ISSN No. 0976-5697

Volume 3, No. 1, Jan-Feb 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A New Method for Coreference Resolution in the Web of Linked Data Based On Machine Learning

Leila Namnik*
Computer engineering department
Science and Research Branch, Islamic Azad University
Khouzestan, Iran
Leila.namnik@gmail.com

Mehran Mohsenzadeh
Computer engineering department
Science and Research Branch, Islamic Azad University
Tehran, Iran
M.mohsenzadeh@srbiau.ac.ir

Mashallah Abbasi Dezfouli
Computer engineering department
Science and Research Branch, Islamic Azad University
Khuzestan, Iran
Masha_abbasi@yahoo.com

Abstract: Web of Linked Data forms a single, globally distributed data space. Establishing RDF links between overlapping but separately constituted RDF datasets still represents one of the most important challenges to achieve the vision of the Web of Linked Data. In Linked Data environment, an object is likely to be denoted with multiple URIs by different data providers. Object coreference resolution is to identify "equivalent" URIs that denotes the same object. One of the most important types of RDF links are "Identity Links", which point at coreferent objects. By common agreement, Web of Linked Data uses *owl:SameAs* predicate to state identity links. Driven by the Linking Open Data (LOD) project, millions of URIs have been explicitly linked via *owl:sameAs*, but potentially coreferent ones are still considerable. Coreference resolution and data linking often relies on fuzzy similarity functions comparing relevant characteristics of objects in the considered datasets and manually tuned metrics for estimating similarity between objects. In this paper, we describe an approach for object coreference resolution in Linked Data, which relies on supervised learning and support vector machines. We propose to employ different similarity functions and combine them with a learning scheme. Initial experiments applying this approach to public datasets have produced encouraging results.

Keywords: Coreference Resolution, Data Interlinking, Linked Data, Semantic Web, SVM.

I. INTRODUCTION

Linked Data principles, first outlined by Tim Berners-Lee in 2006 [1], provide simple guidelines for publishing, interlinking, and accessing structured data in a uniform machine-understandable format. Linked Data is based on well-known Web technologies like URI and RDF, and enables creating "Web of Data", the data infrastructure required for realizing the Semantic Web vision [2]. A fundamental prerequisite of the Semantic Web is the existence of large amounts of meaningfully interlinked RDF data on the Web. The W3C SWEO[i] community project, Linking Open Data (LOD)[ii], is the most important project initiated to bootstrap the Web of Data by publishing and interlinking well-known open web datasets as Linked Data and creating a huge Linked Data cloud. The interlinking of diverse datasets in "Web of Data" will enable users to easily navigate between these datasets in a manner analogous to how users currently navigate from one webpage to another in the "Web of Documents." RDF links are fundamental for the Web of Data as they are the glue that connects data islands into a global, interconnected data space and as they enable applications to discover additional data sources in a follow-your-nose fashion. In Linked Data environment, an object is likely to be denoted with multiple URIs by different data providers. The term 'Coreference' is used to define the situation where multiple URIs identify the same resource. Object coreference resolution is to identify coreference objects.

The most important types of RDF links in Linked Data are "Identity Links", which point at these coreferent objects. By common agreement [1], Web of Linked Data uses the link type, *http://www.w3.org/2002/07/owl#sameAs*, to state identity links. Identity links enable clients to retrieve further descriptions about an entity from other data sources. These links have an important social function as they enable different views of the world to be expressed on the Web of Data.

Often LOD datasets have overlapping domains and hence provide information about the same entity. For small datasets published manually, it is possible to create such links manually but doing so for large datasets is impractical. Data linking therefore often relies on fuzzy similarity functions comparing relevant characteristics of objects in the considered datasets. Indeed, such a task is made difficult by the fact that different datasets need to be re-conciliated while not sharing commonly accepted identifiers (such as ISBN codes), not relying on the same schemas and ontologies (therefore using different properties to represent the same information) and often implementing different formatting conventions for attributes (e.g., using \Berners-Lee, Tim" as the name of a person in one case, and \Tim Berners-Lee" in the other) [3].

In this paper we propose a method that calculates different similarity measures and combining them via a machine learning approach. We take into account knowledge defined in the ontology for reducing the problem solving space. The remainder of this paper is organized as follows. Related work is discussed in Section II. In section III we present definitions and our proposed method for coreference resolution. The experimental results on a real-world datasets are reported in Section IV. Finally, Section V concludes this paper and gives future work.

II. RELATED WORK

The problem of coreference was originally studied in the database community where it is known as record linkage or object identification [4]. With the development of the linked data initiative, it gains importance in the Semantic Web community where it is studied under the names of coreference resolution [5], reference reconciliation [6], and link discovery [7]. Current frameworks for link discovery in Linked Data can be subdivided into two categories: domain-specific and universal frameworks. Domain-specific link discovery framework aim at discovering links between knowledge bases from a particular domain. For example, the RKB knowledge base uses URI lists to compute links between universities and conferences [8]. For each new mapping task, a new program has to be written, wherein the source, target and mapping function must be declared. Another domain-specific tool is GNAT [9], which was developed for the music domain.

GNAT uses similarity propagation on audio fingerprinting to discover links between music data sets. Universal link discovery frameworks are designed to carry out mapping tasks independently from the domain of the source and target knowledge bases. For example, RDF-AI [10] concentrates on the data-level issues which occur when combining datasets using the same schema. The algorithm builds on string (Monge-Elkan) and linguistic (Word-Net) similarity measures to calculate similarities between literal property values, and then invokes an iterative graph matching algorithm to calculate a distance between individuals. While RDF-AI considered datasets described under a unique ontology, the KnoFuss architecture [11] tackles the data interlinking problem whether or not the datasets are described under the same ontology. It is based on a generic component-based approach allow to select the best appropriate method for a given interlinking task. Silk [12] provides a flexible, declarative language for specifying matching heuristics. Silk employs different string based distances. Parameters such as threshold and aggregation mechanisms for specific datasets have to be manually defined by the user. As such, it has the limitations; in particular, it ignores relevant types of evidence: the structure of the semantic data graph and knowledge defined in the ontology. Manually coming up with logic for combining similarity scores is difficult; we have used a learning based approach in this paper.

III. PROPOSED METHOD

This section shows the definitions and details of our method for coreference resolution in the Web Of Linked Data. Let D_1 and D_2 represent two RDF datasets, each one containing a set of resources where identified by URIs and described using the schema and properties defined in the corresponding ontology O_i. In this paper we suppose that two datasets are described under a same ontology O. We consider the problem of coreference resolution as follows: Finding URI_i in our target dataset, D_2, which identify the same resource as a URI_j in our seed dataset, D_1. The procedure of our method involves the following steps:

Step 1: Splitting Primary RDF Datasets by Ontology-Level Features

Generally, the data scale is very large in RDF datasets and comparing a resource in the source dataset with the whole resources in the target dataset might thus lead to tractability issue. We defined usage of schema-level features, as a heuristic to reducing the problem complexity. SPARQL queries used for this purpose.

First we obtained all different ontology classes and then extracted the instances of each class. We used classes that mentioned explicitly by *rdf:type* relations in the schema.

- *SPARQL query:*

SELECT DISTINCT $?C_i$
Where { ?Instance rdf:type $?C_i$ *}*

$C = \{C_1, C_2, \ldots C_i\}$

- *SELECT ?Instance*
  Where ?Instance rdf:type C_1
- *SELECT ?Instance*
  Where ?Instance rdf:type C_2

.....

- *SELECT ?Instance*
  Where ?Instance rdf:type C_i

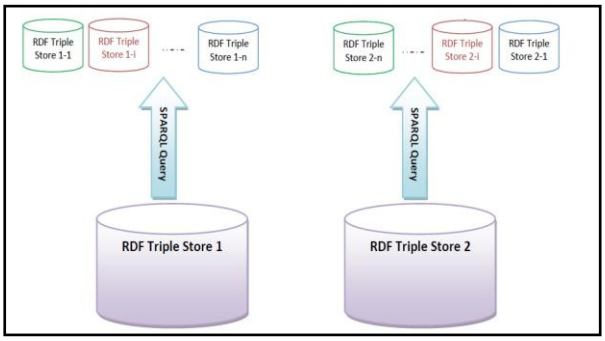


Figure 1: Proposed Method: Step 1

Step 2: Selecting Suitable Class Properties

In this step, first we extracted all of the properties for each class i (RDF Triple Store i). For example, the type *opus:Article* in the opus ontology[iii], has six properties as follows*: rdfs:label, opus:year, opus:journal_name, opus:author, opus:volume, opus:number*. Some properties are more suitable for describing an object so we have chosen

subset of P, where these properties be always available and can be used as a key property. For example, resources of type *opus:Publication* have a key property *opus:isbn*, which is not always available. Alternatively, a publication can be identified by the set of its other properties, such as the title (*opus:title*), the publication year (*opus:year*), and the venue (*opus:venue*). As another example, the foaf:mbox property, describing a personal e-mail address, can be used as a key property. However, it does not guarantee good recall: the same person may be described in different datasets with different emails and in many cases the e-mail address is not known.

Therefore we proposed to choose K suitable properties for each class by considering input datasets characteristics and problem solving domain.

***Step 3:* String Similarity Measures**

Each coreference resolution task requires approximate textual similarity algorithms to support the matching task. There are several well-known string similarity metrics for computing textual similarity, which can be separated into two groups: token-based and character-based techniques [13]. Character-based measures compute similarity between strings by estimating the minimum sequence of changes that transform one string into another. Token-based techniques treat a string as a "bag of words". When data is represented by relatively short strings that contain similar yet orthographically distinct tokens, character-based measures are preferable since they can estimate the difference between the strings with higher resolution. While character-based similarity metrics work well for typographical errors, it is often the case that typographical conventions lead to rearrangement of words (e.g.,"John Smith" versus "Smith, John").

In such cases, character level metrics fail to capture the similarity of the entities. Token-based metrics try to compensate for this problem. The best known character-based string similarity metric is Levenshtein [14] which defined as the minimum number of insertions, deletions or substitutions necessary to transform one string into another. This measure is very effective in detecting typo problems. Another popular character-based measure is Jaro-Winkler [15] which is based on the number and order of the common characters between two strings. Another measure is Jaccard [15] which operate at token level, comparing two strings by first tokenizing them and then dividing the number of tokens shared by the strings by the total number of tokens. To take advantage of these measurements, our method uses two character-based measures, Levenshtein and Jaro-Winkler and a token-based measure, Jaccard, in a particular way.

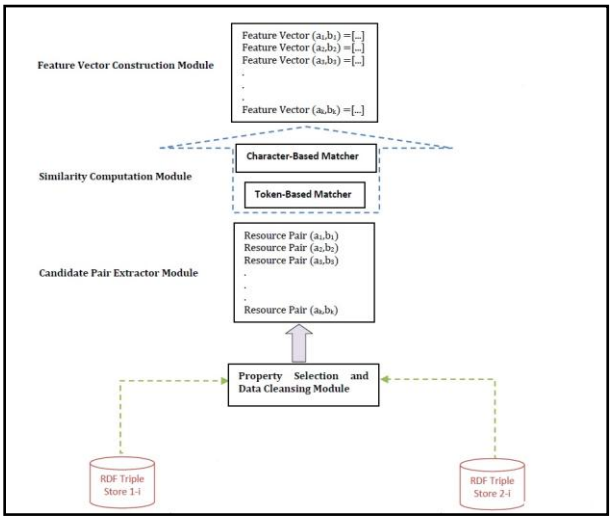
We assume in this paper, there are two different matcher:

1- Token-Based Matcher
2- Character-Based Matcher

In General, matchers use d predefined similarity meatures.

Step 4: Constructing Feature Vectors

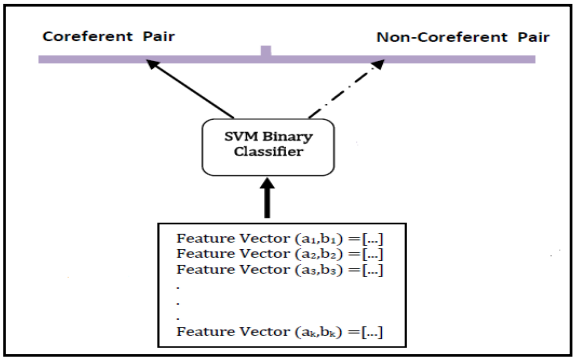
In this step, we constructed Feature Vectors foe each resource pair of RDF triple store1-i and RDF triple store 2-i. therefore we had a k*d-dimensional vector for each resource pair. Each dimension shows approximate textual similarity score between values of property P_i of resources of type C_i, have been calculated by i_{th} Similarity Function in the matchers. In machine learning terminology, these feature vectors forms the basis for classifying the resource pair as a "coreferent" or "non-coreferent".



Proposed Method: Steps 2 to 4

Step 5: Learning Resource Coreferensing

So far, we have proposed three similarity functions to measure the similarity between the selected sets of RDF resources properties. These measurements are not normalized and how well each of them contributes to the final similarity between resources is not clearly known. It is necessary to weight properties according to the true similarity between resources. To tackle the issue of integrating different measurements, we employ a machine learning approach. In step 2, we have constructed feature vectors for all resource pairs. Now, we create a set of coreferent resource pairs with positive labels and a set of non-coreferent resource pairs with negative labels. A binary classifier based on SVM [19], is trained by using different similarity measurements. This binary classifier acts as a parsing function, taking a resource pair as input and generating decision value as output. If it generates positive values, the two input resources are regarded as matched; otherwise, unmatched. Then *owl:sameAs* links can be generated between matched pairs of resources.



Proposed Method: Step 6

Based on the discussion in [18], we choose SVM as the classification model in this study. The SVM technique is able to learn from small training sets of high-dimensional data with satisfactory precision. In addition the hypothesis learned by the classifier must be independent of relative sizes of the positive and negative training sets, since the proportion of matched pairs in the training set is likely to be much higher than in the actual datasets where coreferences are detected. SVM classifiers provide the following classification function [18]:

$$f(q) = \sum_{i=1}^{l} \alpha_i \, y_i \, K(p_i, q) + b$$

Where K(p,q) is a kernel function used for mapping features into different spaces, α_i is the Lagrangian coefficient of the i_{th} training resource pair P_i, $y_i \in \{+1,-1\}$ is the label of the training resource pair. Given a test resource pair q, we regard q as a coreferent pair if *f(q)> 0*. Besides, this, as *f(q)* indicates the distance of q from the optimal hyperplane, we can use this value to evaluate the confidence level of the pair being coreferent [18]. That is, if *f(r) > f(q)*, then r is more likely to be a coreferent pair than q. Ultimately the *Owl:sameAs* links can be set between coreferent resource pairs, the positive labeled in the result of SVM classification.

IV. EXPERIMENTAL EVALUATION

A. *Datasets:*

We ran our coreference resolution method for Rexa and AKT EPrints dataset pair from the domain of scientific publications. Our datasets were structured according to the SWETO-DBLP [iv] ontology, which extends the FOAF ontology[v], and contained instances of three types*: foaf:Person, opus:Article and opus:Article_in_Proceedings*. The last two, are subclasses of the class *opus:Publication.*

AKT EPrints archive[vi] : This dataset contains information about papers produced within the AKT research project.

Rexa dataset [vii]: This dataset extracted from the Rexa search server, which was constructed in the University of Massachusetts using automatic IE (Information Extraction) algorithms.

Table: 1 Datasets Used in Experiments

Direct Classes	Article	Article_In_Proceedings
Selected Properties	rdfs:label opus:year opus:journal_name opus:volume	rdfs:label opus:year opus:book_title
Number Of Resources		
AKT EPrints	39	245
Rexa	1618	2103

B. *Experimental Methodology and Results:*

We used the LIBSVM [viii][20], a good implementation of the SVM classifier, for learning whit radial basis function as kernel function. There are two parameters for an RBF kernel: C and gamma. Best parameter selection performed using the grid-search and cross-validation. We used 5-fold cross-validation for Article_In_Proceeding class and 4-fold cross-validation for Article class. For V-fold cross-validation, we first divided the training set into v subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining v-1 subsets. With cross-validation, each instance of the whole training set is predicted once so the cross-validation accuracy is the percentage of data which are correctly classified. Traditionally, the quality of the coreferencing output is evaluated by comparing it with the set of true corefrencing and calculating the precision and recall metrics. Our method would allow to estimate the quality of a set of mappings without possessing labeled data or involving the user. Under these conditions, it is not possible to calculate the precision and recall. Therefore, the results are reported as accuracy measure:

a. ACCURACY = (TP + TN) / (TP + TN + FP + FN)

TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative

Table: 2 Results of our Algorithm

Measure	Accuracy on class Article	Accuracy on class Article_In_Proceedings
Our Method	Cross Validation Accuracy : 81.82% Classification Accuracy: 95.45%	Cross Validation Accuracy : 95.2% Classification Accuracy: 93.87%

Table: 3 Results of Previous Methods

Measure / Method	Precision
KNOFUSS	0.92
RDF-AI	0.95

V. CONCLUSION AND FUTURE WORKS

In this paper, we discussed the problem of coreference resolution in the linked data environment. A new method, based on supervised learning has been developed to address this issue. In our method, a binary classifier based on SVM, trained to classify resource pairs as coreferent or non coreferent. Although we assumed that input datasets have been described with a common ontology, our algorithm is flexible to address the issue of different ontologies. If ontologies

differ, first an automatic schema matching systems have been used and the same procedure be done for mapped class resulted by schema matching tools. So the future work of this study includes this state. Another area for future work lies in applying the method on larger datasets. In this state, we must use the clustering methods.

VI. REFERENCES

[1]. T. Berners-Lee. Linked Data - Design Issues. http://www.w3.org/DesignIssues/LinkedData.html

[2]. C. Bizer, T. Heath, T. Berners-Lee. Linked data-The story so far. 2009, International Journal on Semantic Web and Information Systems (IJSWIS), 2009, 5(3):1-22.

[3]. A. Nikoliv, M. Aquin, E. Motta. Unsupervised data linking using a genetic algorithm. Technical Report kmi, 2011.

[4]. A.K. Elmagarmid, P.G. Ipeirotis and V.S. Verykios. Duplicate record detection: A survey. 2007, IEEE Transactions on Knowledge and Data Engineering, 19(1):1-16.

[5]. W. Hu, J. Chen, Y. Qu. A Self-Training Approach for Resolving Object Coreference on the Semantic Web. Proceedings WWW 2011, 87-96.

[6]. X. Dong, A. Halevy, J. Madhavan. Reference reconciliation in complex information spaces. 2005, ACM SIGMOD international conference on Management of data.

[7]. J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Discovering and maintaining links on the web of data. ISWC 2009.

[8]. H. Glaser, I.C. Millard, W.K. Sung, S. Lee, P. Kim, B.J. You. Research on linked data and coreference resolution. Technical report, University of Southampton, 2009.

[9]. Y. Raimond, CH. Sutton, M. Sandler. Automatic Interlinking of Music Datasets on the Semantic Web. Linked Data on the Web workshop, 2008.

[10]. F. Scharffe, Y. Liu, and C. Zhou. RDF-AI: an architecture for RDF datasets matching, fusion and interlink. Proceedings of workshop on Identity, reference, and knowledge representation (IJCAI), 2009.

[11]. A. Nikolov, V. Uren, E. Motta, A. de Roeck. Handling instance coreferencing in the knofuss architecture. 5th European Semantic Web Conference (ESWC 2008).

[12]. J. Volz, Ch. Bizer, M. Gaedke, G. Kobilarov. Silk – a link discovery framework for the web of data. Workshop on Linked Data on the Web (LDOW 2009), 18th International World Wide Web Conference (WWW2009), 2009.

[13]. K. Elmagarmid, P. G. Ipeirotis, V.S. Verykios. Duplicate record detection: A survey. Knowledge and Data Engineering, IEEE Transactions on, Vol. 19, No. 1, pp: 1-16, 2007.

[14]. W. Cohen, P. Ravikumar, S.E. Fienberg. A comparison of string distance metrics for name-matching tasks. Proceedings of IJCAI-03 Workshop on Information Integration, 2003, 73–78.

[15]. W. Winkler. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Bureau of the Census, 1999.

[16]. M. Bilenko, R.J. Mooney. Adaptive duplicate detection using learnable string similarity measures. Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 39-48, 2003.

[17]. L. Ding, J. Shinavier, Z. Shangguan, D. McGuinness. SameAs networks and beyond: Analyzing deployment status and implications of owl:sameAs in linked data. Proceedings of ISWC, pages 145–160, 2010.

[18]. C.C. Chang, C.J. Lin. LIBSVM: a library for support vector machines, 2001.

[19]. V.N. Vapnik. The nature of statistical learning theory. Springer Verlag, Heidelberg, 1995.