Investigating Similarity of Ontology Instances and its Causes

Anton Andrejko and Mária Bieliková

Institute of Informatics and Software Engineering
Faculty of Informatics and Information technologies
Slovak University of Technology, Ilkovičova 3, 842 16 Bratislava
{andrejko,bielik}@fiit.stuba.sk

Abstract. In this paper we present a novel method of comparing instances of ontological concepts in regard to personalized presentation and/or navigation in large information spaces. It is based on the assumption that comparing attributes of documents which were found interesting for a user can be a source for discovering information about user's interests. We consider applications for the Semantic Web where documents or their parts are represented by ontological concepts. We employ ontology structure and different similarity metrics for data type and object type attributes. From personalization point of view we impute reasons that might have caused user's interest in the content. Moreover, we propose a way to enumerate similarity for the particular user while taking into account individual user's interests and preferences.

1 Introduction

Applications providing information from large information spaces can provide a user more relevant content if personalization is used. Personalization of visible aspects is usually based on user characteristics represented in the user model. To provide proper personalization the user model needs to be reasonably populated with user characteristics that are up to date and relevant to the information space being accessed. Several approaches are used to obtain user characteristics. Some information can be acquired when the user is asked explicitly or from observing one's behavior while working with the application. Mining user characteristics from activity logs can be helpful to establishing patterns of needs or interests.

Analyzing content that is presented to a user is a good source of information about the user [1]. If we know user's rating given to displayed content (e.g. user's interest) we can acquire some characteristics by analyzing the content. Since the rating varies we need to understand possible reasons for why it is low or high. For instance, one can stumble upon hundreds of job offers on the Web that advertise a position for Java programmers requiring high school education, at least three years of previous experience, knowing basics of Web technologies, offering motivating salary, etc. Let us have two such offers that have most features similar and differ only in the job location. Assume we get different ratings for

these two offers. The range or variety of the evaluation rating that was derived could have been caused by the job location attribute.

In this paper we present a novel method for comparing instances of ontological concepts aimed at identification of common and different aspects to be used for personalization purposes. Examples used in the paper are from job offers domain that is the subject of a research project NAZOU[1] [2].

2 Related Work

Semantic Web applications typically use ontology methodologies as a base for metadata representation and reasoning. Several approaches to comparison of ontology concepts, or their instances, were mainly developed for the purpose of ontology management. Similarly this problem is also known in ontology mapping, matching or alignment. Their aim is to increase reusability and interoperability between different ontologies covering the same domain. In [3] an approach is described that is aimed at identification of changes in ontology versions on the level of ontology schema and ontology instances using various heuristics.

The approach described in [4] uses three independent similarity assessments. It deals with synonyms to ensure that synonyms refer to the same objects. Semantics are then incorporated and lastly semantic relations (e.g. *is-a*) are used to determine whether connected entities are related to the same set of entity classes. Finally, distance between two concepts is measured by the shortest path.

In [5] an approach is described that conceptualizes ontology mapping in four stages that include similarity of labels, instances, structures and previous mapping results verified by the application. While comparing instances the Edit-Distance method is used in conjunction with a Glue approach based on machine learning techniques [6]. It uses predefined similarity function to compute a similarity value for each pair of concepts and generates the similarity matrix.

A method that accomplishes comparing instances of tourism ontology concepts in two phases is described in [7]. The first phase is focused on preprocessing the concepts. Two graphs are built – the *inheritance graph* organizes ontological concepts according to a generalization hierarchy and the *similarity graph* in which nodes relate to concepts and edges have assigned similarity degree. Similarity is enumerated in the second phase using a three step process. First, structural attributes are used, then hierarchical structure is exploited, and finally a similarity measure is computed as a result of combination of two previous steps.

Comparison with ideal instance related to the particular domain (here job offers) is used in searching based on user's criteria [8]. The method allows searching also offers that do not fulfill criteria fully. The user is allowed to specify for each criterion, whether it has to be fulfilled, its importance, and precision.

A common characteristic for all the mentioned approaches is that they do not investigate causes of similarity. Automated similarity enumeration mimics to human similarity measure if different strategies are used according to clusters

[1] NAZOU – Tools for acquisition, organization and maintenance of knowledge in an environment of heterogeneous information resources, http://nazou.fiit.stuba.sk

of users [9]. Users gave reasons for their assessments which become the basis for machine learning algorithm that assigns users to a cluster. We use an automated approach to quantify and define reasons of similarity, what also contributes to scrutiny of the user model.

3 Similarity Enumeration

Similarity of two objects is expressed as a number from interval $\langle 0, 1\rangle$ where similarity of entirely different objects equals zero and similarity of identical objects equals one. Similarity characteristics are also characterized in *reflexivity* (where an object is identical to itself) and *symmetry* (where if object X equals Y, then Y reciprocally equals X).

For similarity enumeration any aggregation function can be used. We use mean value to enumerate similarity between instances of concepts. The similarity of instances *InstA* and *InstB* is evaluated as follows:

$$sim\,(InstA, InstB) = \frac{\sum_{i=0}^{|A\cap B|} GeneralSM_i\,(SetA, SetB)}{|A \cup B|} \tag{1}$$

where $GeneralSM_i$ encapsulates all similarity measures that are available (e.g. according to attribute type), A and B are sets of attributes instances consist of, respectively. Since an attribute can appear as a multiple, *SetA* and *SetB* are used as a possible set of objects that can be connected to the particular attribute.

When using aggregation of partial similarities the computed result is the same at all the times no matter what is the context. Since each user has different preferences related to similarity, we consider this in the similarity enumeration. It is useful, especially in cases when a user model that holds user's preferences is available. Therefore, we introduce weights to personalize enumeration what allows computing similarity taking into account user's individuality. Now, similarity is evaluated as follows:

$$sim\,(InstA, InstB) = \frac{\sum_{i=0}^{|A\cap B|} weight_i \times GeneralSM_i\,(SetA, SetB)}{\sum weight} \tag{2}$$

where the assigned meaning of variables is the same as in Eq. 1. The variable *weight* is computed for each attribute that two instances have in common. It gets a value from range $\langle 1, w\rangle$ according to the match with corresponding characteristic in the user model. We assume that user's likes should result in more influence on total similarity in our similarity assessment model. If there is a corresponding characteristic in the user model to an attribute of the instance and also the value of the characteristic equals the value of the attribute, the *weight* is set to w. In cases where no match between values is detected, *weight* is selected from the range $(1,w)$ according to the computed closeness to preferred value in the user model, e.g. a city belongs to the same region as the city preferred by the user in the user model but it is not that specific city. Our experiments showed that $weight = 2.0$ is a worthy selection value (see Sect. 5).

4 Method for Ontology Instances Similarity Evaluation

In the Semantic Web applications documents or their parts are represented by ontological concepts. A concept describes a set of real objects [10]. Concepts can be ordered in a hierarchy. Instances of concepts reflect objects from real world. An example of an instance representing a job offer is depicted in Fig. 1.

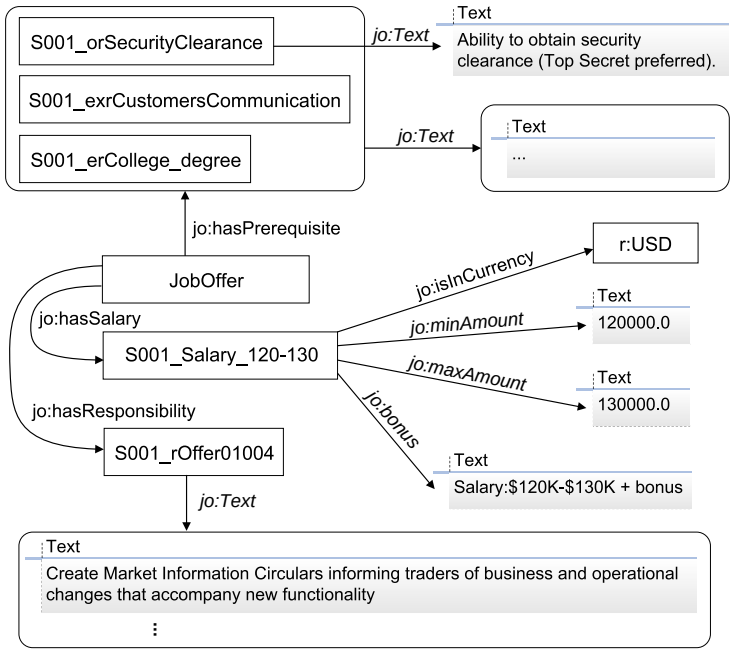

Fig. 1. Example of an instance representing a part of job offer. Each object has its unified identifier, here we present only object's label. *JobOffer* is an identifier of the instance. We use italic font for data type attributes to distinguish them from object type attributes. For simplicity, multiple attributes are surrounded by a rounded box.

If we think about an ontology statement as a triple in form *subject – predicate – object*, an attribute represents predicate. In general, there are *data type* and *object type* attributes. A data type attribute is connected to a literal value that can be of several types defined according XML Schema. An object type attribute expresses the relationship of a concept to another concept, or to an instance.

4.1 Recursive Evaluation of Ontology Instances Similarity

To evaluate similarity we have proposed a method based on recursive evaluation of the attributes and component objects an instance consists of. The main idea is based on looking for common pairs in both attributes and their sequential processing. Basic steps of the method are depicted in Fig. 2.

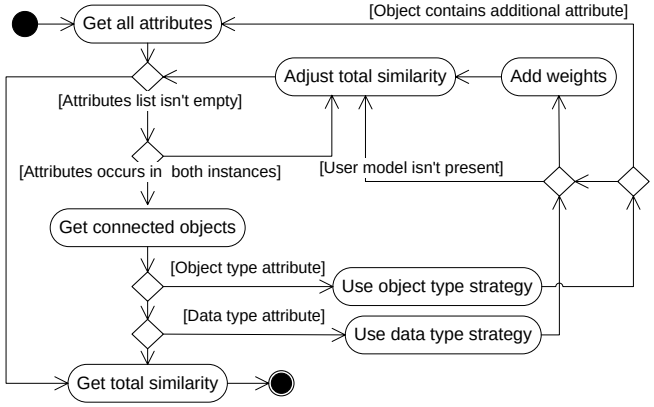

Fig. 2. Basic steps of the method for recursive traversing of instance.

The process of comparison begins with acquiring all the attributes from both instances. An attribute can have single or multiple occurrences in both instances or single/multiple occurrence in one instance only. When the attribute has a single occurrence in both instances, objects (literals) refered to, are evaluated for their similarity. Variety of similarity metrics can be used. If the attribute is data type, the comparison for the attribute terminates after a metrics is used to evaluate similarity between connected literals. Resulting computed similarity measure(s) is aggregated to a total similarity measure. In the case of an object type attribute, a metrics for connected object is used. Furthermore, the comparison is being launched recursively on that object until literals are achieved.

A multiple occurrence is the most specific case we have to cope with. We move solution of this problem to the lower level. Anytime a multiple attribute is acquired only its one occurrence in the instance is considered. Afterwards, all objects (literals) connected to that attribute are acquired from both instances. Instead of dealing with attributes we now have to deal with two sets of objects (or literals) possibly with different cardinalities. Here, a problem of how to figure out which object from the first set should be compared with an object from another set with the contribution to the total similarity emerges (see Sect. 4.2).

In the situation, when single or multiple occurrence of an attribute is present in only one instance we use an assumption that instances are entirely different in the attribute if there is no presence of that attribute in both instances. In regard to similarity definition, the similarity equals zero if two objects have nothing in common. In this case we estimate similarity for such an occurrence of the attribute as equal zero.

4.2 Comparison Metrics and Similarity Measure

We proposed two groups of metrics according to an attribute's type: data type and object type. To evaluate similarity between literals connected to a data

type attribute, any string based metrics can be used[2]. To achieve better results, the semantic type of the literal content is taken into account (e.g. string, date, number are each treated differently). When evaluating the similarity of objects connected to the object type attribute their other characteristics can be considered (e.g. number of attributes and their types, position in taxonomy tree) [11].

Taxonomy distance is a heuristic similarity measure for evaluating similarity between objects connected to the object type attribute. The edge-counting method computes the shortest path between nodes. Distance is defined as the shortest path linked through a common ancestor or as the general shortest path [9]. Since we do not need a result what is closer or further, but a float number between 0 and 1, we proposed our taxonomy distance metrics. It assumes that the more nodes have two objects in common in the taxonomy tree the more they are similar. Similarity is computed as the number of common nodes in the taxonomy divided by number of nodes in the longer path leading to the object (see Fig. 3).

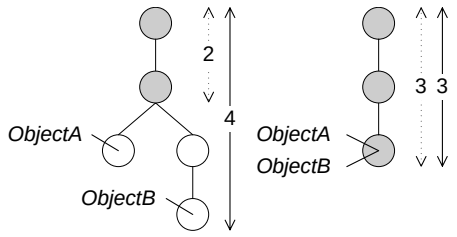

Fig. 3. Taxonomy distance for objects *ObjectA* and *ObjectB* is computed. Common part (nodes) in the taxonomy is emphasized by dotted arrow; solid arrow is used to show longer distance from the root node. For left example $sim(ObjectA, ObjectB) = 2/4 = 0.5$, for right example $sim(ObjectA, ObjectB) = 3/3 = 1.0$.

Identification of relevant pairs using only the object's label is not satisfactory. Each object in the ontology can have a label that could be compared using selected data type metrics. Since the label is optional and does not have to necessarily express any semantics we avoid using it. It should be noted that for automatically acquired instances it is obvious that meaningful labels are not present. We proposed the similarity measure to identify pairs of objects, therefore, a relevance matrix is constructed which size is specified by cardinalities of sets of objects.

The matrix holds similarities for each pair of objects from the sets. In the case of literals, data metrics are used. For objects the recursive algorithm is employed as for the entire instance. Afterwards, an identification of pairs can start. Number of pairs is given by the set with the lower cardinality. Finding pairs

[2] A collection of methods suitable for string comparing is implemented in the open source library SimMetrics, http://www.dcs.shef.ac.uk/~sam/simmetrics.html

with very low similarity measure can be restricted by using a critical threshold as a filter. The algorithm for finding relevant pairs follows these steps:

```
WHILE count(pairs) < count(getSmallerSet(setA, setB)) DO
  SET maxValue to getMaxValue(matrix)
  STORE maxValue in List
  SET coordinates of maxValue to X and Y
  FOR each item in matrix
    IF item.row = X OR item.column = Y
      SET item to -1
    END IF
  END FOR
END WHILE
```

Leftover objects are handled in the same way as described above for attributes that have occurrence in one instance only. An example is shown in Fig. 4.

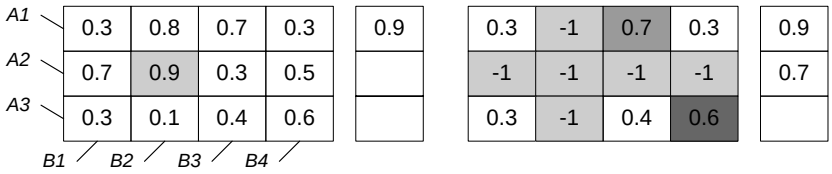

Fig. 4. Identifying relevant pairs from the sets. Similarities in the matrix are random numbers. In first iteration (left) at [A2,B2] is maximal value 0.9 and it is stored. Second row and second column are set to -1. In the next iteration at [A1,B3] is maximal value 0.7. The last coordinate is [A3,B4]. Object B1 is evaluated as a leftover.

Our experimental results show that the way we find related pairs (in case of object type attributes in combination with taxonomy distance) leads to meaningful results. First, identities were found (maximal possible value 1.0). Other found pairs were interpreted as semantically similar by a human. The number of multiple attributes in job offers is usually small (less than 10). Therefore, threshold 0.3 for deciding which pairs are still meaningful is reasonable.

4.3 Investigating Similarity Causes

Our goal is not only to compute the similarity between instances but also to investigate reasons that caused the similarity or difference to be used later for personalization purposes. From the user's evaluation given to content we can deduce user's likes or dislikes. We assume that if the instance includes an attribute that the user likes, it will likely influence his/her rating towards higher (or positive) values. On the other hand, attributes of the content that the user dislikes will influence rating towards lower (or negative) values.

Therefore, we introduced two threshold values that divide attributes into three sets according to their similarity values. Since we are interested in attributes that significantly influence the user's evaluation, we give up splitting outcome interval in the equal parts. An attribute exhibiting similarity greater than the *positive threshold* would be assigned to the positive interval set and the similarity exhibiting lower than the *negative threshold* to negative interval set.

Thresholds were specified experimentally for this job offer domain. We evaluated 55 000 attributes. Attributes with similarity equal 0.0 or 1.0 were not considered to eliminate identities and attributes with no occurrence in both instances. The rest of the attributes were ordered according to similarity measure and the Pareto principle (also known as 80/20 rule) was used. We split the 20% segment in half to select 10 % of highest and 10 % of lowest values. This way, the *positive threshold* was set to 0.65 and *negative threshold* to 0.25. Domain independence of thresholds is subject of further experiments.

Attributes classified by this method can be transformed into user characteristics and then used for filling or updating existing characteristics in the user model. A transformation of attributes to user characteristics as well as their updating in the user model is not included in the scope of this paper. The presented method only prepares inputs for further processing. Using positive and negative set of attributes in combination with user's feedback for characteristics update in the user model would improve user characteristics estimation.

5 Method Evaluation and Conclusions

We described a method for comparing instances of ontological concepts based on recursive traversing of instance's structure. Final similarity is a result of mean aggregation of similarities computed for particular attributes while their type is considered. Introducing similarity computed for individual attributes allows employing semantics from ontology representation. It allowed us to extend similarity enumeration with weights to compute similarity for particular user to be used for personalization purposes. Moreover, we investigate reasons (attributes) that influenced user's evaluation (e.g. interest) of the content. We introduced two thresholds dividing attributes in three sets. From personalization point of view we are interested in only two outer sets (positive and negative). These can be used by other tools for actualization of characteristics in the user model.

We have evaluated proposed method using developed software tool called *ConCom* (Concept Comparer) implemented in Java. Sesame framework was used to access the ontological models represented in OWL DL. Evaluation was processed on an experimental job offer ontology developed in the course of the research project NAZOU. In the experiment, similarity for 10 000 pairs was computed. The experiments showed that computed results fulfill all criterions requested for similarity. In Fig. 5 there is a depiction of a sample of 80 pairs where similarity was computed by *ConCom* for (1) all attributes and (2) for common attributes only. Computed values were sorted out according to the computed similarities in the first way.

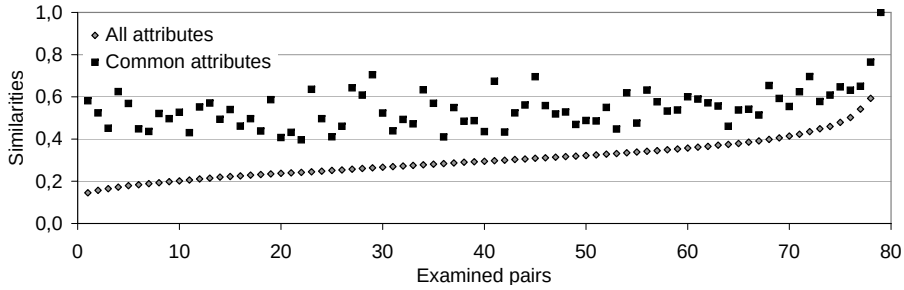

Fig. 5. Similarity computed by *ConCom* in regard to considered attributes.

In the following experiment a user was involved. A sample of 300 job offer pairs was used where 30 randomly selected pairs appeared twice as a check sample. We asked the user to assess similarity on a scale from 0 to 7. Afterwards, acquired values were recounted to similarity interval. The result for a randomly selected set of 40 pairs is depicted in the Fig. 6.

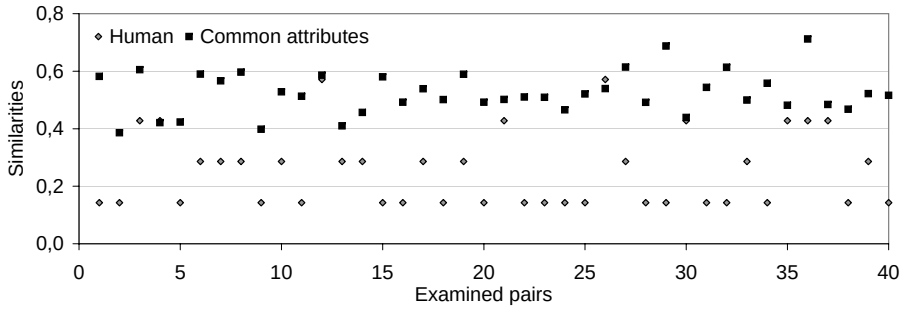

Fig. 6. Similarity estimated by a human and by *ConCom* for common attributes.

We used similarity computed for common attributes to compare with our test subject human evaluation since its values mimic values from evaluation given by a human better. This result could have been caused by the fact, that a human user can easier evaluate lower amount of attributes and especially common attributes. Therefore, for further experiments with the user model we use similarity computed for common attributes. On the other hand, using only common attributes in our experiments resulted in narrow range of similarity values – in 94.1 % computed similarities were from range 0.35 to 0.7, what makes it not very useful for discovering user's characteristics.

To figure weights for personalized similarity a user model was involved consisting of one characteristic only (*hasDutyLocation*). Job offers used in the experiment consisted of an average of sixteen attributes in averaged and contained that attribute with the same value as in the user model. Already doubled weights

cause noticeable change in the similarity – from 0.06 up to 0.10 depending on the number of attributes job offers consist of.

We have started exploring the interest of scientific publications to further investigate domain independence of the method. The achieved results can be useful in user model creation in combination with other methods [12, 13], as a support for clustering algorithms, semantic annotation or repository maintenance tools as well as for recommending similar content in recommending systems. The aim here is to improve semantic search using the method for personalized navigation within ontology instances that represent metadata of large information space.

This work was partially supported by the State programme of research and development "Establishing of Information Society" under the contract No. 1025/04 and by the Scientific Grant Agency of Slovak Republic, grant No. VG1/3102/06.

References

1. Brusilovsky, P., Tasso, C.: Preface to special issue on user modeling for web information retrieval. UMUAI **14**(2-3) (2004) 147–157
2. Návrat, P., Bieliková, M., Rozinajová, V.: Acquiring, organising and presenting information and knowledge from the web. In: CompSysTech'06. (2006)
3. Tury, M., Bieliková, M.: An approach to detection ontology changes. In: ICWE '06: Workshop Proc. of 6th Int. Conf. on Web Eng., Palo Alto, ACM Press (2006)
4. Rodríguez, M.A., Egenhofer, M.J.: Determining semantic similarity among entity classes from different ontologies. IEEE Transactions on Knowledge and Data Engineering **15**(2) (2003) 442–456
5. Liu, X., Wang, Y., Wang, J.: Towards a semi-automatic ontology mapping. In: Proc. of 5th Mexican Int. Conf. on Artificial Intelligence (MICAI'06), IEEE (2006)
6. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to map between ontologies on the semantic web. In: WWW '02: Proceedings of the 11th international conference on World Wide Web, New York, NY, USA, ACM (2002) 662–673
7. Formica, A., Missikoff, M.: Concept similarity in symontos: An enterprise management tool. The Computer Journal **45**(6) (2002) 583–595
8. Pázman, R.: Ontology search with user preferences. In: Tools for Acquisition, Organisation and Presenting of Information and Knowledge. (2006) 139–147
9. Bernstein, A., Kaufmann, E., Burki, C., Klein, M.: How similar is it? towards personalized similarity measures in ontologies. In: 7th International Conference Wirtschaftsinformatik (WI-2005), Bamberg, Germany (2005) 1347–1366
10. Ding, L., Kolari, P., Ding, Z., Avancha, S.: Using ontologies in the semantic web: A survey. In Sharman, R., et al., eds.: Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems, Springer (2007) 79–113
11. Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. Journal of Artificial Intelligence Research **11** (1999) 95–130
12. Andrejko, A., Barla, M., Bieliková, M.: Ontology-based user modeling for web-based inf. systems. In: Advances in Information Systems Development New Methods and Practice for the Networked Society. Volume 2. Springer (2007) 457–468
13. Barla, M., Bieliková, M.: Estimation of user characteristics using rule-based analysis of user logs. In: Data Mining for User Modeling Proceedings of Workshop held at the International Conference on User Modeling UM2007. (2007) 5–14