Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007)

Finding Similarity Relations in Presence of Taxonomic Relations in Ontology Learning Systems

A. R. Vazifedoost, F. Oroumchian, M. Rahgozar

***Abstract*—Ontology learning tries to find ontological relations, by an automatic process. Similarity relationships are one of non-taxonomic relations which may be included in ontology. Our idea is that in presence of taxonomic relations we are able to extract more useful non-taxonomic similarity relations. In this paper we investigate the specifications of an implemented system for extracting these relations by means of new context extraction method which uses taxonomic relations.**

I. INTRODUCTION

ONTOLOGY is informally a collection of concepts with some relations between them. These relations consist of two main categories, i.e. taxonomic and non-taxonomic relations. There have been many researches on making ontology learning automatic through learning methods. Although, it seems that most of these works were conducted in learning taxonomic relations like "IS-A". Although the non-taxonomic relations are empirically the distinctions point between traditional thesauruses and ontologies and therefore have a lot of importance in ontology construction. While there are also some works on non-taxonomic relations, a long distance is remained before to be matured in this field. Similarity relation between two concepts is one of these important non-taxonomic relations.

However apart of its intrinsic importance, we need this kind of relations in the Human Plausible Reasoning (HPR) based systems. The usage of HPR systems is investigated in various fields like, Information Retrieval [15], Document Clustering [16], Question answering [17] and other IR related fields. These systems benefit from a knowledgebase which is ontology by its nature. Similarity relations (or in HPR terminology "SIM" relations) have an important role in inferences which is made by the inference engine of these systems. Thus, in our general task of automatically learning a knowledgebase for these systems we focus here on learning similarity relations from text.

A common approach in learning similarity relationships is using the context of a concept for measuring its similarity with other concepts [18][19][20][21][22]. In fact, this approach gives some features to concepts that make them comparable. This approach is called distributional similarity. The distributional similarity approach states that words which occur within similar contexts are also semantically similar. As a concrete similarity measure we compare a pair of weighted context feature vectors that characterize two words in a text.

Cimiano et.al [18] applied the Formal Concept Analysis and modeled the context of a concept as a vector representing syntactic dependencies. They also applied cosine coefficient for measuring similarity of vectors. Researchers [19] did a similar work with Text2Onto software in which they extracted the context of the concepts and represented them as vectors by using shallow parsing methods. They used Jaccard coefficient for measuring the similarity of the vectors. Also, Sanderson & Croft in [20] used conditional probability of co-occurring terms in the same document. In fact, they used the document in which a concept occurs as the context of that concept. Therefore similarity measure is determined by cosine measure on two documents. Two co-occurred concepts would be more similar in this approach. Also, Pum-Mo Ryu & Key-Sun Choi in [21] used a measure based on the internal context of a concept. If two concepts share many common words, they share common characteristics in a given domain. Other approaches discussed in [22] consider words within a window or neighborhood of a concept as the context of that concept.

The application of measuring similarity in previous works was mostly in clustering algorithms for constructing taxonomic structure, within ontology. Therefore their purpose wasn't extracting an explicit named relation with a label, e.g. Similar-to, between two concepts. Therefore, the similarity was implicit in the clusters.

We explored the reverse direction instead because we need to keep similarity relationships explicit. In the reverse direction we will create the taxonomy first by using the techniques other than clustering. Based on this idea the context of a concept could be determined by its taxonomically related concepts. A similar idea was applied in [1] for integrating two ontologies. The relation that was used in [1] for determining the context was restricted to "IS-A" relationship. Additionally, we use the "Attribute-Of" relations which are broader and more helpful to determine the context. Also in that work, the confidence of relations

Manuscript received November 5, 2006. This work was supported in part by Iran Telecommunication Research Center (ITRC).

A. R. Vazifedoost is with the Database Research Group, faculty of ECE, School of Engineering, University of Tehran, Tehran, Iran (e-mail: a.vazifedoost@ece.ut.ac.ir).

F. Oroumchian is with the College of IT, University of Wollongong in Dubai and member of Control and intelligent processing center of excellence of University of Tehran (e-mail: farhado@uow.edu.au).

M. Rahgozar is member of the Control and Intelligence Processing Center of Excellence, faculty of ECE, School of Engineering, University of Teshran,Tehran, Iran (e-mail: rahgozar@ut.ac.ir).

wasn't addressed but we include that directly in similarity measure.

The remaining of this paper is organized as follows. In section 2 our proposed method is discussed in more detail, while in section 3 the architecture of the system is described. An overview of the experiments is presented in section 4 and finally conclusion is presented in section 5.

II. METHOD

Taxonomic relations are one of the most important relations in ontology. These kinds of relations are usually found in the initial steps of ontology learning process. Non-taxonomic relations often would be found after the taxonomic relations are discovered. Those relations describe relationships such as causal, related-to, possession, and etc. Similarity relations are kind of non-taxonomic relations. These relationships imply that two concepts have similar shared understanding in a specific domain. Our aim is to find such relationships between concepts.

First we should decide what we exactly mean by similarity of two concepts. We define this formally by Jaccard's Coefficient [19]. This measure is a mean for declaring the similarity of two concepts based on their features. Jaccard's Coefficient is defined as:

$$Jaccard - Sim\ (A,B) = \frac{|Features(A) \cap Features(B)|}{|Features(A) \cup Features(B)|} \quad (1)$$

Where A and B are two concepts, and *Features (A)* stands for the set of features which belongs to concept A. That is true also for concept B. This measure estimates the commonness between the features of two concepts. The more features in common the more similar they are. We will use an adoption of this strategy which will be discussed in section 3.

Now, the main question is that what we should consider as the features of a concept in ontology learning system. We use the taxonomic relationships of ontology for this purpose. In fact, what we consider as the features of a concept are the concepts which are related to that concept through taxonomic relationships. Figure 1 illustrates this matter.

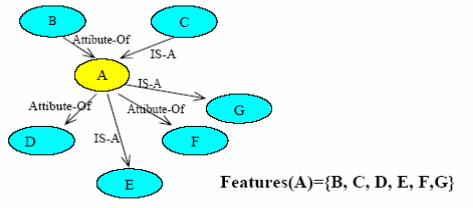

Figure 1-Features of Concept "A"

In this figure, we see a concept *A* which is connected to other concepts, through "Attribute-Of" and "IS-A". It may be either the source or the destination of those relations. However we only consider their connections, regardless of its direction. Therefore all concepts *B* through *G* will be fitted in the feature set of A.

We assume that there are some taxonomically related concepts which are already detected by ontology learning approaches. The taxonomic relations we consider are "IS-A" and "Attribute-of" relations. We will explain the methods we've used for finding these relations, in the next section.

A notable point in our approach is using a similarity measure in reverse of ordinal usage. The ordinal usage refers to application of similarity measures in clustering concepts and thus constructing taxonomic structures [2]. In that case, the similarity measure between two concepts determines how to place them in the appropriate clusters.

We go in opposite direction which considers constructing taxonomy by using other methods instead of clustering. This is noticeable because we denote similarity relation as an explicit and labeled relationship and not an implicit one which is hidden but glues the members of the clustered concepts together in the clustering methods.

III. ARCHITECTURE

The system consists of three main subsystems as depicted in figure 2. First two subsystems are: *IS-A Relation Extractor* and *Attribute-Of Relation Extractor,* which are responsible for extracting taxonomic relations. Next, we have the *Similarity Relation Extractor* which acts based on the inputs from two previous subsystems. We will explore each subsystem in more detail in the following sections.

A common requirement between before extracting taxonomic relations is to obtain a preprocessed corpus. We use General Architecture for Text Engineering (GATE) [7] for this preprocessing task. This task consists of: tokenizing, stemming, sentence splitting and Part-of-Speech tagging.

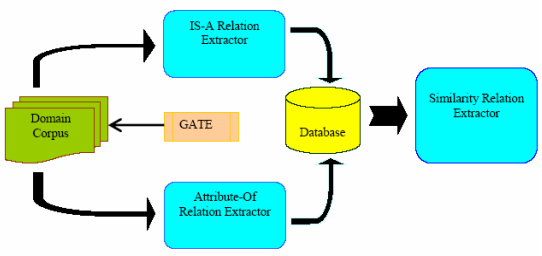

Figure 2 The architecture of the system

A. IS-A Relation Extractor

The detection of "IS-A" relation has been investigated more than any other relation in the ontology learning domains. There are two main categories of approaches for this purpose. Extracting lexico-syntactic pattern which was first proposed by Hearst [3] [4] and clustering methods for constructing taxonomy of concepts which are based on statistical methods proposed in [5][6].

Here we will not explore statistical methods in detail. Instead we will focus on using lexico-syntactic patterns. The

patterns that we use are depicted in figure 3.

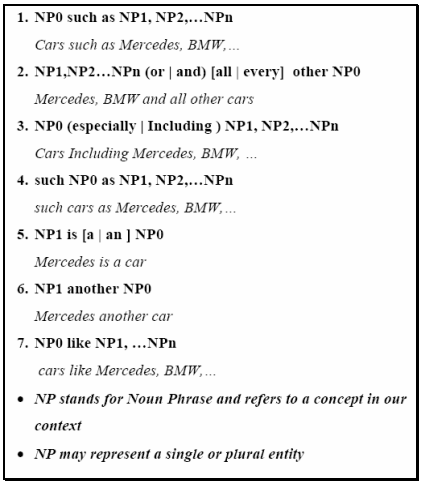

Figure 3 Patterns for extracting "IS-A" relations

These patterns are translated to the JAPE syntax. JAPE is the pattern matching component of GATE. Then, JAPE grammars are applied to the corpus and the matched patterns are determined. When these patterns match with some portion of the text, we can then infer: $\forall NP_i, 1 \le i \le n, \quad IS-A(NP_i, NP_0)$. The extracted relations between each two concepts are finally stored in the database.

Also, we need to specify a confidence value to each extracted relation. The confidence value could be assigned after all of the "IS-A" relations have been stored in the database. The formula we use is as follows:

$$\begin{cases} Conf(GC,Ci) = \dfrac{freq(GC,Ci)}{freq(GC)} & freq(GC) \neq freq(GC,Ci) \\ Conf(GC,Ci) = 0.1 & freq(GC) = freq(GC,Ci) \end{cases} \quad (2)$$

Where the *Conf(GC,Ci)* is the confidence value of a "IS-A" relation. Also *Ci* and *GC* are two concepts for which we have "*Ci* IS-A *GC*" and *freq (GC,Ci)* is the number of times two concepts *GC* and *C* matched through the patterns. Finally, *freq(*GC) is the frequency number of *GC*. If the *freq (GC)* is equal to the *freq (GC,Ci)* then 0.1 is assigned to the confidence. That is to prevent the confidence value to become 1 when *freq(GC) = freq (GC,Ci)*. For example when *freq(GC)* and *freq(Ci)* are both equal to 1 which means just once they have been seen together (just one evidence). We chose to assign a constant value equal to 0.1 to all of these relations. The constant value of 0.1 has been chosen because experimentally it has been observed that it provides a good measure for confidence of such relations in our corpus.

B. Attribute-Of Relation Extractor

We use the approach described in [8][9][10] in using patterns for finding "*Attribute-of*" relations. Although, this approach was used in those works for extracting *whole-part* relations, but these relations are general and *whole-part* relations are just one kind of extracted relation. Therefore we name the extracted relations "Attribute-Of" in order to cover all of them. Although [11][12][13] report efforts in classifying these relations into more detailed ones but here the "Attribute-Of" relation provides sufficient semantic depth for our work.

The approach used for extracting Attribute-Of Relations is similar to what we described before for "IS-A" relations. Figure 4 depicts the "Attribute-Of" patterns.

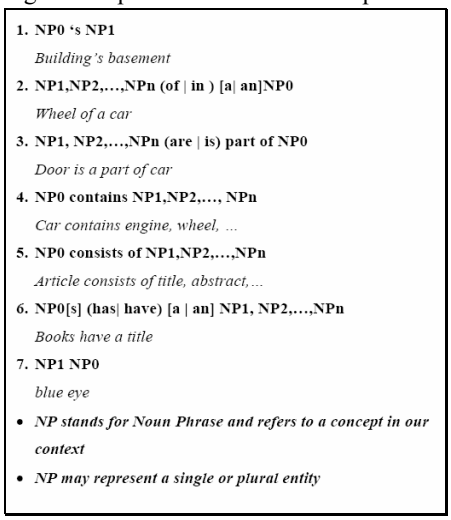

Figure 4 Patterns for "Attribute-Of" relations

These patterns are translated to JAPE grammars as well and then applied to the corpus. When these patterns match with fragments of the text, then one could infer: $\forall NP_i, 1 \le i \le n, \quad Attribute-Of(NP_i, NP_0)$. The extracted relations are stored in the database.

Again we need to assign a confidence value to these relations in a way similar to the "IS-A" relations. The formula for confidence is as follows:

$$\begin{cases} Conf(PC,Ci) = \dfrac{freq(PC,Ci)}{freq(PC)} & freq(PC) \neq freq(PC,Ci) \\ Conf(PC,Ci) = 0.1 & freq(PC) = freq(PC,Ci) \end{cases} \quad (3)$$

Where the *Conf(PC,Ci)* is the confidence value. Also, *Ci* and *PC* are two concepts for which we have "Ci attribute-Of GC" and *freq (PC,Ci)* is the number of times two that concepts *PC* and *C* matched through the patterns. Finally, *freq(P*C) is the frequency number of *PC*. The rational behind adopting 0.1 for the confidence value in the special case is similar to the same for the "IS-A" relations.

C. Similarity Relations Extractor

The last and more important part in our system is the *Similarity Relations Extractor* subsystem. The input to this subsystem is the "Attribute-Of" and "IS-A" relations already stored in the database with their confidence values. For each concept all the related concepts are retrieved as the feature set of that concept as discussed in section 2. A function in database is responsible for retrieving all the features and their confidence values for any given concept. These feature sets are used to calculate the similarity of each pair of concepts.

We use the following formula:

$$Modified\ Jaccard - Sim(A,B) = \frac{\sum_{\forall f,\ f \in Features(A) \cap Features(B)} conf(f)}{\sum_{\forall f',\ f' \in Features(A) \cup Features(B)} conf(f')} \quad (4)$$

Where *conf(f)* represents confidence value of each attribute for each concept. Also, *Features()* is the function which returns the feature set of every concept as well as their confidences.

We use the confidence values of each attribute as the weight of that attribute. This is because all features of a concept don't have the same value and we should reflect this in our measures. Therefore, we don't simply use the count of the members of the union set which is generated from two concept's feature sets. Instead, we use a sum over confidence values of member features in union set. This is the same for intersection of two feature sets where we use the sum of confidences of features presented in the intersection set.

An important issue in this subsystem is the implementation of the pair-wise comparison between the concepts to calculate the similarity between them. It has an order of complexity about $O(n^2)$ which is not acceptable. (e.g. It takes about 260 days for about 17000 concepts in a typical Pentium-IV machine).

However, using database as the storage gives us the chance to reduce the size of comparable concepts. This is through a simple join which holds only the concepts with at least one common feature. The algorithm for finding the similarity measure according to what we discussed so far would be as follows:

1. *C=Find all pairs which have at least one attribute in common*
2. *Foreach (C1,C2) in C do the following:*
   a. *Find the feature set of C1,C2*
   b. *Find the sum of confidence of features in intersection of two previous feature sets.*
   c. *Calculate the Modified Jaccard similarity of (C1,C2) based on previous confidence summation*
   d. *insert the new relation of SIM (C1,C2) with calculated similarity measure in database*
3. *Finish*

The concepts are in various grammatical states, i.e. plural, singular, prefixed etc. In our experiment we treated all of them in the same way. We did so by considering only the stem of a concept when looking for features of that concept. We store the stem of all the concepts along with their surface form. For example for the concept r*elation*, we extract the features of concepts in table 1 as the feature set of the *relation.*

TABLE 1
EXPANDED CONCEPTS WHOSE FEATURE SETS WOULD BE INCLUDED IN THE FEATURE SET OF CONCEPT "RELATION"

discourse relation
industry relations
relation
relation hierarchy
relation slots
relational
relational data
relational database
relational database management system
relational database products
relational database structure
relational databases
relational Markup Language
relational model
relational servers
relational tables
relations
relations firms
relative
relativity
subclass relations
subset relation

This helps to find more similarity between concepts which have common parts. Also it helps finding similarity between ad-hoc concepts. By ad-hoc concepts we mean the concepts that don't exist already in the database. If we have just the stem of one part of that concept it may be possible to measure its similarity to a second concept. For example consider the concept of "*Markup language"* in the ontology. This concept is similar to the concept *"xml".* In order to calculate the similarity between the concepts "*Markup"* and *"xml"*, however the concept "*Markup"* doesn't exists in the ontology. Since the "*Markup Language" is* present there and it contains the stem of *Markup,* when looking for the feature set of the concept "*Markup"* it will include the feature set of *the concept "Markup Language"*. Therefore it could identify the similarity of the concepts "*Markup"* and "*xml"*.

There may be situations where some attributes are repeated more than once in the feature set of a concept. That is because those attributes come from more than one sources. For example, the concept "*relational database"*,

when calling the function *FeatureSet (relational database)* we will get the concept "*size"* twice as a feature with two different confidence values. That is because when we look for features of the concept "*relational database"*, the features of both concepts *"relation*" and "*database*" would be included and the concept "*size*" is in the feature set of both those concepts. However in these situations we only keep the attribute with the greatest confidence value when calculating the similarity measure.

IV. Experiment

Although our experiments are now running and final results aren't available yet but we can explore our experimental environment and give some initial results.

Our test corpus is a portion of INEX2004 [14] corpus. This corpus is originally in xml format but we have removed markups and worked with its free text content. The size of corpus is about 35MB and contains about 1644 paper from IEEE.

The preprocess task including sentence splitting, tokenizing, stemming and POS tagging (Porter algorithm) was performed with GATE. Also, we used JAPE grammars to extract "IS-A" and "Attribute-Of" relations. Table 2, shows some statistics about the extracted taxonomic relations.

TABLE 2
SOME STATISTICAL INFORMATION ABOUT THE CURRENTLY EXTRACTED RELATIONS

Relation Name	Relation count	Concept count	Average confidence
IS-A	11301	9548	0.087
Attribute-Of	53561	22845	0.05

Also we've used SQL Server 2000 as the storage facility. Some functions were implemented directly as user defined functions and stored procedures for improving the speed.

We are working now for extracting similarity relation among about 3464057 pairs of concepts. This is the reduced set's size and just contains the concepts we think may have similarity with each other.

This is a time consuming process and it's working yet. Although some initial results for the sample concept *database* is selected and is depicted in table 3. This concept is selected randomly and without pre estimations.

Table presents all concepts which have a similarity value greater than 0.1 with the concept "*database*". The result seems to be quite reasonable. The concepts who have shared component with *database* are ranked higher by the system. This is because we take into account each part of a multiword concept for extracting its feature set. It means that we increase the member count of the intersection of the feature set for two concepts who share some parts. Although the concept "*database management system"* is ranked lower than "*watermark"*. Therefore we can be sure that there is a trade off between having more shared parts and having more features in common.

TABLE 3
CONCEPTS SIMILART TO *DATABASE*

Concept1	Concept2	Similarity
databases	database	1
database	terrain database	0.603813233
database	database schema	0.558929065
database	database queries	0.511792933
consumer database	database	0.50184292
database	database protection	0.494636127
database	database access	0.440411548
database	database access library	0.324017689
database	virus information database	0.302964223
database	database processing	0.262051963
database	database changes	0.253187694
database technology	database	0.250073974
database processing	database	0.241276648
database changes	database	0.225349067
database	database system	0.181055873
database	outages	0.156714944
database	watermark	0.141398073
database	database management system	0.13935378

V. Conclusions

In extracting similarity relationships from texts a basic decision is how the context of each concept would be determined. Having already extracted taxonomical relations by means of lexico-syntactic patterns we have explored the possibility of using them as a source for finding the context of a concept. The context is determined by the concepts which are connected to a concept through taxonomic relations.

Also we used Jaccard coefficient as the similarity measure between every two feature set. However the Jacquard coefficient was modified to include not only count of features but also the features' confidence values.

The result is in initial states however they seem to be promising about the concepts which share common sub terms. We didn't yet finish the experiment mostly because of time constraints. A detail evaluation of extracted relations will be done after finishing the experiments.

ACKNOWLEDGMENT

This work was supported by Iran Telecommunication Research Center (ITRC) grant program. With sincere thanks to Mrs. Besat Kassaie for her very brilliant ideas which helps us during this work.

REFERENCES

[1] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. "Ontology matching: A machine learning approach". In S. Staab and R. Studer, editors, Handbook on Ontologies in Information Systems. Springer-Velag, 2003.

[2] D. Faure and C. N'edellec." ASIUM: Learning subcategorization frames and restrictions of selection." In 10th Conference on Machine Learning -- Workshop on Text Mining, Chemnitz, Germany, 1998.

[3] M.A. Hearst. "Automatic Acquisition of Hyponyms from Large Text Corpora." In proceedings of the14th international conference on computational linguistics, 1992.

[4] M.A. Hearst. "Automated discovery of wordnet relations. In Fellbaum,Wordnet" an Electronic Lexical database, MIT Press, 1998.

[5] S.A Caraballo. "Automatic construction of a hypenym-labeled noun hierarchi from text". In: Procedings of the 37th Annual Meeting of the association for computational linguistics, 1999, pp. 120-126

[6] P.Cimiano, A. Hotho, S. Staab. "Comparing conceptual, Partitional and agglomerative Clustering for learning taxonomies from text", In: Proceedings of the European Conference on Artificial Intelligence (ECAI'04), IOS Press, 2004, pp. 435-439

[7] H.Cunningham, Y. Wilks and R. Gaizauskas. "GATE -- a General Architecture for Text Engineering." In Proceedings of the 16th Conference on Computational Linguistics (COLING96) ,1996.

[8] E. Charniak and M. Berland. "Finding parts in very large corpora." In Proceedings of the 37th Annual Meeting of the ACL, 1999.

[9] R. Girju, A. Badulescu, and D. Moldovan. "Learning Semantic Constraints for the Automatic Discovery of Part-Whole Relations." In the Proceedings of the Human Language Technology Conference *(HLT),* 2003.

[10] P. Cimiano, J. Volker, *Text2Onto,*" a FrameWork for Ontlogy Learning and Data-dirven Change Discovery," In Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB'05). 2005.

[11] M. Poesio, A. Almuharab. "Identifying Concept Attributes Using a Classifier." In: Proceeding of the ACL on Deep Lexical Aquistion, 2005, pp. 18-27

[12] A. Almuharab, M. Poesio. "Attibute-based and value-based Clustering: an Evaluation". In Proceeding of EMNLP, 2004

[13] A.Collins, R. Michalskl, "The Logic Of Plausible Reasoning A Core Theory", Cognitive Science, Vol. 13, 1989, pp. 1-49.

[14] webpage: http://inex.is.informatik.uni-duisburg.de:2004/

[15] F. Oroumchian, R.N. Oddy, "An Application of Plausible Reasoning to Information Retrieval", ACM's SIGIR 1996

[16] A. Jalali, F. Oroumchian, "An Evaluation of Document Clustering by means of Plausible Inferences." International Journal of Computational Intelligence, 2004.

[17] E. Darrudi, M. Rahgozar, F. Oroumchian, "Human Plausible Reasoning for Question Answering Systems." In proceeding of Advances in Intelligent Systems - Theory and Applications. Luxembourg, 2004.

[18] Cimiano, A. Hotho, S.Staab. "Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis". Journal of AI Research, Vol. 24, 2005, pp. 305-339.

[19] van Rijsbergen. "Information Retrieval." London:Buttersorths, 1979. Second Edition

[20] M.Sanderson, W. Bruce Croft, "Deriving Concept Hierarchies from Text", Research and Development in Information Retrieval, 1999, pp.206-213.

[21] P. Ryu, K. Choi, "Taxonomy Learning using Term Specificity and Similarity", In: Proceedings of the 2nd Workshop on Ontology Learning and Population, 2006, pp 41–48.

[22] M. Shamsfard, A. Abdollahzadeh Barforoush, "The State of the Art in Ontology Learning: A Framework for Comparison", The Knowledge Engineering Review,Volume 18 , Issue4,Decembe, 2003 r, pp.293 – 316.