitra, Noy, and Jaiswal 2005; Albagli, Ben-Eliyahu-Zohary, and Shimony 2012). A selection of this work is discussed in more detail in the Related Work section.

Our work with mapping instances to types is ongoing and critical to the accuracy of the classification. Our intent is to allow for a pluggable representation whereby one can define the set of mappers that would be most appropriate for the data to be processed. We use a distance mapper to measure the similarity of predicate labels. We currently use a Levenshtein (Levenshtein 1966) distance measure. We use a synonym mapper to measure similarity of predicate label synonyms. To obtain synonyms we use WordNet (Miller 1995) and measure similarity between sets of synonyms using Jaccard's similarity.

$$Jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (3)$$

We measure similarity of predicate values by using Jaccard's similarity and measure their frequency of occurrence and common occurrences. We also evaluate predicate values by detecting known patterns using regular expressions. For example, an email address is a commonly occurring pattern. The results of the mappers become the features we use for classification. Ongoing work will measure statistical differences of predicate values and other properties.

Classifying Entity Types

We use a support vector machine (SVM) (Joachims 2002) to develop a model for each entity type. By using one classifier per entity type, we address two important issues: we are able to use a linear classifier for this problem and we are able to horizontally scale using, for instance, a Hadoop cluster, which is relevant to big data problems. The features from the mappers are used to create a model and that model is used to classify new instances.

Null Type and Predicting New Types

We maintain a single type that represents instances that cannot be associated with any of the known types; we call this *unknown* and it is akin to a null type. In terms of a prefilter, our goal is to reduce the number of evaluations; however, it is reasonable to assume that a group of instances will be harder to associate with a type. We use our type model to assist us with predicting a type when one cannot be assigned.

Each predicate is mapped to an entity type and each entity type is ontologically defined giving way to hierarchical relationships, equivalent relations, and pattern similarity relationships. We take an unfiltered approach for unknown instances and use the ontological definitions to then find candidate types. Future work will explore this method and will aide in our work to predict new types from existing types.

Experimentation

With each experiment we randomly sampled data for training and testing and normalized the data. Our first experiment consisted of using DBpedia to define our types, DBpedia data for training, and Arnetminer data (Tang, Zhang, and Yao 2007; Tang et al. 2008) for testing. Our second experiment consisted of using DBpedia to define our types, DBpedia data for training, and Freebase data for testing. Our third experiment consisted of using DBpedia to define our types and DBpedia data for training and testing, using two nonoverlapping samples.

Performance Metrics

We use the standard precision and recall metrics for measuring performance where *TruePositive* values are those that are expected to be true and are predicted to be true, *FalsePositive* values are those predicted to be true but are actually false, and *FalseNegative* are values that should be true but are predicted as false. We experimented with both stratified samples and nonstratified samples for training and testing.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (4)$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (5)$$

Evaluation

Our first experiment used 600 instances of the Arnetminer data set randomly selected and all of the type *Person* with each instance having an average of 11 predicates. We used the DBpedia data to build a training data set of 2000 instances with 240 entity types. When evaluating the filter performance, which is shown in table 1, we saw 100 percent accuracy in its performance in designating a candidate that is consistent with the known entity type. We also were able to classify person types with close to 100 percent accuracy as shown in table 2. The Arnetminer data is a sparse data set with a relatively small distribution of predicates across the data set. As a sensitivity test, we wanted to see how other classifiers classified the *Person* instances. We did find the *Place* and *Organization* classifiers had slightly lower accuracies; however, we saw higher accuracies when testing others such as *CreativeWork* and *Settlement*. There are on average only 11 attributes and there are

Test	Found ≥ 1 Type	Found All Types
Arnetminer	100 percent	100 percent
DBpedia	99 percent	97 percent
Freebase	60 percent	< 10 percent

Table 1. Prefiltering for Candidate Entity Types.

Type	Precision	Recall	f-measure
person	1	.98	.99
place	1	.65	.79
organization	1	.49	.66
creativework	1	.85	.92
settlement	1	.98	.99

Table 2. Arnetminer Results.

Type	Precision	Recall	f-Measure
place	0.6	0.576	0.562
person	0.635	0.629	0.625
athlete	0.336	0.376	0.337
organization	0.345	0.365	0.346
company	0.559	0.557	0.556
musical artist	0.495	0.494	0.49
architectural structure	0.478	0.477	0.473
film	0.444	0.444	0.444
building	0.612	0.61	0.609
book	0.661	0.659	0.658
soccer player	0.595	0.537	0.432
politician	0.6	0.6	0.598
event	0.361	0.371	0.356
body of water	0.446	0.444	0.444
school	0.6	0.6	0.589

Table 3. Sample of Freebase Entity Type Classifications.

null values for a large percentage, for example 95 percent of the title attribute is null.

When we experimented with the Freebase data set we used 2000 instances for training with 240 different entity types and 1000 instances for testing with over 470 different entity types. In table 1, we show filtering results for candidate selection. What we found is that we could relax the filtering algorithm in order for us to recognize more of the potential entity types; however, often it was the case that the DBpedia data set just did not contain the entity type represented in the Freebase data set. For example, 70 percent of the instances contain the type *rdf.freebase.com/ns/common.topic*.[4] In table 3 we show a sample of classification results. As the granularity between entity types in DBpedia is very different than in Freebase, we expected to see lower than average results. We show an example of this difference in figure 3 for entity type *Organization*.

For the DBpedia experiment we used 2000 instances for training with 240 different types, and 1000 instances for testing with 167 different entity types. There was no overlap between instances in the training and test data sets and data was sampled randomly. There were 155 overlapping entity types between the entity types in the test set and the entity types in the training sets. Since the training data and test data were taken from the same data set, we expected to see reasonable results. With regards to candidate filtering, as can be seen in table 1, we often found the types expected. However, the classification results were slightly lower than our assumptions; this can be attributed to the information gain filtering and also the need to optimize the mappers. In table 4 and figure 5, we show the precision, recall, and f-measure scores.

What we found was that when there was lower than expected performance, often the entity types in the test set were not sufficiently represented in the training set. We did not purposely try to create overlap between the training and test set. In the case of Arnetminer and the Freebase data set we are training with a completely different data set without any knowledge of type information, and therefore our preliminary results are encouraging. Our ongoing work will improve upon our existing preliminary implementation. For instance, we are currently working on statistical methods to measure the actual values of the relations and attributes since often label names cannot be mapped. We are also introducing another level of classification that is contextual.

Conclusions and Ongoing Work

Knowing the types of entities mentioned or referenced in heterogeneous graph data is useful and important for record linkage, coreference resolution, and other tasks related to the big data variety and veracity. In big data problems we believe the

Top DBpedia Predicates for Organization	Top Freebase Predicates for Organization
http://xmlns.com/foaf/0.1/name	http://rdf.freebase.com/ns/type.object.key
http://xmlns.com/foaf/0.1/homepage	http://www.w3.org/2002/07/owl#sameAs
http://dbpedia.org/ontology/type	http://rdf.freebase.com/ns/type.object.name
http://dbpedia.org/ontology/location	http://creativecommons.org/ns#attributionURL
http://dbpedia.org/ontology/city	http://www.w3.org/1999/xhtml/vocab#license
http://dbpedia.org/ontology/genre	http://creativecommons.org/ns#attributionName
http://dbpedia.org/ontology/foundingYear	http://rdf.freebase.com/ns/common.topic.article
http://dbpedia.org/ontology/associatedBand	http://rdf.freebase.com/ns/music.record_label.artist
http://dbpedia.org/ontology/associatedMusicalArtist	http://rdf.freebase.com/ns/common.topic.webpage
http://dbpedia.org/ontology/product	http://rdf.freebase.com/ns/common.topic.official_website
http://dbpedia.org/ontology/country	http://rdf.freebase.com/ns/common.topic.image
http://dbpedia.org/ontology/keyPerson	http://rdf.freebase.com/ns/organization.organization.headquarters
http://dbpedia.org/ontology/industry	http://rdf.freebase.com/ns/organization.organization.board_members
http://dbpedia.org/ontology/hometown	http://rdf.freebase.com/ns/organization.organization.date_founded
http://www.w3.org/2003/01/geo/wgs84_pos#lat	http://rdf.freebase.com/ns/location.location.containedby

Figure 3. Top Predicates for Organization.

absence of entity types is a real and continual problem.

Ideally, the data being processed is annotated with type information in an appropriate and rich semantic schema or ontology. However, in many cases, such type information is absent or unavailable. This is especially common when the data has been automatically generated from a table, spreadsheet, log file, or some other data format.

We have described our preliminary work for identifying fine-grained entity types. Our ongoing work will perform benchmark evaluations, will include experiments that use other data sources for bootstrapping, will include experiments that show how performance is affected by relation size, and will apply our approach to a particular domain, such as the medical domain. Our ongoing work will also include adding an additional level of contextual classification, such that, given a context, a certain set of entity types would become candidates for entity type recognition.

Notes

1. From the National Cancer Institute, pubs.cancer.gov.
2. dbpedia.org/ontology.
3. See C. Bizer, A. Jentzsch, and R. Cyganiak, State of the LOD Cloud, lod-cloud.net/state.
4. rdf.freebase.com/ns/common.topic.

Type	Precision	Recall	f-Measure
agent	0.743	0.738	0.736
person	0.781	0.774	0.773
place	0.727	0.724	0.723
populated place	0.772	0.772	0.772
settlement	0.8	0.799	0.799
work	0.843	0.838	0.838
creative work	0.843	0.838	0.838
athlete	0.805	0.798	0.797
species	0.851	0.85	0.85
eukaryote	0.746	0.74	0.738
organization	0.689	0.688	0.687
soccer player	0.895	0.893	0.893
animal	0.943	0.94	0.94
architectural structure	0.667	0.625	0.6
film	0.743	0.735	0.733
artist	0.833	0.813	0.81
album	0.778	0.733	0.722

Table 4: Sample Of DBpedia Entity Type Classifications

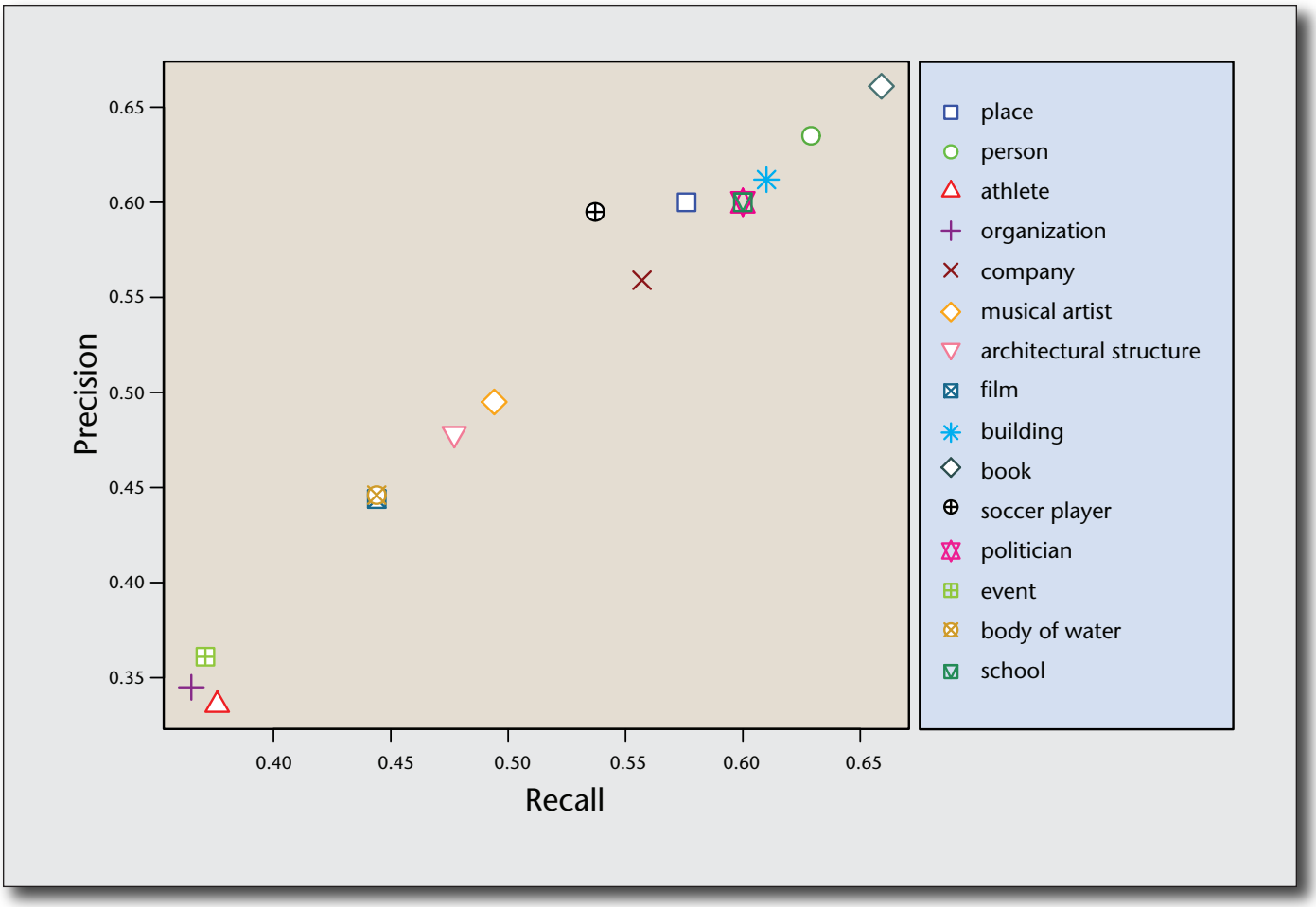

Figure 4. Freebase Precision and Recall

References

Albagli, S.; Ben-Eliyahu-Zohary, R.; and Shimony, S. E. 2012. Markov Network Based Ontology Matching. *Journal of Computer and System Sciences* 78(1): 105–118. dx.doi.org/10.1016/j.jcss.2011.02.014

Araujo, S.; Hidders, J.; Schwabe, D.; and de Vries, A. P. 2011. SERIMI — Resource Description Similarity, RDF Instance Matching and Interlinking. *The Computing Research Repository* (CoRR) July 2011: 1107–1104.

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web: 6th International Semantic Web Conference*, Lecture Notes in Computer Science, Volume 4825, 722–735. Berlin: Springer-Verlag.

Bechhofer, S.; Harmelen, F.; Hendler, J.; Horrocks, I.; McGuinness, D.; Patel-Schneider, P.; and Stein, L. 2004. *Owl Web Ontology Language Reference W3C Recommendation*, 10 February 2004. Cambridge, MA: World Wide Web Consortium E3C. (www.w3.org/TR/owl-ref).

Beckett, D. 2004. *RDF/XML Syntax Specification*. Cambridge, MA: World Wide Web Consortium W3C. (www.w3.org/TR/REC-rdf-syntax)

Berlin, J., and Motro, A. 2002. Database Schema Matching Using Machine Learning with Feature Selection. In *Advanced Information Systems Engineering,* Lecture Notes in Computer Science Volume 2348, 452–466. Berlin: Springer.

Berners-Lee, T.; Hendler, J.; and Lassila, O. 2001. The Semantic Web. *Scientific American* 284(5): 28–37. dx.doi.org/10.1038/scientificamerican0501-34

Bizer, C. 2009. The Emerging Web of Linked Data. *IEEE Intelligent Systems* 24(5): 87–92. dx.doi.org/10.1109/MIS.2009.102

Bizer, C.; Boncz, P.; Brodie, M. L.; and Erling, O. 2012. The Meaningful Use of Big Data: Four Perspectives–Four Challenges. *ACM SIGMOD Record* 40(4): 56–60. dx.doi.org/10.1145/2094114.2094129

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the ACM International Conference on Management of Data,* 1247–1250. New York: Association for Computing Machinery.

Brickley, D., and Guha, R. 2004. *Resource Description Framework (RDF) Schema Specification 1.0*. Cambridge, MA: World Wide Web Consortium E3C. (www.w3.org/TR/rdf-schema).

Do, H.-H.; Melnik, S.; and Rahm, E. 2003. Comparison of Schema Matching Evaluations. In *Web, Web-Services, and Database Systems,* Lecture Notes in Computer Science Volume 2593, 221–237. Berlin: Springer.

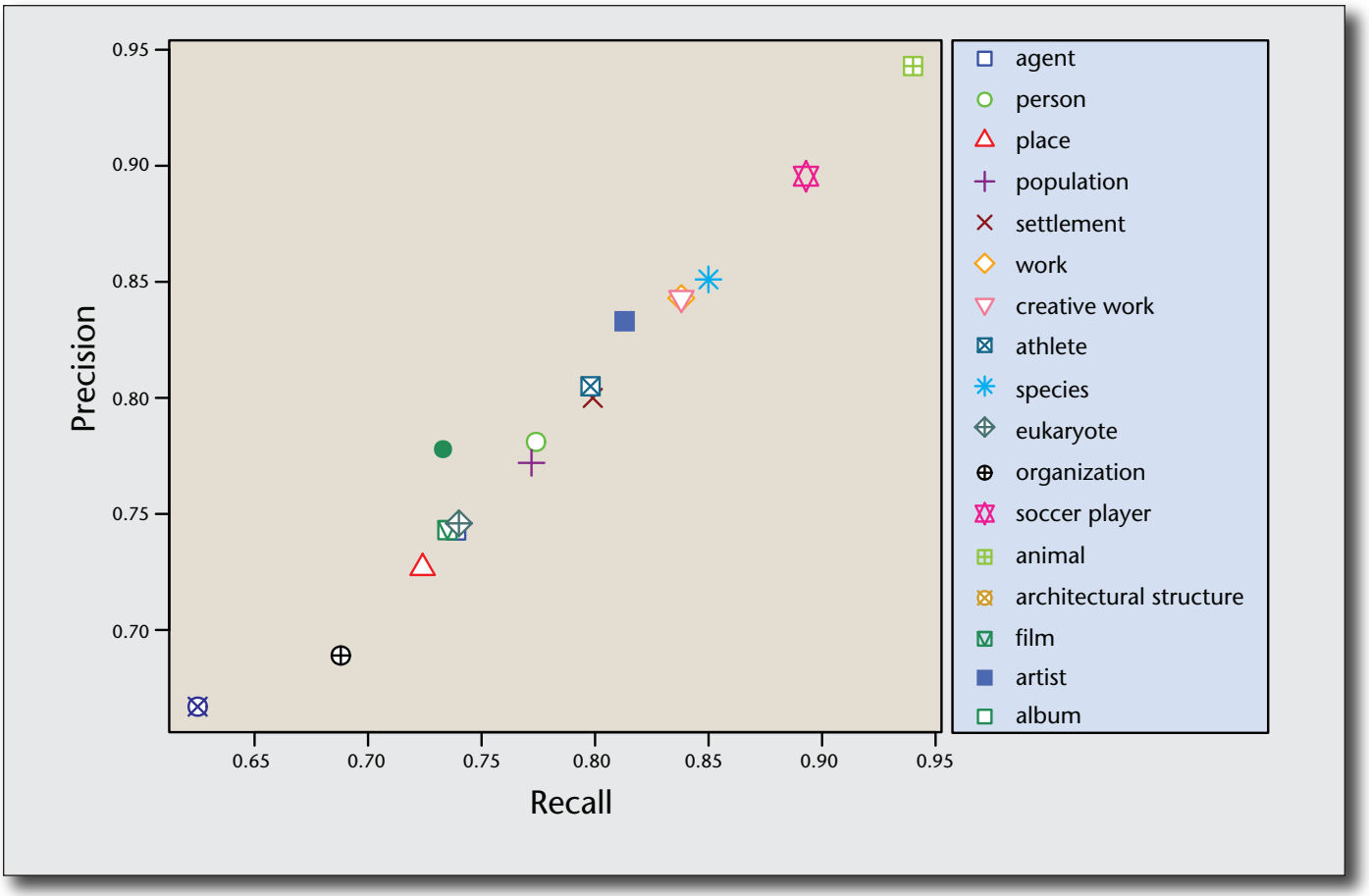

Figure 5. DBpedia Precision and Recall.

Dong, X. L., and Srivastava, D. 2013. Big Data Integration. In *Proceedings of the 2013 IEEE 29th International Conference on Data Engineering* (ICDE). Piscataway, NJ: Institute for Electrical and Electronics Engineers.

Euzenat, J., and Shvaiko, P. 2007. *Ontology Matching*. Berlin: Springer.

Ferrara, A.; Lorusso, D.; Montanelli, S.; and Varese, G. 2008. Towards a Benchmark for Instance Matching. In *Proceedings of the 3rd International Workshop on Ontology Matching,* CEUR Workshop Proceedings volume 431. Aachen, Germany: RWTH Aachen University.

Gomez-Perez, A.; Fernandez-Lopez, M.; and Corcho, O. 2004. *Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*. Berlin: Springer.

Jaiswal, A.; Miller, D. J.; and Mitra, P. 2010. Uninterpreted Schema Matching with Embedded Value Mapping Under Opaque Column Names and Data Values. *IEEE Transactions on Knowledge and Data Engineering* 22(2): 291–304. dx.doi.org/10.1109/TKDE.2009.69

Joachims, T. 2002. *Learning to Classify Text Using Support Vector Machines Methods, Theory, and Algorithms*. Berlin: Springer. dx.doi.org/10.1007/978-1-4615-0907-3

Lenat, D. B.; Prakash, M.; and Shepherd, M. 1985. CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks. *AI Magazine* 6(4): 65–85.

Levenshtein, V. 1966. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics Doklady* 10(8): 707–710.

Mayfield, J.; Alexander, D.; Dorr, B.; Eisner, J.; Elsayed, T.; Finin, T.; Fink, C.; Freedman, M.; Garera, N.; Mayfield, J.; McNamee, P.; Mohammad, S.; Oard, D.; Piatko, C.; Sayeed, A.; Syed, Z.; and Weischedel, R. 2009. Cross-Document Coreference Resolution: A Key Technology for Learning by Reading. In Learning by Reading and Learning to Read: Papers from the 2009 AAAI Spring Symposium. Technical Report SS-09-07. Menlo Park, CA: AAAI Press.

McAfee, A., and Brynjolfsson, E. 2012. Big Data: The Management Revolution. *Harvard Business Review* 90(10): 60–68.

McCallum, A.; Nigam, K.; and Ungar, L. 2000. Efficient Clustering of High-Dimensional Data Sets with Application to Reference Matching. In *Proceedings of the Second Annual International Conference on Knowledge Discovery in Data,* 169–178. New York: Association for Computing Machinery.

Miller, G. 1995. Wordnet: A Lexical Database for English. *Communications of the ACM* 38(11): 39–41. dx.doi.org/10.1145/219717.219748

Mitra, P.; Noy, N. F.; and Jaiswal, A. R. 2005. Omen: A Probabilistic Ontology Mapping Tool. In *The Semantic Web: ISWC 2005,* Lecture Notes in Computer Science, Volume 3729, 537–547. Berlin: Springer.

Nadeau, D., and Sekine, S. 2007. A Survey of Named Entity

Recognition and Classification. *Lingvisticae Investigationes* 30(1): 3–26. dx.doi.org/10.1075/li.30.1.03nad

Nikolov, A.; Uren, V.; Motta, E.; and Roeck, A. 2009. Overcoming Schema Heterogeneity Between Linked Semantic Repositories to Improve Coreference Resolution. In *Proceedings of the 4th Asian Conference on the Semantic Web,* Lecture Notes in Computer Science volume 5926, 332–346. Berlin: Springer.

Nikolov, A.; Uren, V.; and Motta, E. 2010. Data Linking: Capturing and Utilising Implicit Schema Level Relations. In *Linked Data on the Web 2010*. CEUR Workshop Proceedings volume 628. Aachen, Germany: RWTH Aachen University.

Nottleman, H., and Straccia, U. 2007. Information Retrieval and Machine Learning for Probabilistic Schema Matching. *Information Processing and Management* 43(3): 552–576. dx.doi.org/10.1016/j.ipm.2006.10.014

Paulheim, H., and Bizer, C. 2013. Type Inference on Noisy RDF Data. In The Semantic Web – ISWC 2013, Lecture Notes in Computer Science, Volume 8218, 510–525. Berlin: Springer. dx.doi.org/10.1007/978-3-642-41335-3_32

Rao, D.; McNamee, P.; and Dredze, M. 2010. Streaming Cross Document Entity Coreference Resolution. In *Proceedings of the 23rd International Conference on Computational Linguistics* (Posters), 1050–1058. Stroudsburg, PA: Association for Computational Linguistics.

Ratinov, L., and Roth, D. 2009. Design Challenges and Misconceptions in Named Entity Recognition. In *Proceedings of the 13th Conference on Computational Natural Language Learning*. Stroudsburg, PA: Association for Computational Linguistics. dx.doi.org/10.3115/1596374.1596399

Reeve, L., and Han, H. 2005. Survey of Semantic Annotation Platforms. In *Proceedings of the 2005 ACM Symposium on Applied Computing,* 1634–1638. New York: Association for Computing Machinery. dx.doi.org/10.1145/1066677.1067049

Seddiqui, M., and Aono, M. 2010. Ontology Instance Matching by Considering Semantic Link Cloud. In *9th WSEAS International Conference on Applications of Computer Engineering*. Athens, Greece: World Scientific and Engineering Academy and Society

Shvaiko, P., and Euzenat, J. 2008. Ten Challenges for Ontology Matching. In *On the Move to Meaningful Internet Systems: OTM 2008,* Lecture Notes in Computer Science, Volume 5332, 1164–1182. Berlin: Springer.

Singh, S.; Subramanya, A.; Pereira, F.; and McCallum, A. 2011. Large-Scale Cross-Document Coreference Using Distributed Inference and Hierarchical Models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Volume 1, 793–803. Stroudsburg, PA: Association for Computational Linguistics.

Sleeman, J., and Finin, T. 2010a. Computing FOAF Co-Reference Relations with Rules and Machine Learning. Paper presented at the Third International Workshop on Social Data on the Web held at the International Semantic Web Conference, Shanghai, China, 8 November.

Sleeman, J., and Finin, T. 2010b. A Machine Learning Approach to Linking FOAF Instances. In Linked Data Meets AI: Papers from the AAAI Spring Symposium. Technical Report SS-10-07. Menlo Park, CA: AAAI Press.

Song, D., and Heflin, J. 2011. Automatically Generating Data Linkages Using a Domain-Independent Candidate Selection Approach. In *The Semantic Web: ISWC 2011,* Lecture Notes in Computer Science, Volume 7032, 649–664. Berlin: Springer.

Suchanek, F. M.; Abiteboul, S.; and Senellart, P. 2012. Paris: Probabilistic Alignment of Relations, Instances, and Relations. *Proceedings of the VLDB Endowment* 5(3): 157–158 dx.doi.org/10.14778/2078331.2078332

Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; and Su, Z. 2008. Arnetminer: Extraction and Mining of Academic Social Networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 990–998. New York: Association for Computing Machinery. dx.doi.org/10.1145/1401890.1402008

Tang, J.; Zhang, D.; and Yao, L. 2007. Social Network Extraction of Academic Researchers. In *Proceedings of the 2007 IEEE International Conference on Data Mining*, 292–301. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Uryupina, O.; Poesio, M.; Giuliano, C.; and Tymoshenko, K. 2011. Disambiguation and Filtering Methods in Using Web Knowledge for Coreference Resolution. In *Proceedings of the 24th International Florida Artificial Intelligence Research Society Conference*. Palo Alto, CA: AAAI Press.

Jennifer Sleeman is a Ph.D. student in computer science at the University of Maryland Baltimore County. Her research interests include the semantic web, machine learning, coreference resolution, and knowledge base population.

Tim Finin is a professor of computer science and electrical engineering at the University of Maryland Baltimore County with more than 30 years of experience in applications of AI to problems in information systems and language understanding. His current research is focused on the semantic web, information extraction from text, and security and privacy. He is an AAAI Fellow, received an IEEE Technical Achievement award, and is the University of Maryland Baltimore County Presidential Research Professor for 2012–15. Finin received an S.B. from MIT and a Ph.D from the University of Illinois at Urbana-Champaign and has held full-time positions at the University of Maryland, Unisys, the University of Pennsylvania, and the MIT AI Laboratory. He is editor in chief of the *Elsevier Journal of Web Semantics* and a coeditor of the Viewpoints section of the *Communications of the ACM*.

Anupam Joshi is the Oros Family Chair professor of computer science and electrical engineering at the University of Maryland Baltimore County. He is the director of University of Maryland Baltimore County's Center for Cybersecurity and University of Maryland Baltimore County's Cyberscholars program. He obtained a B. Tech degree in electrical engineering from IIT Delhi and a master's and Ph.D. in computer science from Purdue University. His research interests are in the broad area of intelligent systems and networked computing with a focus on data management and security and privacy in mobile and pervasive computing environments, semantic web technologies, and text and graph analytics, especially as applied to social media and health care. He has published more than 200 technical papers, which have lead to more than 14,000 citations and an *h*-index of 65. He has filed and been granted several patents, and his research has been supported by DARPA, NSF (including a CAREER award), DoD, NASA, IBM, Microsoft, Qualcomm, Northrop Grumman, and Lockheed Martin.