An Ontology Mapping Algorithm between Heterogeneous Product Classification Taxonomies

Wooju Kim[1], Sangun Park[2], Siri Bang[3], and Sunghwan Lee[4]

[1,3,4] Department of Information and Industrial Systems Engineering, College of Engineering, Yonsei University, 134, Shin-Chon Dong, Seoul, South Korea
{wkim@yonsei.ac.kr,feeljazz@nate.com,siri27@nate.com}
[2] The u-City Research Institute, Yonsei University, 134, Shin-Chon Dong, Seoul, South Korea {sangun.park@gmail.com}

Abstract. Research on ontology merging and mapping is one of the most important issues in the Semantic Web because ontologies are developed and used by various sites and organizations respectively. Electronic commerce is the area that require ontology mapping on product comparison over different product classification taxonomies of various shopping malls. But, a strict mapping strategy may lead a customer's configuration to search failure. Therefore we suggest a mapping algorithm for product matching that can provide more products by increasing sensitivity with reasonable decrease of specificity. We performed a comparative evaluation between our algorithm and PROMPT with 6 experimental sets.

Keywords: Semantic Web, Ontology Mapping, e-Commerce, Information Retrieval

1 Introduction

Research on ontology merging and mapping is one of the most important issues in the Semantic Web environment because ontologies are developed and used by various sites and organizations respectively. In electronic commerce area, each shopping mall has its own vocabulary and product hierarchy that cause a semantic interoperability problem [8]. Gathering and merging product information from tremendous shopping malls in most product comparison sites depends on manual work by human. But, it is extremely inefficient to manage promptly changing information about products. That is, electronics commerce is the domain which essentially needs automatic ontology mapping on product names and attributes for efficient product search over multiple shopping malls.

Most research on ontology mapping [1][3] focuses on precision because incorrect matching among different ontologies can cause severe problems. PROMPT [5] is one of the approaches that adopt such conservative strategies with exact matching. But, product search in comparison shopping requires more flexible mapping between user's configuration and products. According to the Boston Consulting Group [7], 48% of all users have experienced unsatisfactory search results on desired products and 28% of all product purchase tryouts could not reach purchase because of search

failure. A strict mapping strategy that may involve search failure is not desirable because customers want rich information on products. Therefore, our research objective is to increase the number of matched products with the customer's configuration in automatic product mapping compared to the other ontology mapping approaches. This can be achieved by increasing recall rate with reasonable decrease in precision.

2 Sensitivity and Precision

Precision can be calculated by dividing the number of correctly matched terms by the number of all matched terms [2]. Therefore, if one wants to enhance precision, the best way is to minimize incorrectly matched terms. That is the reason that most approaches of ontology mapping adopt conservative and strict strategies. Meanwhile, **sensitivity** divides the number of correctly matched terms by the number of terms that should be matched [2]. Strict matching strategies try to increase precision as much as possible in spite of low sensitivity. But, those strict strategies are not desirable in comparison shopping as we mentioned in Section 1. **Specificity** is used with sensitivity together for classification performance measures and calculated by dividing the number of correctly not matched terms by the number of terms that should not be matched [2]. If we try to increase sensitivity by matching more products, specificity can be worse because correctly non-matched terms will decrease. Therefore, we use sensitivity and specificity in the performance evaluation and comparison of our algorithm and PROMPT.

Then, how to increase sensitivity compared to exact matching? The easiest way is using synonyms from WordNet [4]. By matching all synonyms of the given product, we can match more products and increase the chance of matching more correct products. But, it can also decrease precision. So, using synonym alone is not recommendable. In WordNet, a word has different senses and each sense has its own synonyms. If we can choose an appropriate sense of the given product from WordNet, it is possible to prevent precision from dropping too much by narrowing the synonym range. In this paper, we propose an ontology mapping algorithm for product matching based on above idea.

3 Product Matching through Ontology Mapping

3.1 Word Sense Disambiguation for Product Categories

Selection of an appropriate sense for a given product is important in order to keep precision at a reasonable level. If we use synonyms of all senses of the product, it will decrease precision because incorrect matching can increase. But, word sense disambiguation can enhance precision. The basic idea of word sense disambiguation is comparing a product hierarchy and hypernym hierarchies of senses of the product in WordNet. The sense *notebook* that is a computer has a different hypernym hierarchy with that of a book for notes as shown in Fig. 1. By comparing the product hierarchy

of ODP (Open Directory Project) [6] in the left column of Fig. 1 and hypernym hierarchies in WordNet of the right column, we can choose a proper sense for *notebook*.

The first step of disambiguation is searching for hypernyms from a hierarchy of a sense that match with upper categories of the product as shown in the formula (1). *CS()* returns a set of hypernyms that match to a given upper category *x* from a given sense hierarchy *p*.

$$cs(x, p) = \{h \mid h \in SYNSETS(x) \text{ and } h \in hypernyms(p)\}$$
$$\text{where } x \text{ is an upper category of the product hierarchy} \tag{1}$$

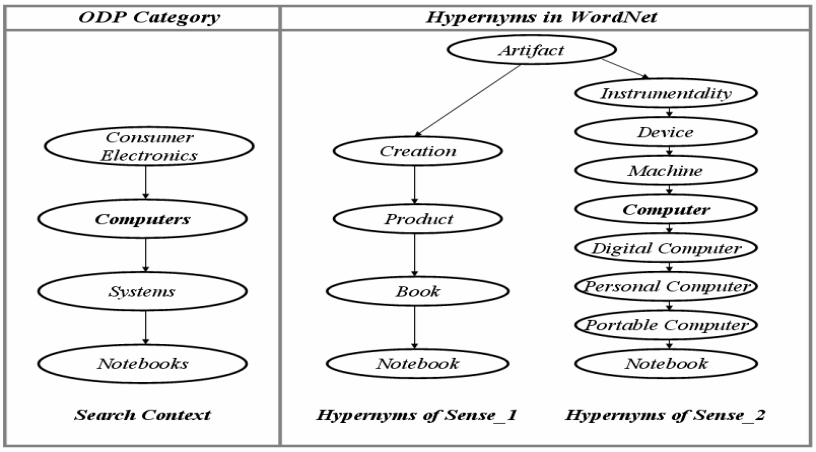

Fig. 1. A Product Hierarchy of ODP and Corresponding Hypernym Hierarchies in WordNet

The next step is calculating a measure represents the similarity between an upper category and a sense. If a matching hypernym is close to the sense, then the similarity is high because a closer hypernym is more important. The function *hypernymproximity()* returns the similarity by calculating a minimum distance between the matching hypernym and the *base* node of the sense in the hypernym hierarchy as shown in (2).

$$hypernymproximity(x, p) = \begin{cases} \dfrac{1}{Min\_dist(cs(x, p), base)} & if\ cs(x, p) \neq \phi \\ 0 & otherwise \end{cases} \tag{2}$$

The last step is calculating similarity between a product and senses. The function *pathproximity()* adds all *hyperproximity* of a given sense and divides it by the number of nodes of the product hierarchy as shown in (3).

$$pathproximity(p) = \frac{\sum_{x \in upper\_categories(base)} hypernymproximity(x, p)}{n} \tag{3}$$

3.2 Generation of Candidates for the Best Matching Category Path

Once we found an exact sense for the product from WordNet, the next step is to search for the candidates for the best matching category path from a target ontology. After the completion of search, we need to delete redundant categories of the product. To do this, the algorithm generates serial hierarchies of the categories by extracting all upper categories.

3.3 Choice of the Best Matching Product Category

To choose the best matching product category, we designed two measures for the calculation of similarities between the given product hierarchy and candidates. One is *co-occurrence* and the other is *order-consistency*. The measure, *co-occurrence* is the ratio of the number of common categories between a source hierarchy and a target hierarchy to the number of categories of the target hierarchy. However, *co-occurrence* is not enough to represent similarity because *co-occurrence* cannot measure orders of categories in the hierarchy. The other measure, *order-consistency* compares this order of categories. The final similarity between a source product and a target product is the average of *co-occurrence* and *order-consistency*. We choose a threshold on the similarity to determine whether we match the source product with the target product or not. We expect that the matching result will be changed by controlling not only the ratio of *co-occurrence* and *order-consistency* to the similarity but also the threshold.

4 Empirical Evaluation and Results

In this section, we compare the mapping results between our algorithm and PROMPT. PROMPT compares two different taxonomies and automatically recommends the matching terms by using synonyms [5].

To conduct an experiment, we selected two well-known shopping malls – Amazon.com and Buy.com – and ODP [6]. We constructed product ontologies from Amazon.com, Buy.com, and ODP respectively for our experiment. The product ontology of Amazon.com consists of 136 nodes, Buy.com consists of 225 nodes, and ODP consists of 133 nodes. A set of the experiment consists of one source ontology and one target ontology. Therefore, there are 6 sets in the experiment.

Table 1. Performance Results on Sensitivity and Specificity

Experimental Set	**Sensitivity**		**Specificity**	
	Our Algorithm	PROMPT	Our Algorithm	PROMPT
Amazon → Buy	96.9%	61.7%	56.4%	91.1%
Amazon → ODP	93.3%	25.7%	78.9%	84.5%
Buy → Amazon	93.5%	56.0%	61.0%	94.8%
Buy → ODP	97.2%	40.6%	69.5%	89.6%
ODP → Amazon	92.9%	36.0%	50.5%	88.1%
ODP → Buy	85.7%	60.9%	70.5%	84.7%
Average	**93.3%**	**46.8%**	**64.5%**	**88.8%**

Table 1 shows the performance results on sensitivity and specificity. On average, sensitivity of our algorithm is better than PROMPT by 46.5% and worse by 24.3%. It shows that our objective is successfully achieved. The maximum and minimum differences of sensitivity are 67.6% and 24.8% respectively while the maximum and minimum differences of specificity are -37.6% and -5.6% respectively.

5 Conclusion

In this paper, we proposed an ontology mapping algorithm that provides efficient product matching between heterogeneous product classifications. And we performed a comparative evaluation between our algorithm and PROMPT with 6 experimental sets. The experiment results showed that our algorithm is more effective than PROMPT in product comparison of the electronic commerce domain.

There is an interesting future research issue. Sensitivity and specificity can be changed by controlling not only the ratio of *co-occurrence* and *order-consistency* to the similarity but also the threshold as we described in Section 3. We expect that we can find the optimal values of the parameters – the ratio and the threshold. We are planning to conduct experiments finding the optimal values.

Acknowledgments. This research was funded mainly by the Ministry of Information and Communication in Republic of KOREA - National Project (Project management of Institute for Information Technology Advancement).

References

1. Ehrig, M., and Y. Sure: Ontology Mapping - An Integrated Approach. Lecture Notes in Computer Science, No. 3053 (2004) 76-91.
2. Han, J. and M. Kamber: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers (2000) 325-326.
3. Kalfoglou, Y. and M. Schorelmmer: Ontology mapping: the state of the art. The Knowledge Engineering Review, 18(1) (2003) 1-32.
4. Miller, G. A.: WordNet a Lexical Database for English. Communications of the ACM, 38(11) (1995) 39-41.
5. Noy, N.F., and M.A. Musen: The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping. International Journal of Human-Computer Studies, 59(1) (2003) 983-1024.
6. Open Directory Project. http://www.dmoz.com (2006).
7. Pecaut, D., M. Silverstein, and P. Stanger: Winning the Online Consumer Insights into Online Consumer Behavior. A Report by the Boston Consulting Group, <http://www.bcg.com> (2000).
8. Veltman, K.H.: Syntactic and Semantic Interoperability: New Approaches to Knowledge and the Semantic Web. New Review of Information Networking, 7 (2001) 159-184.