Preprint version of: Jennifer Sleeman and Tim Finin, Type Prediction for Efficient Coreference
Resolution in Heterogeneous Semantic Graphs, 7th IEEE Int. Conf. on Semantic Computing, Sept. 2013.

# Type Prediction for Efficient Coreference Resolution in Heterogeneous Semantic Graphs

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

***Abstract*—We describe an approach for performing entity type recognition in heterogeneous semantic graphs in order to reduce the computational cost of performing coreference resolution. Our research specifically addresses the problem of working with semi-structured text that uses ontologies that are not informative or not known. This problem is similar to coreference resolution in unstructured text, where entities and their types are identified using contextual information and linguistic-based analysis. Semantic graphs are semi-structured with very little contextual information and trivial grammars that do not convey additional information. In the absence of known ontologies, performing coreference resolution can be challenging. Our work uses a supervised machine learning algorithm and entity type dictionaries to map attributes to a common attribute space. We evaluated the approach in experiments using data from Wikipedia, Freebase and Arnetminer.**

## I. Introduction

In natural language processing systems, coreference resolution is the task of determining when expressions in a document refer to the same thing, e.g., *Barack Obama*, *the President* and *he*. The term is also used for the more difficult task of linking expressions in multiple, independent documents, often described as *cross-document coreference resolution* [1]. A very similar problem arises in processing semantic graphs, whether encoded in the Resource Description Framework (RDF) or using some other semantic graph representation language. Here, the problem can be framed as determining when two nodes representing a thing (i.e., an instance rather than a class or property) denote the same object in the world.

The problem in a semantic graph is trivial if the nodes have associated identifiers that are identical, but that is typically not the case. Within a single graph, akin to the NLP intra-document coreference resolution case, we might hope that instances denoting the same underlying entity have such identifiers. But in RDF graphs this is generally not true, due to the use of instances created as "blank nodes". It is almost never the case when integrating semantic graphs that have been produced independently, whether from different sources or even the same source. In RDF graphs, we can assert that two nodes denote the same entity with the *owl:sameAs* property, but we still have the problem of determining when they are the same, so that asserting their equivalence is appropriate.

Very similar problems are long standing in databases and record processing, where the problem is referred to as *record linkage* [2], [3]. The process of identifying coreferent records is often called *deduplication* and is important for such useful tasks as maintaining a high quality mailing list for marketing purposes.

Performing coreference resolution over a collection of instances to determine which ones represent the same underlying entity inherently has an $O\left(n^2\right)$ cost, since in general, the potential coreference of every pair of instances has to be considered. In situations where the entities may be of a different type, the problem can be simplified by only considering pairs of the same type. While this does not reduce the "big-O" complexity of the task, it can result in a significant practical speed-up. In NLP information extraction contexts, the process of recognizing entity references typically also identifies their basic type (e.g., person, organization, place, event, chemical compound) and often a subtype (e.g., places might have subtypes like geo-political entity, populated place, facility and building).

Researchers have used different approaches to reduce the practical cost by filtering the number of instance evaluations by type or other features [4], [5], [6], [7], [8]. In our previous work [9], [10] we used a pairwise approach to determine when two people coreferred to each other and a filter, which applied low-cost rules to reduce the number of pairs of instances that needed to be evaluated. We evaluated instance data for which we knew the ontologies used and we also constrained the problem to data sets that were related to people. If an instance used an ontology that was not known, we simply ignored the information.

In our current work, we specifically address the issue of working with heterogeneous data, data which could originate from multiple sources and where the ontologies may often not be known. We also extend the problem to cover multiple types, rather than just one (i.e., people). In an effort to reduce the number of instances that need to be evaluated, we examine a way to distinguish entity types. By doing so, we partition the data into discrete groups of types and then apply a coreference resolution algorithm to each grouping. In this work we specifically address a way to group instances of the same type even when we cannot determine the type from the ontological definitions expressed in the data. Our problem then is closely related to the entity recognition problem in information extraction, i.e., the process of recognizing entities and their type (e.g., a person, location or organization) [11], [12], [13].

Typically when working with RDF data or other formalisms, the entity types and the entity properties can be explicitly known by means of a fairly well understood ontology. A significant amount of research addresses matching instances given well understood ontologies [14], [15], [16]. When ontologies are not accessible or not understood, or when several non-aligned ontologies are used, determining coreference by reasoning over the ontology is difficult and often impossible. We believe that this problem will become more common and significant with the increased addition of semantic annotation of big data applications.

Interoperability and integration are core problems addressed in database and ontology matching [17], [18], [19], [20], [21], [22]. Heterogeneous data, data sets which originate from various repositories, are typically harder to map simply because it is harder to establish that one attribute in a given schema is the same as an attribute defined in another schema [23].

Interoperability has also fueled the research related to the linked open data (LOD) [24] where heterogeneous repositories that use custom schemas are linked to each other by means of RDF and OWL assertions. Repositories are typically structured and formatted using RDF, then classes and properties are linked to other classes and properties that are represented in the LOD. The number and size of LOD collections has grown significantly in the past five years, however the total number of linked datasets as of 2011 is still relatively small (about 300) [25] and the degree of interlinking is often modest. This implies there is still quite of bit of unlinked data and among the data not linked it is likely a large majority use custom schemas. Nikolov et al. [26], [27] describe the problem of mapping heterogeneous data as it relates to coreference resolution, 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.

What we propose is a way to filter instances that need to be evaluated by grouping instances of the same type. Coreference resolution algorithms can then be performed over collections of instances of the same type. Additional filtering can also be used to reduce the cost further. However, a key aspect of this problem is addressing how to predict entity types given that the instance data originates from different sources and uses different schemas to represent the data. Again, in a structured environment, if one had access to the schemas then the schema could provide this information, however, we are a specifically addressing the problem of not having access to the schemas.

## II. Related Work

Recent work by Paulheim et al. [28] describes an approach for performing type inference using link analysis. They address the problem of using RDFS reasoning to infer types. Their work is based on the premise that certain relations occur with particular types of entities. They use a classification model to assign type probabilities and associate weights with properties that indicate how strongly they support particular type assertions. This work is comparable in that they are identifying types. Their weighting function can be compared to our use of entropy to establish which attributes best support specific types. However, their approach differs in that they use link analysis to develop their model. Instead, we develop dictionaries based on common ontologies and then map attributes associated with an instance with attributes found in the dictionary. We take this approach in order to support data represented using different ontologies, specifically when the schemas may not be known a priori.

Named entity recognition as performed during information extraction assumes that the text being processed is unstructured. Nadeau and Sekine [12] specifically address unstructured text and provide a survey of named entity recognition that includes both supervised to unsupervised approaches. In the supervised cases, rules are often induced. A deficiency of this approach is the need for large annotated corpus, which requires a significant and expensive effort. They describe the features that are typically used and categorize these features into three groups. Word-based features include case, punctuation, whether there are digits, morphology, parts of speech and various functions. A second feature category is list based, using dictionaries, stop words, common abbreviations, synonyms, etc. A third looks at document measures, such as occurrences, anaphora, enumerations, co-occurrences, etc. Our work uses a subset of these features that are relevant to semi-structured text to recognize entity types. With semi-structured data, many of the features are not possible to analyze because they do not exist. For example, there is no sentence structure,specifically in the semi-structured graph-based data.

Semantic annotation research typically performs entity type recognition but receives its input from an information extraction tool. This is more closely related to our work since semantic annotation tools are working with structured text, however the structure is based upon a tool that processes unstructured text. A survey by [29] defines semantic annotation as the mapping between ontology instances and classes. Semantic annotation tools are typically pattern based or use machine learning. A number of tools were benchmarked in this survey. We highlight this work because it is most closely related to our work, however, there is a clear distinction in that we do not receive information from an extraction tool but rather work directly with the RDF graphs or other similar formalisms. This makes the problem slightly more challenging as we are working with limited information. They use annotation recall and precision to compare tools, which we also use to measure our performance.

The concept of mapping attributes has been researched as it relates to database matching. Early work by Berlin et al. [17] describes research on database mapping using machine learning. In this work, they stress how data mapping is a labor-intensive job. Their Automatch system automates the schema-matching process using a machine learning approach. Their approach, in which a classifier is built using data from having domain experts map knowledge of attributes to a common dictionary, is foundational to our mapping approach. They saw over a 70% harmonic mean in their evaluation. Our work builds on this idea of mapping, however we take the approach of automating this concept by generating mappings from the DBpedia ontology and information gain.

There is a significant amount of work related to schema and ontology matching. The survey by [19] formalizes the schema/ontology matching problems and highlights promising work. We refer to more recent work in this area that specifically addresses heterogeneous data and is relevant to our work.

Work by Nikolov et al. [27] addresses the issue of automatic instance linking. They use LOD resources and knowledge of how instances are related to perform schema-level mappings. However, it is unclear how this work would handle a repository that is not linked to existing LOD resources or worse, how they would support data by which they did not have access to the schema. This is the problem we specifically address, as without this knowledge of the schema, mappings are harder to achieve. By mapping attributes to a known set of attributes associated with specific entity types, we can support instance matching without this schema knowledge or dependence upon the resources in the LOD cloud.

Work by [30] addresses the computational problem of coreference resolution by proposing a candidate selection algorithm that eliminates the need to compare each instance with every other one. The relevance of their work is their algorithm, which sets out to find candidate selection keys that are discriminating. They take a subset of instances based on some category, which could be an entity type. From this they then calculate three metrics: discriminability, coverage and F-Measure. They then use these keys, which are predicates, to perform candidate selection. The discriminability calculation is similar to our work which uses information gain. The key distinction between this work and ours is that they assume they have access to the knowledge needed to categorize the instances in order to begin the key discovery. Our work includes addressing the problem of how to learn these categories.

Work by [31] describes a process of matching ontologies based on the use of an upper ontology of very general concepts that are shared by many domains [32]. They specifically address the interoperability problem and how databases must share some “commonly understood concepts and relationships” and describe this common knowledge as a “semantic dictionary”. They then describe how to use an upper ontology for ontology matching. What is significant about this work is they highlight the importance of having a shared set of concepts and relationships and this description leads to the concept of an upper ontology. We can conceptually think of our dictionaries as ontological descriptions of our entity types.

## III. Problem Definition

Given data that is structured or semi-structured, when schemas are not known or not informative enough to determine entity types, there needs to be some other way to determine entity types. This is particularly challenging for heterogeneous data, where data can originate from multiple sources and when the context from information extraction tools is not present. We constrain this problem by identifying instances of type person, location and organization. The results of this work can be used to support coreference resolution and can act as a filter, limiting the number of instance pairs that need to be evaluated.

*Definition 3.1:* Given a set of instances $I$, if a pair of instances is coreferent then, $coref(I_1, I_2)$. Given $I_1$ has a set of attributes $(a_1, a_2, ...a_n)$ where $a \in A$ and $I_2$ has a set of attributes $(b_1, b_2, ...b_n)$ and $b \in B$, then $similarity(A, B)$ is used to establish coreferent instances, where highly similar attributes sets would mean there is a higher likelihood of $coref(I_1, I_2)$.

In order to reduce the number of instances that need to be evaluated, we try to establish each instance type. Since we do not know the meaning of $a$ or $b$ then we try to map $a$ and $b$ to a common dictionary set. We do this by first generating a set of dictionaries.

*Definition 3.2:* For each of the entity types in $person, location, organization$ we define a set of attributes $(p_1, p_2, ...p_n)$, $(l_1, l_2, ...l_n)$, $(o_1, o_2, ...o_n)$ that represent each type. We then use this information to determine if $person(I_1)|location(I_1)|organization(I_1)$ and $person(I_2)|location(I_2)|organization(I_2)$. This information can inform the coreference resolution algorithm as to whether evaluating $I_1$ and $I_2$ is necessary. Based on this mapping and labeled instances, we train a classifier to recognize which mappings belong to which entity types and then build a model that could be used to classify non-labeled instances.

## IV. Methodology

Our methodology includes a way to automatically build entity type dictionaries from existing data. We do this by choosing a data set that is rich with the properties found in a given ontology. For each entity type we build the dictionary based on calculating the information gain for each attribute. Our goal is to find attributes that define the type, but due to noisy data sets, we introduce a measure based on entropy that measures the uncertainty [33]. We then use a training set of labeled data and map the instances in the labeled data to our dictionaries. We do this using a set of mappers which each emit a score that is used as a feature. The features are then used with a supervised algorithm to create a classification model. We then perform mappings on unlabeled data and classify these instances using the supervised model. The results of this classification are instances classified as person, location or organization.

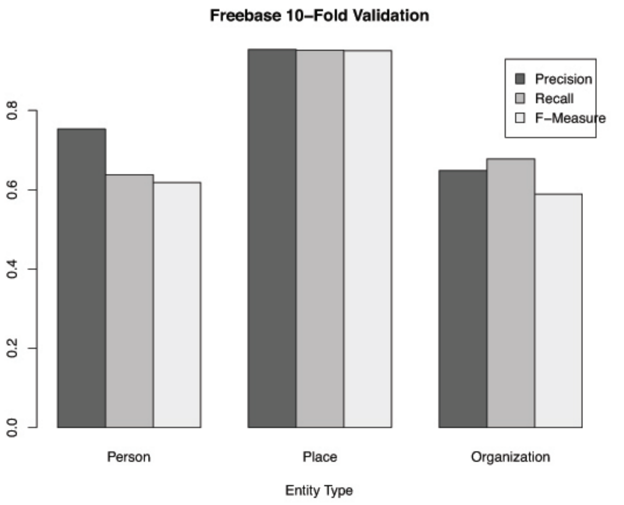

Fig. 1: Freebase ten-fold Validation

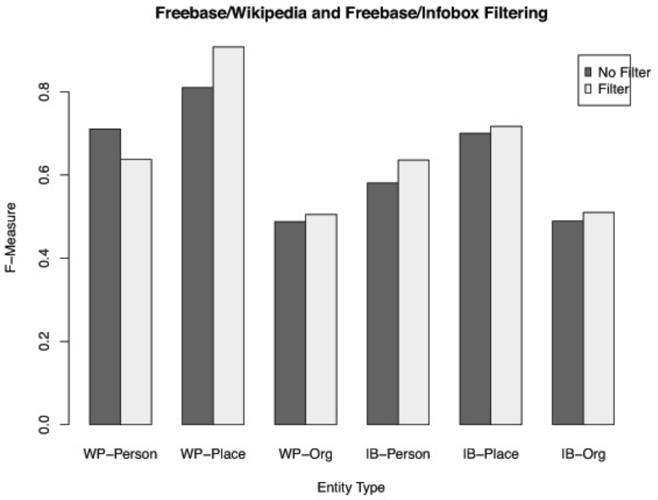

Fig. 2: Effects of Information Gain Filtering

This approach enables us to map attributes from different domains, hence supporting heterogeneous data. We use the DBpedia ontology [34] as a basis for the set of attributes for each entity type. We supplement this list with attributes from other common ontologies

TABLE I: Top 13 Attributes with high Information Gain.

| Attribute Name | Information Gain |
|---|---|
| foundation | 1.4822327945 |
| populationdensitykm | 1.4654167504 |
| headquarters | 1.4643749028 |
| almamater | 1.4511015423 |
| latm | 1.436957334 |
| lats | 1.435142071 |
| logo | 1.3989488118 |
| owner | 1.3923388965 |
| latd | 1.3851197789 |
| founder | 1.3507960305 |
| longm | 1.3420910151 |
| residence | 1.3420910151 |
| occupation | 1.3235410984 |

TABLE II: Top 13 Attributes with low Information Gain.

| Attribute Name | Information Gain |
|---|---|
| state | 0.0055689232 |
| image | 0.0514882243 |
| othername | 0.0643758458 |
| leader | 0.2039877679 |
| website | 0.2099404982 |
| language | 0.2127317307 |
| year | 0.2331585486 |
| name | 0.3119579608 |
| fullname | 0.3727722093 |
| branch | 0.3864105288 |
| area | 0.4058270139 |
| province | 0.4155409915 |
| nickname | 0.4256291604 |

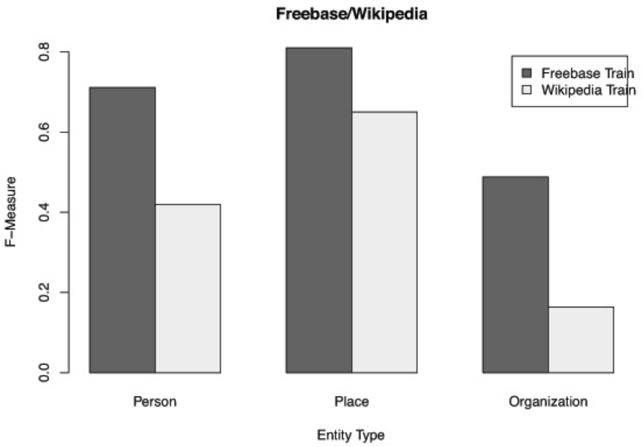

Fig. 3: Freebase/Wikipedia Freebase/Infobox Filtering

such as the Friend of a Friend ontology [35]. Using the 2011 DBpedia infobox dataset [36], we calculate the information gain for each attribute. Information gain allows us to measure which attributes best distinguish a class given our set of classes [33] and is well known for feature selection in various domains [37]. We chose the DBpedia infobox dataset because it is well known and has a good representation of the three entity types we wished to evaluate.

Assuming there are N classes (in this case we have three classes) we calculate entropy using the following equation:

$$Entropy = -\sum_{i}^{N} p(x_i) log_2 p(x_i) \qquad (1)$$

Since we are working with three classes, the maximum value if the instances were equally distributed would be 1.5850[33]. Given a sample set $S$ and attribute set $A$, the gain is calculated as follows:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \qquad (2)$$

Mapping to a common set of attributes is common among database and ontology mapping [17], [20], [21], [31], [22]. Abstractly speaking, one could think of this common set of attributes as an ontological representation. To perform this mapping we use a set of “mappers”: the first analyzes the attribute names by using a Levenshtein [38] distance measure, the second uses WordNet[39] synsets for expanding the attribute name both for the dictionary and the attribute associated with the instance being evaluated, and the third uses a common set of patterns to map attribute values. For example, there is a pattern that distinguishes an email address from other attributes. The score generated by each mapper becomes a feature in the feature vector. We then build a classification model using a Support Vector Machine (SVM). We perform the same mapping process for instances to be classified and then use the SVM model to classify the instances resulting in entity type classifications.

## V. Experimentation

With all experiments, we randomly selected a subset of instances with an equal distribution of persons, locations and organizations except when working with the Arnetminer dataset [40], [41], which is a dataset about people. We tested using the Freebase [42] dataset using 2000 instances with approximately 400 features,

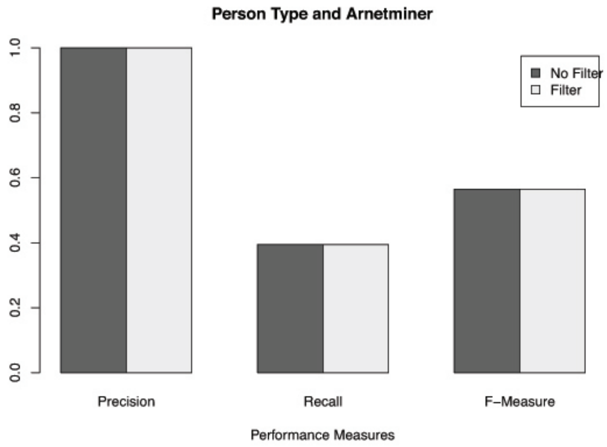

Fig. 4: ArnetMiner Instances

the Wikipedia data [43] using 3000 instances with about 400 features and the Arnetminer dataset contains 4000 instances with approximately 400 features (however all of the instances are one class). We used Weka [44] to run the experiments and standard precision and recall metrics for measuring performance:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (3)$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (4)$$

In our first experiment we used the Freebase dataset and ten-fold cross validation. This gave us some idea as to the general performance of the entity typing. With the second experiment we examined the effects of filtering using two different data sets. In the third experiment we looked at how well the entity typing performed using one dataset for training and the other for testing. We used the Freebase dataset for training and the Wikipedia dataset from Wikipedia [43] for testing and then Freebase for testing and Wikipedia for training. In the fourth experiment we used a dataset that in which entity types are inherently harder to classify. This data set specifically represents people, however useful information is sparse and there is a sizable amount of noisy information that could reduce the overall performance.

## VI. Evaluation

We used the output from our information gain calculation to weight our attributes. This weighting penalizes attributes that have low information gain across entity types. Figure 5 shows the results of processing the attributes and their associated gain. Table I and table II show examples of attributes that had the highest and lowest information gain.

In the first experiment, shown in Figure 1, we performed a ten-fold cross validation using the Freebase dataset for each entity type. The location entity type was most successfully classified and organization had lower success.

The second experiment, as seen in Figure 2, compares the effects of filtering given two data sets. We trained using the Freebase data set but then compared Wikipedia test data sets with Infobox test data sets. The goal with this experiment was to show that we often see an improvement when we apply filtering. We measured the effects of filtering on each type, using the Freebase dataset to build the model and the Wikipedia and Infobox dataset to test the model. Given the Wikipedia data set, the person entity type seemed to have worse performance when applying filtering. This was however not the case for the rest of the types.

In the third experiment, as seen in Figure 3, we first used Freebase for training and Wikipedia for testing then we switched and used Wikipedia for training and Freebase for testing. For each type we saw better performance when we used Freebase for training, which implies the Wikipedia data set does not generalize as well as the Freebase data set. When we combined the two data sets and took random samples, we saw F-measures in the 80% range.

In the fourth experiment, as seen in Figure 4, we tested the Arnetminer dataset using the Freebase data set to generate the model. The data set is both noisy and offers sparse set of attributes. For example, in some cases there is only an email and name attached to a record representing a person. We noticed that if we reduced the size of the training set the accuracy improved. We believe that this is due to the fact that we are reducing the number of negative cases.

We tested the various classifiers using these different data sets and what we saw was in one case decision trees performed very well, but in other cases it performed about the same as the SVM or worse. The naive Bayesian classifier consistently performed worse than the SVM. The SVM performed consistently well in comparison with the other classifier and hence this confirms that the SVM was the best choice for this particular problem. However, when testing non-linear kernels we saw no change in performance in comparison with a linear kernel.

In reviewing the benchmark described in [29], where there were six semantic annotation approaches evaluated,

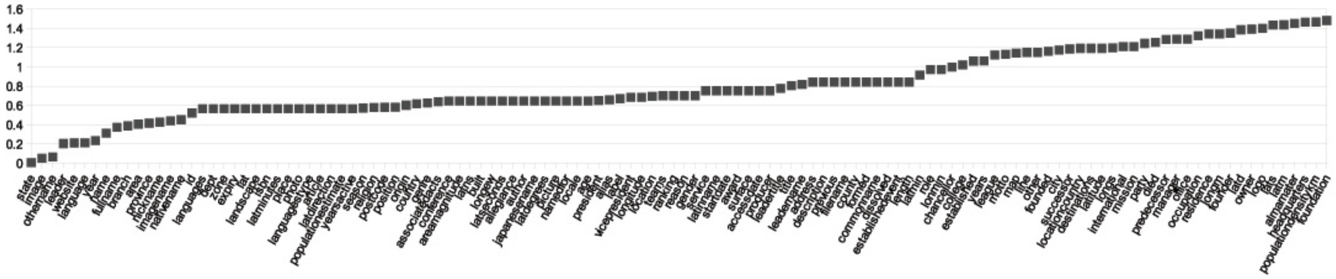

Fig. 5: We computed the information gain of attributes and used the results to weight them.

we believe that our method compares favorably. Since they are using information extraction tools and we are not and since their F-measure scores range from 24.9% to 92.9%, we consider our results for this baseline approach to be encouraging.

## VII. Conclusions

We described an approach for predicting the entity type of instances in heterogeneous semantic graphs in order to reduce the cost of performing coreference resolution. The problem is similar to performing coreference resolution for unstructured text, but cannot take advantage of the linguistic clues available in natural language documents. In the absence of known ontologies, performing coreference resolution can be challenging. We use supervised machine learning and entity type dictionaries to map attributes to a common attribute space. We evaluated the approach in experiments on data from Wikipedia, Freebase and Arnetminer.

Our baseline approach to performing entity type recognition for semantic graphs provide a way to support heterogeneous data, particularly when the ontologies used are not accessible or prove to be not very informative. Since we map to a common set of attributes for each type we can tolerate different data sets. Since we automatically generate this mapping without manual work, we provide an efficient way to support various data sets that may use custom schemas. When performing coreference resolution in a heterogeneous environment, one way to partition the data is by grouping instances by entity type. This along with other filtering mechanisms can help reduce the overall cost of performing coreference resolution.

## References

[1] J. Mayfield, D. Alexander, B. Dorr, J. Eisner, T. Elsayed, T. Finin, C. Fink, M. Freedman, N. Garera, J. Mayfield, P. McNamee, S. Mohammad, D. Oard, C. Piatko, A. Sayeed, Z. Syed, and R. Weischedel, "Cross-document coreference resolution: A key technology for learning by reading," in *Proc. AAAI Spring Symp. on Learning by Reading and Learning to Read*, March 2009.

[2] H. Dunn, "Record linkage," *American Journal of Public Health*, vol. 36, no. 12, p. 1412, 1946.

[3] H. Newcombe, J. Kennedy, S. Axford, and A. James, "Automatic linkage of vital records," *Science*, vol. 130, p. 954959, 1959.

[4] A. McCallum, K. Nigam, and L. Ungar, "Efficient clustering of high-dimensional data sets with application to reference matching," in *The Sixth Int. Conf. on Knowledge Discovery and Data Mining*. ACM SIGKDD, 2000, pp. 169–178.

[5] D. Rao, P. McNamee, and M. Dredze, "Streaming cross document entity coreference resolution," in *Int. Conf. on Computational Linguistics (COLING)*, November 2010, pp. 1050–1058.

[6] S. Singh, A. Subramanya, F. Pereira, and A. McCallum, "Large-scale cross-document coreference using distributed inference and hierarchical models," *Association for Computational Linguistics*, 2011.

[7] O. Uryupina, M. Poesio, C. Giuliano, and K. Tymoshenko, "Disambiguation and filtering methods in using web knowledge for coreference resolution," in *the 24th Int. Florida Artificial Intelligence Research Society Conf.*, 2011.

[8] D. Song and J. Heflin, "Automatically generating data linkages using a domain-independent candidate selection approach," in *The International Semantic Web Conference*. Springer Berlin Heidelberg, 2011, pp. 649–664.

[9] J. Sleeman and T. Finin, "A machine learning approach to linking foaf instances," in *Spring Symposium on Linked Data Meets AI*. AAAI, January 2010.

[10] ——, "Computing foaf co-reference relations with rules and machine learning," in *The Third Int. Workshop on Social Data on the Web*. ISWC, November 2010.

[11] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in *The seventh conference on Natural language learning at HLT-NAACL 2003*, vol. 4. Association for Computational Linguistics, 2003, pp. 188–191.

[12] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30.1, pp. 3–26, 2007.

[13] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *The Thirteenth Conf. on Computational Natural Language Learning*. Association for Computational Linguistics, 2009.

[14] A. Ferrara, D. Lorusso, S. Montanelli, and G. Varese, "Towards a benchmark for instance matching," in *Int. Workshop on Ontology Matching, volume 431, 2008*, 2008.

[15] M. Seddiqui and M. Aono, "Ontology instance matching by considering semantic link cloud," in *9th WSEAS Int. Conf. on Applications of Computer Engineering*, 2010.

[16] S. Araujo, J. Hidders, D. Schwabe, and A. P. de Vries, "Serimi-resource description similarity, rdf instance matching and interlinking," in *CoRR*, vol. 1107.1104, 2011.

[17] J. Berlin and A. Motro, "Database schema matching using machine learning with feature selection," in *Proceedings of the Conf. on Advanced Information Systems Engineering*. Springer, 2002, pp. 452–466.

[18] H.-H. Do, S. Melnik, and E. Rahm, "Comparison of schema matching evaluations," *Web, Web-Services, and Database Systems*, pp. 221–237, 2003.

[19] P. Shvaiko and J. Euzenat, "A survey of schema-based matching approaches," *Lecture notes in computer science*, vol. 3730, pp. 146–171, 2005.

[20] P. Mitra, N. F. Noy, and A. R. Jaiswal, "Omen: A probabilistic ontology mapping tool," in *Int. Semantic Web Conf.* Springer Berlin Heidelberg, 2005, pp. 537–547.

[21] H. Nottleman and U. Straccia, "Information retrieval and machine learning for probabilistic schema matching," *Information processing and management 43.3*, pp. 552–576, 2007.

[22] S. Albagli, R. Ben-Eliyahu-Zohary, and S. E. Shimony, "Markov network based ontology matching," *Journal of Computer and System Sciences 78.1 (2012)*, pp. 105–118, 2012.

[23] A. Jaiswal, D. J. Miller, and P. Mitra, "Uninterpreted schema matching with embedded value mapping under opaque column names and data values," *Knowledge and Data Engineering, IEEE Transactions on 22.2*, pp. 291–304, 2010.

[24] C. Bizer, "The emerging web of linked data," *IEEE Intelligent Systems*, vol. 24, no. 5, pp. 87–92, 2009.

[25] C. Bizer, A. Jentzsch, and R. Cyganiak, "State of the lod cloud," http://lod-cloud.net/state/, 2011.

[26] A. Nikolov, V. Uren, E. Motta, and A. Roeck, "Overcoming schema heterogeneity between linked semantic repositories to improve coreference resolution," in *A.Gomez-Perez and Y. Yu and Y. Ding eds.: The SemanticWeb, Fourth Asian Conf. ASWC 2009*, vol. 5926. Spring 2009, December 2009, p. 332346.

[27] A. Nikolov, V. Uren, and E. Motta, "Data linking: Capturing and utilising implicit schema level relations," in *Int. Workshop on Linked Data on the Web*, 2010.

[28] H. Paulheim and C. Bizer, "Type inference on noisy rdf data," in *International Semantic Web Conference*, 2013.

[29] L. Reeve and H. Han, "Survey of semantic annotation platforms," in *The 2005 ACM symposium on Applied computing*. ACM, 2005, pp. 1634–1638.

[30] D. Song and J. Heflin, "Automatically generating data linkages using a domain-independent candidate selection approach," in *Int. Semantic Web Conf.*, 2011.

[31] V. Mascardi, A. Locoro, and P. Rosso, "Automatic ontology matching via upper ontologies: A systematic evaluation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 609–623, 2010.

[32] B. Chandrasekaran, J. R. Josephson, and V. R. Benjamins, "What are ontologies, and why do we need them?" *Intelligent Systems and Their Applications, IEEE*, vol. 14, no. 1, pp. 20–26, 1999.

[33] M. Bramer, *Principles of data mining*. Springer, 2007.

[34] DBpedia, "Dbpedia," http://dbpedia.org/ontology/, 2013.

[35] D. Brickley and L. Miller, "Foaf vocabulary specification .98," August 2010, http://xmlns.com/foaf/spec/.

[36] DBpedia, "Dbpedia data set," http://dbpedia.org/Datasets, 2011.

[37] J. Hua, W. D. Tembe, and E. R. Dougherty, "Performance of feature-selection methods in the classification of high-dimension data," *Pattern Recognition*, pp. 409–424, 2009.

[38] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[39] G. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38(11), pp. 39–41, 1995.

[40] J. Tang, D. Zhang, and L. Yao, "Social network extraction of academic researchers," in *ICDM'07*, 2007, pp. 292–301.

[41] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: Extraction and mining of academic social networks," in *KDD'08*, 2008, pp. 990–998.

[42] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proc. ACM Int. Conf. on Management of Data*. ACM, 2008, pp. 1247–1250.

[43] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: a nucleus for a web of open data," in *Proc. 6th Int. Semantic Web Conf.* Berlin, Heidelberg: Springer-Verlag, 2007, pp. 722–735. [Online]. Available: http://dl.acm.org/citation.cfm?id=1785162.1785216

[44] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explorations*, vol. 11, 2009.