Conceptual Summarization using Ontologies and Nearest Neighborhood Clustering

Elahe Gavagsaz, Mahmoud Naghibzadeh, Mehrdad Jalali
Department of Software Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran
egavagsaz@mshdiau.ac.ir, naghibzadeh@um.ac.ir, jalali@mshdiau.ac.ir

Abstract - **Conceptual summarization aims to provide a database which comprises an abstraction of the entire document content. To effectively provide conceptual summarization, we have presented an approach that is used for conceptual querying. The approach is based on utilizing an ontology for similarity measure between concepts and the nearest neighborhood clustering algorithm for concepts clustering. The results show an improvement in the runtime and tolerant as regards noise.**

Keywords-conceptual summarization; ontology; nearest neighborhood clustering

I. INTRODUCTION

We can use ontologies for the organization of concepts, structure and relations within a knowledge domain. Use of ontologies as tools for information access provides a foundation for enhanced, knowledge-based approaches to surveying, indexing and querying of document collections.
In this paper we address an approach to conceptual summarization based on instantiated ontology. The main goal is preparing a tool for conceptual summarization of documents when they are used in conceptual querying. We use an ontology that includes the set of concepts, for investigation of the concepts. Conceptual investigation of set of documents can be performed by extracting a set of essential concepts that are the local points of the documents.

Summarization is a process of transforming sets of similar low level objects into more abstract conceptual representations [6], and more specifically, a summary for a set of concepts in the form of a smaller set of concepts. For instance {program, conductor} as summary for {virus, chip, compiler, bus} are or {device} as summary for {printer, monitor, mouse}.

We introduce an approach to conceptual summarization in which ontology plays a key role as reference for the conceptualization. We use nearest neighborhood clustering for concepts clustering and use ontology for similarity measurement between concepts. The semantic grouping that results from the clustering process can then lead to a summary by, for instance, taking a least upper bound of each of the clusters.

The purpose of the ontology in the context and conceptual summarization is to define and relate concepts that may appear in a document collection which may then be used in the summary.

Sources for knowledge base ontologies may have various forms and a taxonomy can be complemented with, for instance, word and term lists as well as dictionaries for definition of vocabularies and for handling of morphology. The well-known resource WordNet [5] is among the more interesting and useful resources for general ontologies.
To establish a general ontology we need an atomic concepts A, semantic relations R, and the set of well-formed terms L that is defined the in following:

$$L = \{A\} \cup \{ x\,[r_1 : y_1, \ldots, r_n : y_n] \mid x \in A, r_i \in R, y_i \in L\}$$

and taxonomy T over the set of atomic concepts A, we can use an inclusion relation "$\leq$" over all well-formed terms of the language L [3],[4], [8].
Therefore, the general ontology $O = (L, \leq, R)$ encompasses a set of well-formed expressions L derived in the concept language from a set of atomic concepts A, an inclusion relation generalized from the taxonomy relation in T, and a set of semantic relations R.

To form an instantiated ontology, we assume a general ontology $O = (L, \leq, R)$ and a set of concepts C. The instantiated ontology $O_C = (L_C, \leq_C, R)$ is a restriction of O to cover only the concepts in C and corresponds to "upper expansion" L_C of C in O

$$L_C = C \cup \{x \mid y \in C, x \in L, y \leq x\}$$
$$\text{“}\leq_C\text{”} = \{(x, y) \mid x, y \in L_C, x \leq y\}$$

Conceptual querying concerns retrieval of concepts appearing in an instantiated ontology- thus at a conceptual level to investigate the concepts appearing, or the content of the documents holding these concepts. we will use the knowledge captured in instantiated ontologies for deriving conceptual summarizations, in principle, any collection of text such as a single document, as set of documents. Thus, given a set of concepts C, we want to move towards a smaller set of representative concepts covering C, that is, towards an appropriate summary that includes what's most characteristic about C. Obviously, this is solely dependent on the structure of the instantiated ontology in use, but since these normally are structured as hierarchies, the use of more general concepts as a cover of subsumed concepts in C will lead to a summary with fewer elements.

978-1-61284-353-7/11/$26.00 ©2011 IEEE

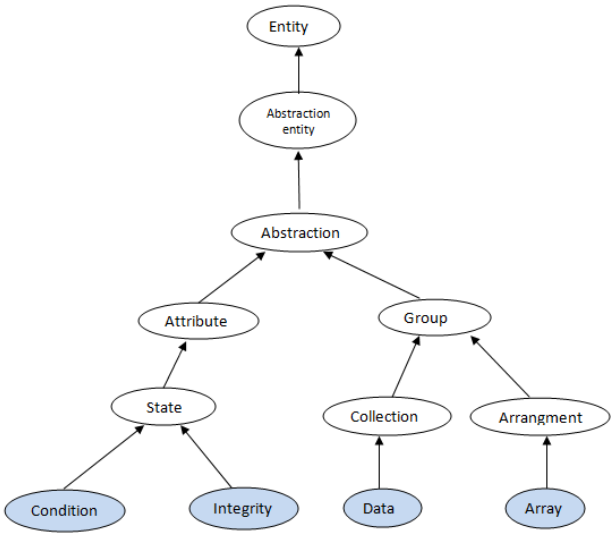

Figure 1: An instantiated ontology based on WordNet ontology and the set of instantiated concepts {condition, integrity, data, array}

Figure 1 shows an example of an instantiated ontology[7]. The general ontology is based on WordNet and the ontology shown is "instantiated" the following set of concepts:

$$C = \{\text{condition, integrity, array, data}\}$$

One approach to provide summaries is thus to divide the set of concepts into groups or clusters and to derive for each a representative concept-for instance the least upper bound (*lub*) for the group. We shall assume the presence of some kind of filtering or extraction mechanism that for a given text collection can produce the set of concepts appearing in the text.

The paper is organized as follows: Section 2 summarizes related work in the area where Subsection 2.1 introduce connectivity clustering approach and Subsection 2.2 talks about similarity clustering approach, Section 3 describes our proposed approach and Section 4 presents evaluation of algorithms results and conclude this paper in Section 5.

II. RELATED WORK

Andreasen and Bulskov [1] proposed two directions for deriving summaries: one based directly on connectivity in the ontology and the other drawing on statistical clustering applying similarity measures.

A. 2.1. Connectivity clustering

This approach is clustering based on connectivity in an instantiated ontology. Basic idea is to cluster a set of concepts based on their connections to common ancestors, for instance grouping two siblings lead to their common parent, and in addition to replace the group by the common ancestor. Thus with a bottom-up hierarchical clustering view, connectivity clustering is to move towards a smaller number of larger clusters and so smaller number of more general concepts.

In this approach, in each stage general concepts replaces some concepts without to assume priority for selection. If we want to form only some of the possible clusters, we should use the restrictions that are some kind of priority mechanism for selection of clusters. Among *deepness*, *redundancy, support* are important properties that might use to priority.

B. 2.2. Similarity clustering

A way to replace reasoning with simple computation is applying measures of similarity derived from the ontology. One obvious way to measure similarity in ontologies is to evaluate the distance between the concepts being compared, where a shorter distance implies higher similarity and vice versa [2].

In similarity clustering is based on similarity measurement over the set of concepts. The similarity measurement is based on ontology for clustering of concepts. With a given path length dependent similarity measurement derived from the ontology a *lub* centered, agglomerative, hierarchical clustering can be performed. Initially each "cluster" corresponds to an individual element of the set to be summarized. At each particular stage the two clusters which are most similar are joined together. This is the principle of conventional hierarchical clustering. However rather than replacing the two joined clusters with their union as in the conventional approach they are replaced by their *lub*.

However to take into account also the importance of clusters in terms of their sizes, the summary can be modified by the support of the generalizing concepts, *support*(x, C), that for a given concept specifies the fraction of elements from the set C covered:

$$support(x, C) = \frac{|\{y | y \in C, y \leqslant x\}|}{|C|}$$

This approach to summarization will not be very tolerant as regards noise in the clusters given. To get around this problem a soft definition of *lub* is introduced which is and are combined with crisp clusters to get more specific cluster-based summaries.

A soft definition of *lub* for a (sub)set of concepts $C\square$ should comprise "upper boundness" as well as "leastness" (or "least upperness ") expressing respectively the portion of concepts in $C\square$ that are generalized and the degree to which a concept is least upper with regard to one or more of the concepts in $C\square$. "Upper boundness" can be expressed for a set of concepts $C\square$ by $\mu_{ub(C\square)}$ simply as the support as regards $C\square$:

$$\mu_{ub(C\square)}(x) = support(x, C\square)$$

covering all generalizations of one or more concepts in C□ and including all concepts that generalizes all of C□ (including the top most concept *Top*) as full members.

"Leastness" can be defined on top of a function that expresses how close a concept is to a set of concepts C□ such a $dist(C$□, $y) = min\ x \in C$□$dist\ (x, y)$, where $dist(x, y)$ expresses the shortest path upwards from x to y, as follows:

$$\mu_{lu(C,\lambda)}(x) = \begin{cases} 1 & \text{when } \lambda = 0 \vee x = Top \\ 1 - \dfrac{dist(C', x)}{dist(C', Top) + \dfrac{1}{\lambda} - 1} & \text{otherwise} \end{cases}$$

where 0□ □□1 is a leastness parameter with □ = 1 corresponding to the most restrictive version of "leastness" and with the other extreme □=0 corresponding to no restriction at all (all upper concepts be comes full members). A simple soft least upper bound *flub* can now be defined as the product between μ_{lu} and μ_{ub}

$$\mu_{flub(C',\lambda)}(x) = \mu_{lu(C',\lambda)}(x) * \mu_{ub(C')}(x)$$

Notice that a *lub* for C is not necessary a best candidate among the *flub* elements. Thus, again with a division (crisp clustering) of C into $\{C_1, \ldots, C_k\}$, the basis for the summary here is the set of fuzzy sets $\{flub(C_1), \ldots, flub(C_k)\}$ leading to the summary

$$\delta(\{C_1, \ldots, C_k\}) = \left(\bigcup_{i=1}^{k} flub(C_i)\right)$$

As in the *lub*-based case the summarizers should in addition be weighted by support, thus we can weight all elements in each $\{flub(C_1), \ldots, flub(C_k)\}$ with the support of $flub\ (C_i)$ in C:

$$\delta(\{C_1, \ldots, C_k\}) = \left(\bigcup_{i=1}^{k} flub(C_i)\right) \otimes \left(\sum_{x \in flub(C_i)} \frac{|C_i|}{|C|} \Big/ x\right)$$

III. PROPOSE APPROACH

Concerning the concept clustering is division of set of data in subsets or separate clusters. This classification of data in clusters is based on a set of special properties and according to it grouping of data will be formed.

There are different methods for clustering which one of them is hierarchical clustering that is used in similarity clustering approach. Another clustering method which is more important to us is nearest neighborhood clustering method. This method is in the way that for clustering a series of data, each data will be studied separately. In this clustering method for classification of data we have to assume some specified value which is known as radius. The procedure of algorithm in this approach is as following:

1. Set i= 1 and k= 1. Assign pattern x_1 to cluster C_1.
2. Set i= i+1. Find nearest neighbor of x_i among the patterns already assigned to clusters. Let d_m denote the distance from x_i to its nearest neighbor. Suppose the nearest neighbor is in cluster m.
3. If d_m greater than or equal to t then assign x_i to C_m where t is the threshold specified by the user. Otherwise, set k= k+1 and assign x_i to a new cluster C_k.
4. If every pattern has been considered then stop else go to step 2.

As we see in above mentioned algorithm, the first datum is the center of the first cluster. The next datum will be compared with the center of cluster and the distance between them will be computed afterward. If the distance between this datum with center of first cluster was less than radius, this datum will be added to it, otherwise a new cluster must be made and this datum must be put in the center of that cluster. Concerning this, each datum will be compared with center of clusters. If this datum was placed in radius of one of center of clusters, that datum will be added to it. If it isn□t so, one will be added to the number of clusters. This procedure must be followed for all data collections. After study of all members of the collection, algorithm will be ended.

In our proposed approach, for classification of the concepts, we use the nearest neighborhood clustering approach. The reason of using this approach is because in similarity clustering approach, we compute similarity between each two of them in each stage. The two concepts which are more similar, will be selected and will be canceled from the set of concepts and their *lub* will be replaced. This procedure must be followed to reach to a concept. It seems that computing similarity is somehow more in this approach. The negative impact will be in the running time algorithm, when the number of concepts is high. Suppose n is the number of concepts in the set, the number of similarity calculation for this collection will be as follows:

$$number\ of\ similarity\ calculation = \binom{n}{2} + \binom{n-1}{2} + \cdots + \binom{2}{2}$$

Due to this relationship, algorithm running time will be $O(n^3)$. On the other hand, when an outlier is placed between a collection of concepts and as in computing *lub,* definitions is improved, still outlier has its negative effect to obtain general concepts. It also in some cases lead up to get a general concepts with higher weight. This subject somehow keeps us away from main goal (main goal is to reach to suitable concepts to provide suitable summary from all the documents).

We use the nearest neighbor clustering method for the classification of set of concepts to achieve two goals: (1) to

reduce the execution time of algorithm. (2) noise not have much impact in the final result.

The first concept will be placed as the center of first cluster. We calculated the similarity of the second concept with the first cluster center. Way to measure the similarity in the proposed approach is shared node method [2]. As has been said before in this clustering method we need to specify a number, we called the radius. (0.6 or 0.7 for this project is a good number because it represents, given a 60 or 70 percent similarity between the two concepts.) If the calculated similarity between the second concept and the first cluster center is more than the radius, this concept is added to the first cluster. If it is not so, a new cluster is formed and the second concept is placed in the new cluster center. Each of the concepts should be studied. For every concept, the similarity is calculated with all the center of clusters. We consider maximum similarity among the calculations performed. If it is greater than or equal to the radius, to be added into cluster, otherwise we increase the number of clusters and build a new cluster and the concept put as the new cluster center. The process to review all concepts would continue and thus concepts are classified.

The first advantage of using this clustering method is, reducing the number similarity calculation between concepts. As we see, in the similarity clustering approach similarity calculation is done between both concepts in each step. But in this approach to calculate the similarity for each concept only center of clusters are used. If n is the number of concepts and k is the number of clusters, we calculate the similarity for concepts that will be center of clusters (1+ 2+ …+ (k-1)) times. For each of the remaining concepts, (n-k), k similarity calculation is done, in practice. Finally, the total number of similarity calculations will be (1+ 2+ ...+ (k-1) + (n- k) k). In the worst case that all the concepts are noise and the amount of similarity between them is less than the desired radius, the number of clusters is n and the number of similarity calculations is $O(n^2)$. While the number of similarity calculations in the similarity clustering algorithm was $O(n^3)$.

After classification of concepts, for each cluster, based on instantiated ontology, *lub* is calculated. Based on the definitions, to compute the weight in the similarity clustering algorithm, the weight of each lub is computed. Thus the importance of each of them to stay on the database of conceptual querying is characterized.

The problem which has been studied before is the problem of outliers. Where noises are concerned some concepts are not closed to the concepts in set of the result of summarization, they have negative effect impact (negative effect means to reach to the result of summarization toward the most general concepts). Here another reason of using the nearest neighborhood clustering method is determined. Using this method in the proposed approach has the effect of placing noises in a single unit or in less unit clusters. At the time of computing the weight, they will not get considerable weight and it will not be among the selected concept for the collection of final summarization. It is another objective in using nearest neighborhood clustering method that we follow.

To prove this claim we must implement both the similarity clustering algorithm and the proposed algorithm. Instantiated ontology prepared in a specific domain and according to it experiments are done and using appropriate parameters, the two algorithms are compared.

Assume execution of the algorithm on a set of concepts, the number of retrieved relevant concepts is P, the total number of retrieved concepts is N and the total number of existing relevant concepts, which should have been retrieved, is M. Precision and recall are defined as follows:

$$precision = \frac{P}{N} \ , recall = \frac{P}{M}$$

Another parameter that is used is accuracy, which combines precision and recall and will have the following:

$$accuracy = 1 - \frac{(N-P)+(M-P)}{M} = recall(2 - \frac{1}{precision})$$

By these three parameters a good comparison can be done between these two algorithms.

IV. EVALUATION

For comparison of the proposed algorithm and similarity clustering algorithm, we have implemented both algorithms. Set of concepts that are assumed, are fifty specialized computer times. Using WordNet and Protégé
software on the set of concepts, we have prepared an instantiated ontology. Experiments conducted on the subset of the set of concepts.

The first issue in the test results is the issue of measuring the number of similarity calculations for each collection. Similarity measurement for the 10 experiments performed, has been calculated and is shown in Table 1. As you can see in all cases, the difference in the number of similarity calculation of similarity clustering algorithm and proposed algorithm is very impressive. Another comparing the execution time of both algorithms has been seen in Figure 2.

As previously described in addition to running time of our algorithm, we used three other parameters to compare the two algorithms. The first parameter is the precision. As you see in the Figure 3, except in experiment 4 and 10, in other cases, the proposed algorithm has better results. This means that, in most cases, the number of retrieved relevant concepts by the proposed algorithm to all the concepts that have been retrieved is more.

In Figure 4, comparison of the two algorithms using recall parameter is displayed.

TABLE 1: Number of similarity calculated

Number of concepts	*Similarity clustering*	*Proposed algorithm*
20	1140	99
23	1771	133
25	2300	164
26	2600	154
29	3654	194
29	3654	235
33	5456	297
34	5984	285
37	7770	366
50	19600	630

In all experiments, except the second one, the results of proposed algorithm are better. This subject means that in most of the cases, the number of retrieved and relevant concepts in compare to the number of the concepts which must be retrieved is more in proposed algorithm.

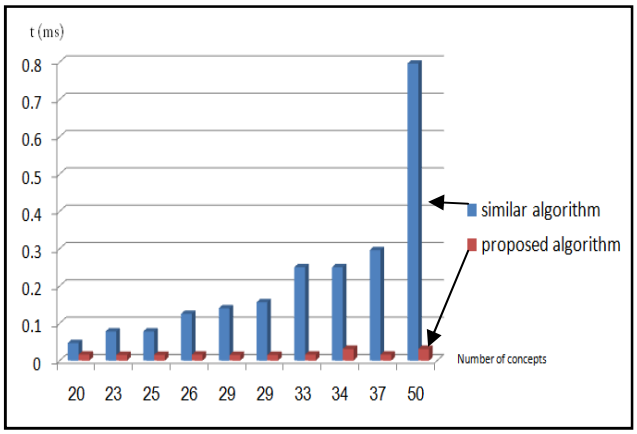

Figure 2: Compare run time of algorithms

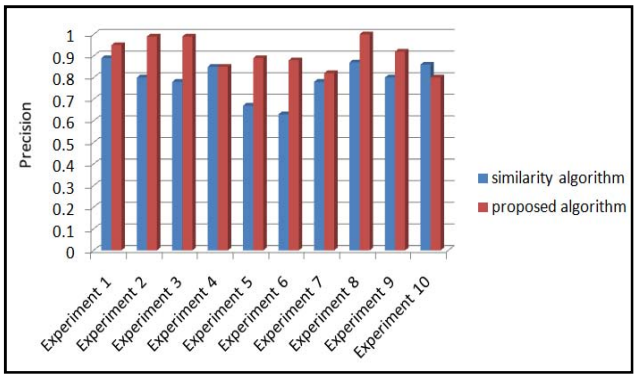

Figure 3: Compare precision in ten experiments

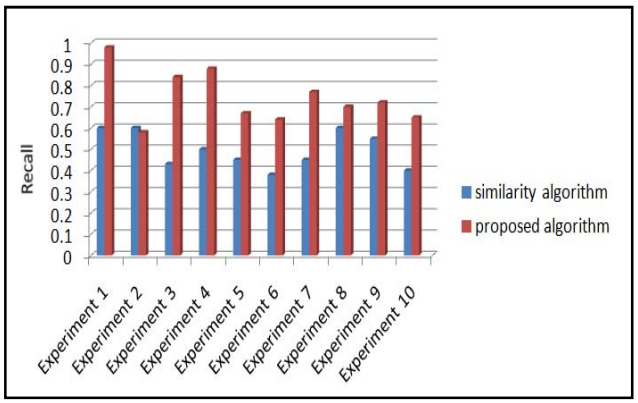

Figure 4: Compare recall in ten experiments

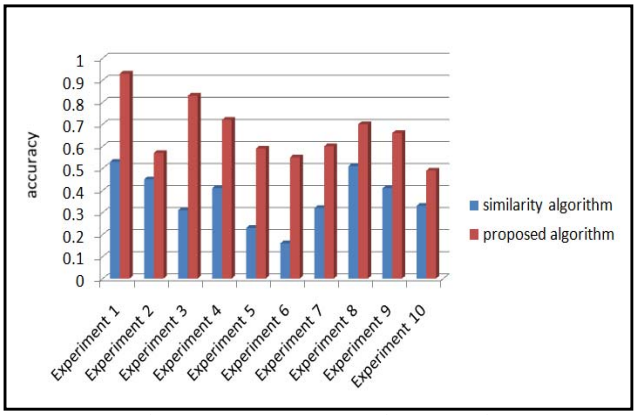

Figure 5: Compare accuracy in ten experiments

At the end, accuracy for both algorithms we are calculated. Comparison can be seen in Fig. 5. In all cases, the proposed algorithm accuracy is better than the similarity clustering algorithm.

V. CONCLUSION

In this paper we presented an approach for conceptual summarization that used nearest neighborhood clustering based on instantiated ontology. The main goal is preparing a tool for conceptual summarization of documents when they are used in conceptual querying.

We have reviewed both methods presented previously. Connectivity clustering is clustering based solely on connectivity in the instantiated ontology. The basic idea is to cluster a given set of concepts based on their connections to common ancestors, for instance grouping two siblings due to their common parent, and in addition to replace the group by the common ancestor. The second method is the similarity clustering. This method uses similarity measuring between concepts for their classification. Similarity measurement is based on ontology and through communication between the concepts that are based on hierarchical clustering methods.

The method presented here also uses the classification of concepts. Using the extracted common ancestor between

concepts that have been in a cluster, summarization operation is performed. We use nearest neighborhood clustering for classification of concepts. This clustering method is also based on similarity calculation and instantiated ontology. The first advantage of using it, is reducing the similarity calculation between concepts and the algorithm execution time. The second advantage, the impact of outlier on the final result is less.

The results of experiment show that the proposed algorithm running time is less than similarity clustering algorithm and the accuracy of the proposed algorithm is higher than the similarity clustering algorithm.

At the end it is recommended that the proposed algorithm to be tested with larger and more diverse datasets and is compared with other approaches.

REFERENCES

[1] T. Andreasen, H. Bulskov, "Conceptual querying through ontologies", *Fuzzy Sets and Systems*, Elsevier North- Holland, 2009, pp. 2159- 2172.

[2] T. Andreasen, R. Knappe, H. Bulskov, "Domain- specific similarity and retrieval", *11th International Fuzzy Systems Association World Congress IFSA 2005*, University Press, 2005, pp. 496- 502.

[3] J. F. Nilsson, "A logico- algebraic framework for ontologies- ONTOLOG", Proc. First Internat. Onto- Query Workshop, 2001.

[4] R. Knappe, H. Bulskov, T. Andreasen, "Perspectives on ontology- based querying", *International Jour-nal of Intelligent Systems*, John Wiley & Sons, 2007, pp. 739- 761.

[5] G. A. Miller, "Wordnet: a lexical database for english", *Communications of the ACM*, ACM, 1995, pp. 39- 41.

[6] R. R. Yager, E. E. Petry, "multicriteria approach to data summarization using concept hierarchies", *IEEE Transactions on Fuzzy Systems*, 2006.

[7] H. Bulskov, R. Knappe, T. Andreasen, "On measu- ring similarity for conceptual querying", *In Proc. of the 5th International Conference on Flexible Query Answering Systems*, Springer, 2002, pp. 100- 111.

[8] P. A. Jensen, J. F. Nilsson, "Ontology- Based Semantics for Prepositions", *Text, speech and language technology*, Syntax and Semantics of Prepositions, Springer, 2006, pp. 229- 244.