Opaque Attribute Alignment

Jennifer Sleeman #1,Rafael Alonso #2, Hua Li #3, Art Pope #4, Antonio Badia *5

#*SET Corporation an SAIC company*
SET Corporation an SAIC company, Arlington, VA 22201, USA, http://www.setcorp.com
[1]jsleeman@setcorp.com
[2]ralonso@setcorp.com
[3]hli@setcorp.com
[4]apope@setcorp.com

**University of Louisville*
Louisville, Kentucky, 40292, USA
[5]abadia@louisville.edu

***Abstract*— Ontology alignment describes a process of mapping ontological concepts, classes and attributes between different ontologies providing a way to achieve interoperability. While there has been considerable research in this area, most approaches that rely upon the alignment of attributes use label-based string comparisons of property names. The ability to process opaque or non-interpreted attribute names is a necessary component of attribute alignment. We describe a new attribute alignment approach to support ontology alignment that uses the density estimation as a means for determining alignment among objects. Using the combination of similarity hashing, Kernel Density Estimation (KDE) and Cross entropy, we are able to show promising F-Measure scores using the standard Ontology Alignment Evaluation Initiative (OAEI) 2011 benchmark.**

I. Introduction

Opaque Attribute Alignment (OAA) identifies similarity among attributes using a method that is not reliant upon attribute names or semantics. Rather it uses the instance data itself to evaluate similarity between attributes. We present OAA as a basis for performing ontology alignment.

Schema matching [1] describes a process of matching schema elements such as database attributes. Ontology matching [2] involves matching the elements of an ontology which can include instances, classes, attributes and relations. There is commonality between the two matching processes, and different approaches can exploit different aspects to achieve alignment. We focus on ontology matching in particular and how OAA can support ontology matching functions. Semantics alone do not always provide enough information to make an alignment assertion, therefore string-based evaluations of attributes may be used to augment semantic-based analysis. We argue that this is when accounting for opaque attributes will benefit the alignment success.

Work by [3] describes a need for handling opaque attributes which have obfuscated, or nonobservable names. For example, opaqueness becomes an issue when evaluating attributes that have the same name but are not the same, or attributes that have different names but are the same. OAA treats the problem of ontology alignment without depending upon, or being confused by the attribute name.

TABLE I
Opaqueness Examples

Opaqueness	Ontology 1	Ontology 2
Same name different meaning	ont1:name Alberto Trombetta	ont2:name Africa
Different name same meaning	ont1:location California	ont2:State CA

Using Kernel Density Estimation (KDE) enables OAA to be an unsupervised technique, which is necessary for domain independence, since it avoids the problems of choosing appropriate training data and domain transfer. The KDE approach uses the distribution of the data to estimate a density. We take advantage of this technique to address the problem of opaque attributes.

There are however, known limitations to this approach which we will address in this paper. We will address the limitations of performance, demonstrating results on non-numeric data and noisy data. We will show how using sampling and implementing optimizations to KDE, we overcome performance issues. We will show a competitive approach to overcoming the issue of working with non-numeric data for statistical analysis by using a similarity hash.

II. Related Work

There is a considerable amount of work in the general area of ontology alignment and attribute alignment. For an excellent overview we recommend the book by Schvaiko and Euzenat [2]. Here we focus our review on work most closely related to the present project; namely, attribute alignment based on the *opaque* or *non-interpreted* view.

Previous research related to attribute alignment can be classified along several dimensions ([4]). Approaches that take into account the values in an attribute are called *interpreted*; approaches that disregard the actual values and study the attribute as a whole (describing it through statistical or information theoretic measures) are called *non-interpreted* or *opaque*. The work presented here fits in this latter category.

The particular technique that we use, KDE, is popular in the field of image analysis and shape analysis [5], [6], [7]. However, as far as the authors are aware, it has rarely been used in the present context, namely, attribute alignment. One of the few works to do so is [8]; like the present work, they also use KDE over the instances of a class to determine semantic similarity. However, we make distinctions between categorical and continuous data, using two kernels. Work by [8] handles non-numerical data by transforming it into numerical data. We adopt this approach as well. However, while we use a similarity hash algorithm to deal with strings, they use a traditional edit distance. The edit distance has a number of shortcomings: it may give the same distance for two very different strings with reference to a given one -that is, given strings s, s_1 and s_2 it may be the case that $d(s, s_1) = d(s, s_2)$ even though s_1 and s_2 are quite different.

Another work that uses a kernel-based method is [9], which addresses the question of whether two sets of observations are generated by the same distribution. Most work on opaque attribute alignment uses information theoretic measures. The seminal work of Kang and Naughton ([3]) uses this approach, considering the values in a domain (attribute or column) of a relational database as a probability distribution, and uses their entropy as a measure of similarity. Also, the mutual information and conditional information between a pair of attributes in the same relation is computed. Most papers, including [10], [11], [12], [13], use information theory ideas expanded on the basic intuition of a seminal paper by Resnick [14] that analyzes the similarity of two objects in a taxonomy based not just on the hierarchical distance between them, but also on the probabilities of said objects occurring at all. One of the few techniques that does not use information theory directly is that of [15]. The approach relies on an *ensemble* of basic non-parametric classifiers, since it is meant to be universally applicable. However, the classifiers used are very weak, and may not work in differently typed domains (strings vs. numbers).

There is considerable work in the area of ontology alignment. As representation of modern systems, we overview the three top systems in the 2010 OAEI benchmark: ASMOV, RiMOM, and AgrMaker.

ASMOV ([16]) analyzes four features: lexical elements (basically, labels), relational structure (ancestor-descendant hierarchy), internal structure (property restrictions for concepts; types, domains, and ranges for properties; data values for individuals), and extension (instances of classes and property values). Measures obtained by comparing these four features are combined into a single value using a normalized weighted average. A distinguished feature of ASMOV is that *semantic verification* is used in these matches. Verification attempts to find if a mapping is semantically inconsistent with the information on either ontology. Inconsistent mappings are removed but remembered so the algorithm will not consider them again. Because general inconsistency checking is too complex (undecidable in the general case), ASMOV uses a tailored algorithm that considers five specific types of inconsistencies.

RiMON [17] also uses several strategies: the *name based strategy* calculates the edit distance between labels of two entities; the *metadata based strategy* considers the information of each entity as a document. Using standard Information Retrieval ideas (tf-idf weight for labels, cosine comparison), a similarity is computed. Finally, the *instance based strategy* also constructs a document for each entity, but it includes instances of the entity, as well as their properties. Potential alignments above a threshold are created by each individual strategy, and then combined. A similarity-flooding-like algorithm [18] is used to add more alignments.

Finally, AgreementMaker [19] hierarchically layers three different matchers: concept-based, structural and instance-based. Here, classes are compared based on their extensions, and properties are compared based on their range and domain. The results of individual matchers are put together with a Linear Weighted Combination.

It is notable that all systems use a combination of the same basic methods. They differ in how they combine the methods, but in essence the same three types of basic matchers are used: lexical, structural, and extensional. Note that when the instances are complex objects, a recursive procedure can be used. However, when the instances are simply names, only the lexical approach is available. Likewise, comparison of properties is different when the domain/range is a class or a datatype: for the latter case, only the lexical approach is applicable. Since a lexical analysis is in many cases carried out first to get some initial candidates for matching, it can be argued that modern systems still rely on a very crude initialization method. Hence, the Opaque Attribute Alignment approach proposed here can be seen as a sophisticated matcher that can be used as a module in any of the systems mentioned, to provide a more reliable indicator of possible initial matches.

III. Approach

Given two data sources $d1$ and $d2$ each of which has a set of attributes $A1$ and $A2$, we build a common probability space for each pair of attributes from $A1$ and $A2$. We determine the closeness of each pair of attributes using KDE to estimate a density over the instances of each attribute.

KDE is conventionally used with numeric data. The OAA algorithm converts non-numeric distributions into numeric distributions that can then be evaluated by KDE. We use a similarity hash algorithm that generates similar hashes for similar strings and produces a numeric representation of a non-numeric data value. As part of a normalization process, all strings are lower-cased, various punctuations are removed, and conversions are made when possible.

Cross entropy is measured between the two distributions of comparable attributes. The resulting cross entropy is used as a similarity metric to determine whether a pair of attributes within the data set are alignable.

A. Using Multiple Regressors

OAA makes use of multiple regressors that support continuous and discrete data; the use of multiple regressors is described in [20]. We define discrete data as data that has a finite set of possibilities, for example, days of the week, or gender. We define continuous data as data that can have an infinite amount of possibilities, for example temperature or last name. We use a heuristic that samples portions of the data to find repeatability among data items for a particular attribute. Based on the outcome of this heuristic we then use the appropriate kernel, discrete or continuous of nature.

Well known in the statistics community, KDE is non-parametric and estimates the probability density function of a random variable based on sampling [21]. KDE estimates a probability of density from a sample of data [22].

The estimated density is:

$$\hat{f}\left(x\right) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x-x_i}{h}) \tag{1}$$

where $x...x_i$ is the set of independent observations and h is the smoothing parameter (bandwidth). The bandwidth can have a significant effect on the estimation, and appropriate parameter estimation on h is key to the applicability of KDE. For a discussion of KDE's properties, see [23], [21].

K is the kernel function which satisfies the following condition:

$$\int_{-\infty}^{\infty} K\left(x\right) dx = 1$$

1) Continuous Regression: Kernel choice and parameter estimation can be addressed by many performance metrics. In this work, Mean Integrated Squared Error (MISE) and Asymptotic Mean Integrated Square Error (AMISE) are used to assess estimation error from the true density [23], [21]. In choosing a kernel and its associated smoothing parameters, we aim to minimize AMISE. We use the Epanechnikov kernel because it has been shown to be more efficient than a number of other kernels [23]. It is defined as:

$$K\left(x\right) = \frac{3}{4}\left(1-x^2\right) 1\left(|x| \leq 1\right) \tag{2}$$

Selection of the bandwidth is an important research topic; [24], [25], [26], [27], [28] show different current approaches to optimizing the selection process. When OAA employs a continuous kernel, Silverman's Rule of Thumb [21], [23] is used to calculate the bandwidth.

$$h = b \times min\left\{\hat{\sigma}, \frac{IQR}{1.34}\right\} \times n^{-\frac{1}{5}} \tag{3}$$

where n is the size of the sample, $\hat{\sigma}$ is the standard deviation of the sample, IQR is the interquartile range and b is a constant which is chosen based on the kernel used.

If the sample is normally distributed, this method is said to give optimal bandwidth. If the sample is not normally distributed, this method is said to give a bandwidth not far from optimal if the distribution is close to normal [29].

2) Discrete Regression: OAA makes use of Aitchison & Aitken's [30], [20] kernel in order to estimate densities for discrete data. Though kernel selection has less effect on the overall outcome as compared to bandwidth selection [23], our experiments have shown that using multiple kernels, namely Aitchison & Aitken's kernel for discrete data and Epanechnikov kernel for continuous data, yielded better performance than using a single kernel. Recall with discrete data we have a finite set of values; we can think in terms of categorizing this data.

For further theoretical discussion related to Aitchison & Aitken, refer to [30], [20].

B. Non-Numeric Data and Hashing

Using KDE presents a problem when working with non-numeric data. To overcome this issue, we use a hash representation of strings to represent data in a numeric way. Many hash functions are meant to reduce, or eliminate collision altogether, resulting in strings that may be lexically similar, but hashed to very different values. Our goal was to create hashes of similar strings that are also similar. We found work related to similarity hashing[31] that was promising based on previously conducted experiments [31], [32]. We implemented a 128-bit version of similarity hashing based on [31].

C. Cross entropy

The probability distributions generated by our kernel density methods are compared by generating a common probability space for each pair of distributions. Based on their cross entropy, a decision is made as to whether we consider the attributes alignable. We compared Cross entropy with other ways to measures distances and experiments showed that this method is competitive with other methods such as Kullback-Leibler divergence.

$$H\left(p,q\right) = -\sum_{x} p\left(x\right) log\, q\left(x\right) \tag{4}$$

where p and q are discrete probability distributions.

D. Sampling

KDE is known to have a complexity of O(MN)[33], where there are N points to evaluate by M samples. Though we have begun investigating optimizations suggested by [33], [28] and others, we have tested and show that with sampling we are able to keep the sampling size within a certain bound with minimal reduction in F-Measure scores.

IV. Experiments

Experiments included testing OAA using Ontology Alignment Evaluation Initiative (OAEI) Benchmark to measure precision, recall and F-Measure. We also conducted experiments which compare using a single continuous kernel with using both a categorical and continuous kernel. Our third experiment shows how using a similarity hash compared to other string-handling methods we tested. Finally, our sampling experiments show how we use sampling to reduce overall computing time without significantly impacting performance.

We use the Ontology Alignment Evaluation Initiative (OAEI) which provides a benchmark for testing ontology alignment [34]. We parsed each of the benchmark files, then retrieved the associated RDF files. We then parse the expected alignments and use this as ground truth along with additional custom ground truth based on common URIs. Since we currently are testing attribute alignment only, we exclude the class alignments from our evaluation. The OAEI includes a list of tests used to compare a reference ontology with variations. The tests are classified as concept tests and systematic tests that include missing instances, missing attributes, synonyms, misspelled data values, obfuscated attribute names and flattened hierarchies, see [34] for a full description. The reference ontology contains 33 named classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals [34]. This benchmark is used for the OAEI matching evaluation, final participant results of the 2011 evaluation are available [35].

The first version of our system analyzes the effectiveness of OAA. We modified our expected output to only include attributes and not class alignments. The next version of our system will test both attribute and class alignments. The results of figure 1 show our overall F-Measure score of 55% which is promising for our future work. When running each test we plotted recall in relation to precision. We also show the average precision, F-Measure and recall across test sets 1xx and 2xx.

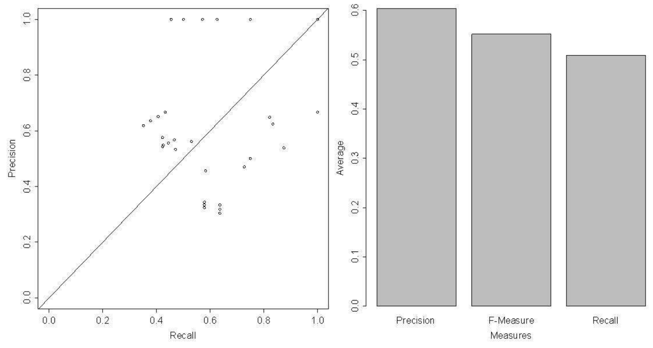

Fig. 1. Result Measures

We used the same benchmark to test the difference between using a single continuous kernel (Epanechnikov) and using mixed kernels (Epanechnikov and Aitchison & Aitken's Kernel), see figure 2. Mixed kernels offered overall better performance with F-Measure scores of 34% for a single kernel and 55% for mixed kernels.

A. Non-numeric Conversions

We developed an experiment to test different non-numeric to numeric conversions. Using the OAEI Benchmark, we tested using a simple cryptographic function, an implementation of Soundex [36], a simhash implementation based on the work of [31], [32] using 128 bits and a modified version of simhash that reduces collision. We show in figure 3 that simhash is competitive to the other implementations.

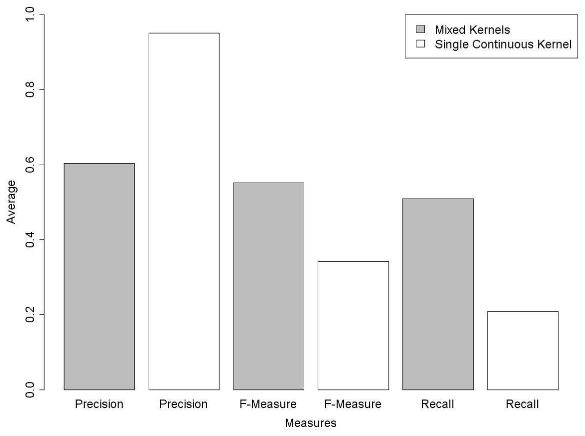

Fig. 2. Comparing Mixed Kernels to Single Continuous Kernel

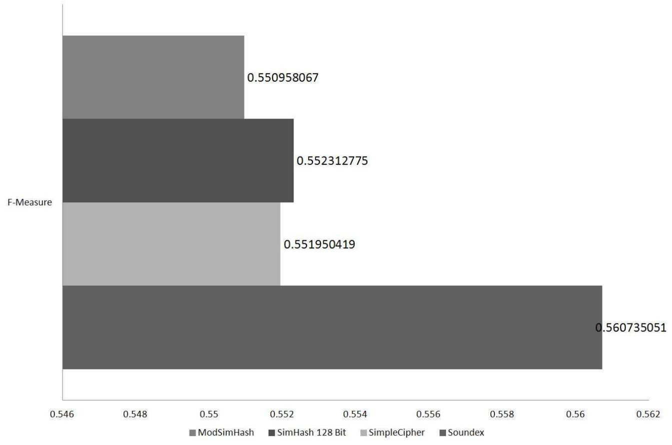

Fig. 3. Comparing Non-Numeric Conversions

B. Sampling

We developed experiments to test the effects of sampling when using kernel density estimation. In figure 4 and figure 5 we show that when we used a Monte Carlo simulation and made small, incremental changes to the sample size, there was a threshold reached where increasing the sample size would not improve the F-Measure further. This experiment allowed us to learn the smallest sample size that could be used without degrading accuracy and F-Measure scores. We tested using a public data set from the U.S. Census and further tested with two proprietary data sets. The data included structured and semi-structured input (RDF). Datasets 1 and 2 were semi-structured and dataset 3 was structured. Dataset 2 was wider than dataset 1 and 3 with over 30 attributes per data source and was a 'noisier' dataset with less consistency across tuples, more typographical errors and missing data. Dataset 3 had between 10 and 30 tuples per data source, where datasets 1 and 2 contained a larger number of tuples in the range of 4,000 tuples.

We compared different sampling sizes in relation to accuracy and time. Our initial experiments explored establishing

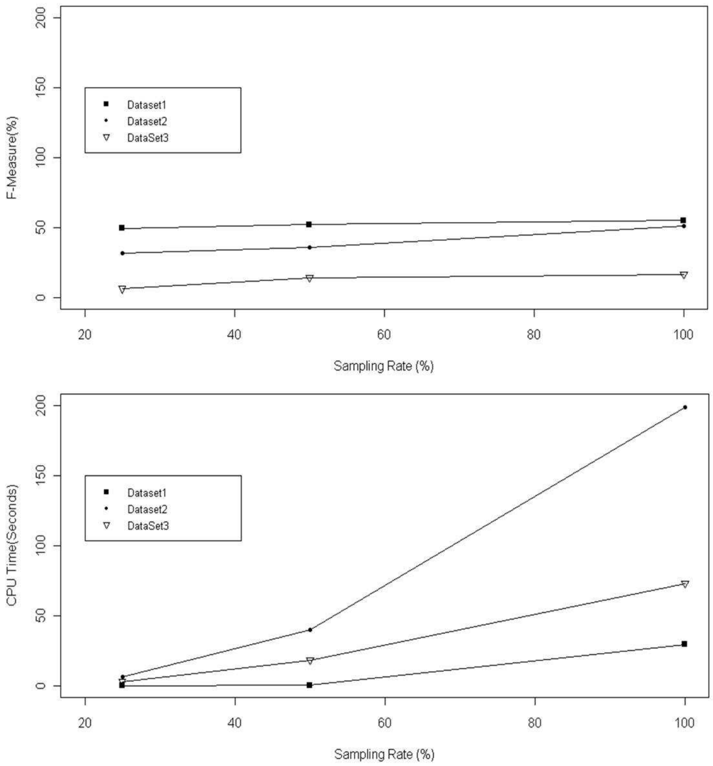

Fig. 4. Effects of Sampling on Performance and Computing Time

a sample size threshold. Sampling effects on performance showed that at a 50% sampling for data set 3, the loss in F-Measure scores was about 16%. Sampling resulted in a reduction in computing time of about 54% (50% sampling).

Our final experiment, shown in figure 5, used a larger data set (about 7,000 triples for each data source). We measured performance while incrementally increasing our sample size. Our goal was to test when our F-Measure scores would start to plateau. Our test took approximately 25 seconds to complete with an F-Measure score of 35%.

C. Evaluation

Through our experimentation we have proven the usefulness of KDE to perform opaque attribute alignment. Using the OAEI benchmark our F-Measure scores for test sets 1xx and 2xx was 55%. For initial results, we believe this is promising. We have shown that by using sampling, we can reduce overall computing time without a significant reduction in F-Measure scores. By using additional improvements described in research by [33], [28], we could further improve computing time. Our experiments have shown that similarity hashing can be used for converting non-numeric to numeric data. The authors note that there are known information losses with this type of conversion and are exploring clustering methods to further improve this approach.

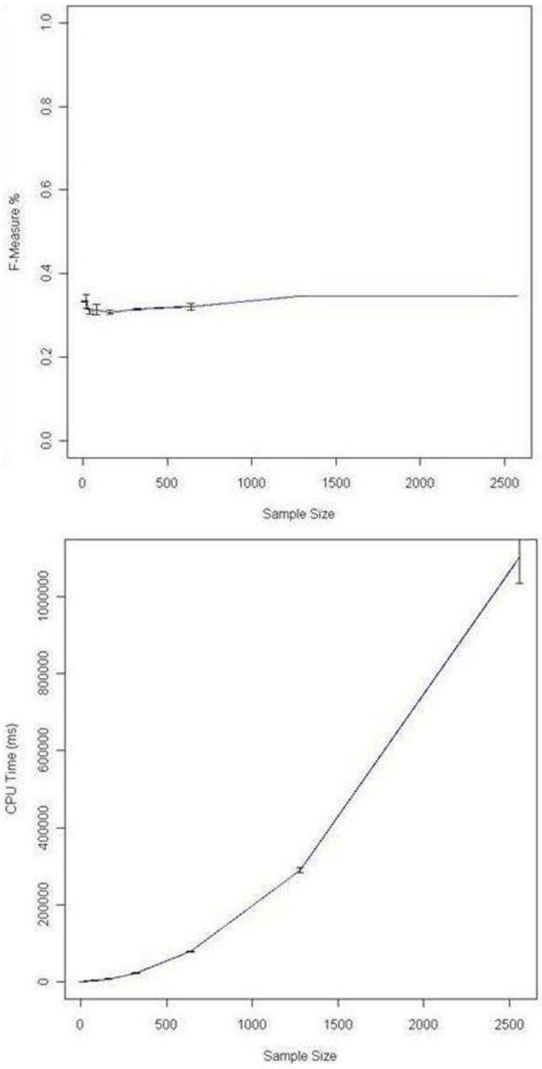

Fig. 5. Effects of Sampling on Performance and Computing Time

V. Conclusion and Future Work

We described a promising new approach for opaque attribute alignment, which is foundational to our future ontology alignment work. We have shown a way to convert non-numeric data into a numeric format with promising results. We have also shown how we use sampling to reduce computing time. Our future work will include aligning classes and instances using OAA. We are also experimenting with a clustering method for representing data values. Our research will continue to explore ways to improve bandwidth selection, computing time, and will further improve existing methodology.

Acknowledgment

This material is based in part upon work supported by the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory contract number FA8650-10-C-7060. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, AFRL or the U.S. Government.

REFERENCES

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

[2] J. Euzenat and P. Shvaiko, *Ontology matching*. Heidelberg (DE): Springer-Verlag, 2007.

[3] J. Kang and J. F. Naughton, "On schema matching with opaque column names and data values," 2003, p. 205216.

[4] P. Shvaiko and J. Euzenat, "A survey of schema-based matching approaches," University of Trento, Tech. Rep. DIT-04-087, 2004, also: Journal of Data Semantics, LNCS 3730, pp. 146-171, 2005.

[5] A. Elgammal, R. Duraiswami, and L. S. Harwood, D. andDavis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," in *Proc. of the IEEE*, vol. 90, no. 7, 2002, pp. 1151–1163.

[6] A. Mittal and N. Paragios, "Motion-based background subtraction using adaptive kernel density estimation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. II, 2004, pp. 302–309.

[7] D. Comaniciu and P. Meer, in *Mean shift: A robust approach toward feature space analysis*, vol. 24, no. 5. IEEE Trans. Pattern Anal. Machine Intell, 2002.

[8] Y. Xue, H. H. Ghenniwa, and W. Shen, "Instance-based domain ontological view creation towards semantic integration," *Expert Systems with Applications*, vol. 38, no. 2, pp. 1193 – 1202, 2011.

[9] K. Borgwardt, A. Gretton, M. Rasch, H. Kriegel, B. Schlkopf, and S. AJ, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–357, 2006.

[10] D. Lin, "An information-theoretic definition of similarity," in *Proc. of the Fifteenth International Conference on Machine Learning*, 1998, pp. 296–304.

[11] P. Pantel, A. Philpot, and E. Hovy, "Aligning database columns using mutual information," in *Proc. of Conference on Digital Government Research (DG.O-05*, 2005, pp. 205–210.

[12] J. Kang, T. S. Han, D. Lee, and P. Mitra, "Establishing value mappings using statistical models and user feedback," in *In ACM CIKM*, 2005.

[13] B. T. Dai, N. Koudas, D. Srivastavat, A. K. H. Tung, and S. Venkatasubramaniant, "Validating multi-column schema matchings by type," in *In 24th International Conference on Data Engineering (ICDE)*, 2008, pp. 120–129.

[14] P. Resnik, "Using information content to evaluate semantic similarity in a taxanomy," in *International Joint Conference for Artificial Intelligence (IJCAI)*, 1995, pp. 448–453.

[15] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, *Handbook on Ontologies in Information Systems*. Springer-Velag, 2004, ch. Ontology Matching: A Machine Learning Approach.

[16] Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabula, "Ontology matching with semantic verification," *Web Semantics: Science, Services and Agents on the World Wide Web*, 2009.

[17] A. Wang, X. Zhang, L. Hou, Y. Zhao, and J. Li, "Rimon results for oaei 2010," pp. 194–201.

[18] I. Bhattacharya and L. Getoor, "Collective entity resolution in relational data," *TKDD*, vol. 1, no. 1, 2007.

[19] I. F. Cruz, F. P. Antonelli, and C. Stroe, "Agreementmaker: Efficient matching for large real-world schemas and ontologies," in *Proc. of the Very Large DataBases Conference (VLDB)*, 2009.

[20] J. Racine and Q. Li, "Nonparametric estimation of regression functions with both categorical and continuous data," *Journal of Econometrics*, vol. 119(1), pp. 99–130, 2004.

[21] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall, 1986.

[22] T. Hastie, R. Tibshirani, and J. Friedman, in *The Elements of Statistical Learning*. Springer, 2001, pp. 182–189.

[23] D. Scott, *Multivariate density estimation: theory, practice, and visualization*. John Wiley and Sons Inc., 1992.

[24] B. Hansen, "Bandwidth selection for nonparametric distribution estimation," 2004. [Online]. Available: http://www.ssc.wisc.edu/~bhansen

[25] X. Zhang, M. King, and R. Hyndman, "Bandwidth selection for multivariate kernel density estimation using mcmc," 2004.

[26] S. Sheather and M. Jones, "Reliable data-based bandwidth selection method for kernel density estimation," *Journal of the Royal Statistical Society*, vol. 53, pp. 683–690, 1991.

[27] B. Turlach, "Bandwidth selection in kernel density estimation: A review," 1993.

[28] V. C. Raykar and R. Duraiswami, "Very fast optimal bandwidth selection for univariate kernel density estimation," 2005.

[29] Hardle, Muller, Sperlich, and Werwarz, *Nonparametric and Semiparametric Models, An Introduction*, 1995.

[30] J. Aitchison and C. G. G. Aitken, "Multivariate binary discrimination by the kernel method," *Biometrika*, vol. 64, pp. 413–420, 1976.

[31] M. Charikar, "Similarity estimation techniques from rounding algorithms." ACM Press, 2002.

[32] G. Manku, A. Jain, and A. Sarma, "Detecting near-duplicates for web crawling." ACM Press, 2007.

[33] C. Yang, R. Duraiswami, N. Gumerov, and L. Davis", "Improved fast gauss transform and efficient kernel density estimation." 2003, pp. 464–471.

[34] O. A. E. I. O.-. Campaign, "Ontology alignment evaluation initiative," "http://oaei.ontologymatching.org/2011/benchmarks/", 2011.

[35] J. Euzenat, A. Ferrara, W. van Hage, L. Hollink, C. Meilicke, A. Nikolov, F. Scharffe, P. Shvaiko, H. Stuckenschmidt, O. Svab-Zamazal, and C. T. dos Santos", ""final results of the ontology alignment evaluation initiative 2011"."

[36] D. Holmes and C. M. McCabe, "Improving precision and recall for soundex retrieval," in *IEEE International Conference on Information Technology Coding and Computing (ITCC*, 2002.