A Comparison of Two Ontology-based Semantic Annotation Frameworks

Quratulain Rajput and Sajjad Haider
Artificial Intelligence Lab, Faculty of Computer Science
Institute of Business Administration
Karachi, Pakistan
{qrajput,sahaider}@iba.edu.pk

Abstract. The paper compares two semantic annotation frameworks that are designed for unstructured and ungrammatical domains. Both frameworks, namely ontoX (ontology-driven information Extraction) and BNOSA (Bayesian network and ontology based semantic annotation), extensively use ontologies during knowledge building, rule generation and data extraction phases. Both of them claim to be scalable as they allow a knowledge engineer, using either of these frameworks, to employ them for any other domain by simply plugging the corresponding ontology to the framework. They, however, differ in the ways conflicts are resolved and missing values are predicted. OntoX uses two heuristic measures, named level of evidence and level of confidence, for conflict resolution while the same task is performed by BNOSA with the aid of Bayesian networks. BNOSA also uses Bayesian networks to predict missing values. The paper compares the performance of both BNOSA and ontoX on the same data set and analyzes their strengths and weaknesses.

Keywords: Ontology, Information Extraction, Bayesian Network, Machine Learning, Semantic Annotation.

1 Introduction

A large amount of useful information over the web is available in unstructured or semi-structured format. This includes reports, scientific papers, reviews, product advertisements, news, emails, Wikipedia, etc [1]. Among this class of information sources, a significant percentage contains ungrammatical and incoherent contents where information is presented as a collection of words without following any grammatical rules. Many of the popular retail sites such as eBay[1] and craigslist[2] fall into this category. These websites are made up of user-provided data, called *posts,* which contains useful information but due to lack of semantics are not easily searchable. Moreover, the automatic extraction of required information from such posts is also a big challenge [2]. The semantic web technologies, such as OWL/RDF, provide a major advancement to handle this challenge [3] as they aid in the semantic annotation of new and existing data sources.

[1] www.ebay.com
[2] www.craigslist.org

The semantic annotation process for existing data sources consists of many phases including information extraction, knowledge management, storage of extracted data in RDF/OWL, and user interfaces [4]. Much of the research, however, has been focused on extracting information from varying type of data sources. Leander et al. [5] provides a good survey of the techniques used in this area. Recently, a lot of research has been done on ontology based semantic annotation as it helps in making the application independent from domain knowledge and easily scalable. Few of the important contributions in this area are BYU [6], MnM [7], OntoX [8] and BNOSA [9] [10] which is an extension of our previous work OWIE [11] and E-OWIE [12].

This paper compares the performance of ontoX and BNOSA frameworks. These frameworks are selected due to their similarities in the way they exploit ontology for knowledge building, rule generation and information extraction. They, however, differ in the way conflict resolution is handled. ontoX uses heuristic measures while BNOSA uses Bayesian networks to resolve conflicts in case multiple values are extracted for a single attribute.

The rest of the paper is organized as follows. Section 2 describes ontoX and BNOSA while Section 3 compares the similarities and differences between them. The performance of both frameworks on the same data set is analyzed in Section 4. Finally, Section 5 concludes the paper and provides future research directions.

2 Selected Semantic Annotation Framework

2.1 ontoX: Ontology Driven Information Extraction

Burcu et al. [8] presented a framework to extract relevant data from existing web documents. The methodology suggests an ontology-driven information extraction process. It builds a domain specific ontology which is utilized during the information extraction phase. A tool, ontoX, was implemented based on the proposed scheme. The ontoX system consists of three main modules: i) Ontology Management Module, ii) Rule Generation Module and iii) Extraction Module.

Ontology Management Module (OMM): This module builds a domain-specific ontology to be used for information extraction. Different constructs of the ontology such as classes, object properties and data-type properties (also called attributes) as well as additional properties defined in the owl:AnnotationProperty element are utilized during the information extraction phase. The additional properties based on context keywords, constraints, quality properties and temporal properties are defined for each construct in the ontology.

- *Keywords:* The context keywords help in finding the location of relevant information in a corpus.
- *Constraints:* Constraints are used to narrow down the range of possible values belonging to an attribute. Standard data types are used to define such ranges. For example, *int* data type specifies that the value is of integer type, *float* data type specifies that the value is of float type, and so on. The only exception is the *string* data

type which is specified as xsd:Name[3] with the aid of named-entity-probability heuristic.

- *Quality properties:* Quality properties enrich the constructs in ontology by creating two additional properties. The first one is *confidence-level*, which takes values between [0, 1]. It indicates the level of confidence of the ontology engineer that the construct is relevant. It also helps in situations when the same value is assigned to two different attributes. If such a situation arises, the property with the higher confidence level is considered the winner. The other property is *relevance*, which takes one of the two values {true, false}. It tells the system that a user is not very interested in this property, but if found then extract it because the construct is part of the domain of interest.
- *Temporal properties:* Temporal properties provide two kinds of services: temporal extraction and change management. With temporal extraction property, a user can state if she wants her input data to be extracted and can suggest valid-time-begin and valid-time-end properties for every construct defined in the ontology. With change management property, users can provide suggestions regarding out-of-date concepts if they do not appear in the corpora anymore.

Rule Generation Module (RGM): This module is responsible for generating extraction rules for each data type defined in the ontology. The rules are used to identify possible values of each attribute.

Extraction Module (EM): This module applies the rules generated by RGM to extract values corresponding to each attribute. The extraction module consists of two main steps: pre-processing and extraction. The pre-processing step removes the stop words (such as “and”, “it”, “at”, “to”, etc.), and finds the location of the pre-defined data types. The extraction step assigns the located data to the corresponding attribute. If more than one keyword is assigned to an attribute, then data is searched/located in the neighborhood of all the keywords. If more than one value is assigned to an attribute, ontoX uses a heuristic, level of evidence, to resolve this conflict. In case of same value being assigned to more than one attributes, ontoX uses another heuristic, confidence level, to resolve this conflict.

2.2 BNOSA: Bayesian Network and Ontology based Semantic Annotation

BNOSA (Bayesian Network and Ontology based Semantic Annotation) is a semantic annotation framework that is designed to extract relevant information from unstructured and ungrammatical domains such as craigslist [9]. It utilizes ontology as well as Bayesian networks to perform information extraction and semantic annotation tasks. The information extraction in BNOSA is conducted in two phases:

Phase-I: Similar to ontoX, this phase utilizes different constructs of ontology such as classes, object properties and data type properties for information extraction. It also stores additional information in owl:AnnotationProperty elements. The additional information consists of context keywords and value constraints. Unlike ontoX, BNOSA does not define Quality and Temporal properties.

[3] The value space of xsd:Name is a set of all strings that match the Name production of XML 1.0.

– *Context Keywords:* The context keywords help in finding the location of relevant information in a corpus.
– *Constraints*: This feature is also similar to the corresponding feature in ontoX except the way *string* data types are handled. BNOSA applies simple pattern matching rules for all instances of a string type attribute. All such instance values are stored in the comment section of the ontology.

Once information is specified in the ontology, this phase generates rules for each data type at run time. The context keywords are located first and then with the help of these rules, data within the neighbourhood of these context keywords is searched. The extracted data is then assigned to the corresponding attribute.

Phase-II: If more than one value is assigned to an attribute or no value is extracted at all then Bayesian networks are used in Phase-II for conflict resolution and missing value prediction. Phase-II is mainly divided into two modules: Bayesian network learning module and prediction module.

– *Learning Module:* This module first performs data cleaning and data pre-processing (discretization of continuous data and removal of anomalies), and then learns the probabilistic relationships that exist among the attributes by learning the structure and parameters of a BN.
– *Prediction Module:* If there are missing and/or conflicting values in the extracted data set then this module uses Bayesian inference mechanism to predict missing values and to resolve conflicts. In case of missing values, all the non-missing/non-conflicting values are considered as hard evidence and the posterior marginal probability of the missing attribute is computed. The value with the highest probability is considered the missing value if it satisfies a particular threshold value. In case of multiple values belonging to an attribute (conflict resolution), the attribute's value is considered as missing and all the non-missing/non-conflicting values are entered as hard evidences in the BN. The posterior marginal probability of this attribute is computed next. Among the multiple values, the value with the highest posterior probability is selected as the winner and is assigned to the corresponding attribute.

3 Comparative Analysis

This section identifies the similarity and dissimilarity of ontoX and BNOSA in terms of their knowledge specification mechanisms and information extraction processes.

3.1 Ontology Specification:

Both approaches use ontology to define the domain knowledge as well as some additional information that is utilized during the information extraction phase.

The similarities in ontology specification are:

– The comment section of the owl:AnnotationProperty is used to define and store the context keywords. It must be stated, however, that finding/defining all relevant

keywords requires manual analysis of documents collection which in itself is a labor intensive and time consuming process.
- Ontology constructs are used to define domain knowledge.
- The ranges of possible values of an attribute are constrained by defining its data type such as int, float, date, etc.

The differences in ontology specification are:
- ontoX uses owl:AnnotationProperty to define few additional properties as well. This includes quality and temporal properties. Such properties are not defined by BNOSA.
- To handle a string type attribute, ontoX defines its range through xsd:Name property if the attribute has large number of instances. In case of only few instances, it uses the enumeration construct. BNOSA, on the other hand, defines all possible instance values in the comment section without specifying context keywords.

3.2 Information Extraction:

To extract values of each attribute, rules need to be generated. Both approaches define ranges of attributes' values as data types provided by OWL. The rules, thus, are generated according to the corresponding primitive data type. These rules, defined in the form of regular expressions, are used to extract values from the text.

The similarities in the extraction process are:
- Both approaches try to locate an attribute value within the neighborhood of the corresponding context keywords.
- Each attribute can have more than one keywords associated with it therefore more than one value can be extracted for an attribute.

The differences in the extraction methodology are:
- ontoX considers the neighborhood of a context keyword as the area ranging from its previous keyword to its next keyword. BNOSA, on the other hand, considers a fixed number of characters surrounding the keyword as its neighborhood.
- If one value is found to be relevant to more than one attribute than ontoX resolves it using confidence level, while BNOSA assigns this value to each of the attributes.
- To select one value from more than one extracted value, ontoX computes the level of evidence, while BNOSA resolves it using the Bayesian network.
- BNOSA also uses Bayesian networks to predict missing values which is not performed by ontoX.
- OntoX also has an ontology change detection mechanism which is not implemented in BNOSA yet.

4 Experimental Results

This section evaluates the extraction results of ontoX and BNOSA when they are applied on the same data set. To compare the results, the same data set has been selected as used by Burcu et al. [8], which is a collection of digital camera reviews available on a retail

website[4]. A sample data is shown in Table 1. The evaluation is done on the basis of recall and precision values. The values are computed as:

$$Recall = \frac{C}{N}, \quad Precision = \frac{C}{C + I}$$

where N is the number of values extracted from the document, C is the number of correctly extracted values, and I is the number of incorrectly extracted values.

Table 1. Camera data available in reviews

Mega Pixel	Optical Zoom	Display	Storage Medium	Movie Format	Battery
5.0		2.5	SD	QuickTime	
5.0	12	1.8	SD	QuickTime	Lithiumion
	3	2.0	SD		AA
7.1	4	2.0			
6.0	12	2.0			
5		1.8	XD	AVI	lithium
5	3			VGA	
	4		FC		

As discussed earlier, BNOSA uses Bayesian network to predict missing values and to resolve conflicts. But before being used for this purpose, a Bayesian network needs to be learned from the available data. The first step in this process requires discretizing numerical values. The discretized data is then used to train the Bayesian network. Fig. 1 shows the Bayesian network learned from the discretized camera data set.

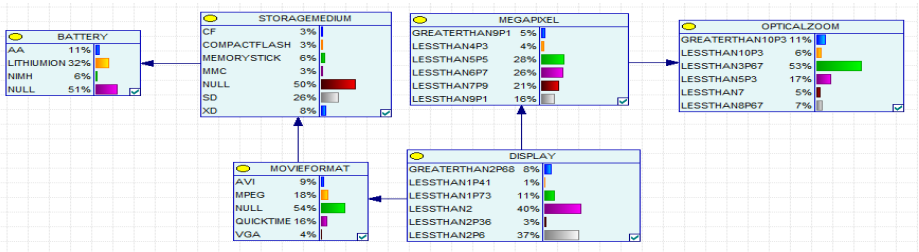

Fig. 1. Bayesian network learned model for digital camera

Once a BN has been learned, it is used for conflict resolution and for missing value prediction as explained in Section 2.2. ontoX, on the other hand, uses level of evidence for conflict resolution. Unlike BNOSA, there is no mechanism in ontoX to predict the missing values. To keep the results and the analysis of the experiment consistent, the prediction feature of BNOSA is not utilized during the experiments. Similarly, when there are a large number of possible instances belonging to a string type attribute, such as Model of a camera, ontoX models these instances through xds:Name property and uses a heuristic to resolve conflicts. The results reported in [8] show a good rate of recall and precision values for such string data types. BNOSA, on the other hand, stores all the possible values in the comments sections. Despite some obvious limitations, the approach used by ontoX - for large number of instances belonging to a string data type - is more generic and is superior in terms of its expressiveness. The method employed by BNOSA, however, would still generate better results as it stores all the instances in the

[4] http://www.steves-digicams.com/

comments section of the ontology. It is for these reasons; string data types with large number of instances are not compared in this analysis. String type attributes with small number of instances, however, are still being considered because both ontoX and BNOSA store them in the ontology albeit differently.

Table 2. Extraction results using ontoX

	Number of fact	Correctly identified facts	Incorrectly identified facts	Recall	Precision
Megapixel	137	70	63	0.51	0.52
Optical zoom	124	105	22	0.84	0.82
Display size	113	93	23	0.82	0.80
Storage medium	61	15	56	0.25	0.22
Movie format	56	41	59	0.73	0.41

Table 3. Extraction results using BNOSA

	Number of fact	Correctly identified facts	Incorrectly identified facts	Recall	Precision
Megapixel	137	121	7	0.88	0.95
Optical zoom	125	104	17	0.83	0.86
Display size	113	108	19	0.96	0.85
Storage medium	61	46	5	0.75	0.90
Movie format	55	50	52	0.91	0.49

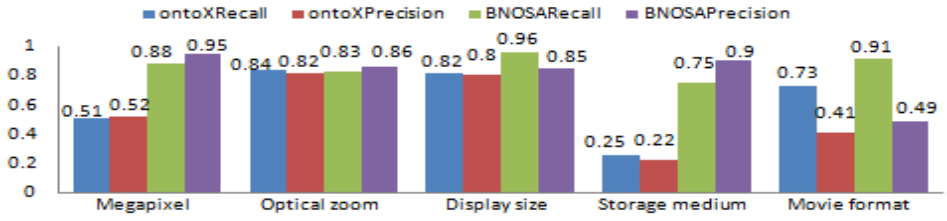

Fig. 2. Graphical view of extraction result

The results of the extraction process by ontoX and BNOSA are shown in Table 2 and Table 3, respectively. The corresponding graphical representation of the results is shown in Fig. 2. It can be seen from the tables and the figure that in general BNOSA performs better than ontoX and, in few cases, the difference in the recall and precision values is significantly higher. For example, the recall and precision values of ontoX for MegaPixel is within the range of 0.5. In contrast, BNOSA produces recall and precision values for the same attribute in the range of 0.9. Other entries can be read and interpret in a similar fashion.

This difference in the performance is primarily due to different heuristics/methods used by these approaches. For instance, when selecting the neighborhood area of a keyword, BNOSA considers a fixed number of characters on either side of the keyword. ontoX, on the other hand, uses the whole area between a keyword and its left and right neighboring keywords. This, in some cases, may produce more conflicts. Another reason might be the use of level of evidence in ontoX. Being a user-defined metric/heuristic, an inaccurate specification may degrade the performance of the whole ontoX framework. In contrast, BNOSA uses Bayesian networks which learn the probabilistic relationship from

the available data set and can be considered more robust when compared to a user-defined metric. It must be stated, however, that BN learning requires the availability of a large data set to accurately reflect the probabilistic relationship that exist among the variables in a problem domain.

5 Conclusion

The paper compared the information extraction capabilities of two semantic annotation frameworks, namely BNOSA and ontoX. Both frameworks model and store knowledge about a problem domain in an ontology and uses it during the information extraction phase. They, however, differ in ways, information is located, conflicts are resolved and missing values are predicted. Experiments were conducted on a sample data set to analyze the strengths and weaknesses of both frameworks. The results suggest that BNOSA performs significantly better than ontoX on the selected data set. The statement, however, cannot be generalized unless thoroughly tested on many data sets and that too of a much larger size as compared to the one selected in this study. The comparison of BNOSA with ontoX and few other information extraction techniques on multiple large data sets is one of the future research directions.

Acknowledgements. The first author is extremely grateful to Dr. Burcu Yildiz for helping her in running ontoX as well as for sharing the camera data set which made the comparison reported in this paper possible.

References

1. R. Ittersum and E. Spalding, "understanding the difference between structured and unstructured documents," 2005.
2. M. Michelson and C.A. Knoblock, "Semantic annotation of unstructured and ungrammatical text," *Proceedings of the 19th international joint conference on Artificial intelligence*, Edinburgh, Scotland: Morgan Kaufmann Publishers Inc., 2005, pp. 1091-1098.
3. G. Antoniou and F.V. Harmelen, *A semantic Web primer*, MIT Press, 2004.
4. L. Reeve and H. Han, "Survey of semantic annotation platforms," *Proceedings of the 2005 ACM symposium on Applied computing*, Santa Fe, New Mexico: ACM, 2005, pp. 1634-1638.
5. A.H.F. Laender, B.A. Ribeiro-Neto, A.S.D. Silva, and J.S. Teixeira, "A brief survey of web data extraction tools," *Sigmod Record.*, vol. 31, 2002, pp. 84-93.
6. D.W. Embley, C. Tao, and S.W. Liddle, "Automating the extraction of data from HTML tables with unknown structure," *Data and Knowledge Engineering.*, vol. 54, 2005, pp. 3-28.
7. M. Vargas-Vera, E. Motta, J. Domingue, M. Lanzoni, A. Stutt, and F. Ciravegna, "MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup," 2002, pp. 379--391.
8. B. Yildiz and S. Miksch, "ontoX-A Method for Ontology-Driven Information Extraction," *Lecture Notes in Computer Science*, vol. 4707, 2007, pp. 660-673.
9. Q. Rajput and S. Haider, "BNOSA: A Bayesian Network and Ontology based Semantic Annotation Framework," *submitted*.
10. Q. Rajput,"Semantic Annotation Using Ontology and Bayesian Networks," *Advances in Artificial Intelligence*, 2010, pp. 416-418.
11. Q.N. Rajput, S. Haider, and N. Touheed, "Information Extraction from Unstructured and Ungrammatical Data Sources for Semantic Annotation," *International Journal of Information Technology*, vol. 5(3), 2009.
12. Q.N. Rajput and S. Haider, "Use of Bayesian Network in Information Extraction from Unstructured Data Sources," *International Journal of Information Technology*, vol. 5(4), 2009.