DODDLE-OWL: On-the-fly Ontology Construction with Ontology Quality Management

Takeshi Morita[1], **Yoshihiro Shigeta**[1], **Naoki Sugiura**[1], **Naoki Fukuta**[1],
Noriaki Izumi[2], **and Takahira Yamaguchi**[3]
[1]Shizuoka University, 3-5-1 Johoku, Hamamatsu, Shizuoka 432-8011, Japan
[2]National Institute of AIST, 2-41-6, Aomi, Koto-ku, Tokyo, Japan
[3]Keio University, 4-1-1 Hiyoshi, Kohoku-ku, Yokohama-shi, Kanagawa, Japan
morita@ks.cs.inf.shizuoka.ac.jp, yamaguti@ae.keio.ac.jp

1 Introduction

In this paper, we propose a software environment for user-centered on-the-fly ontology construction named DODDLE-OWL (Domain Ontology rapiD DeveLopment Environment - Web Ontology Language extension). The architecture of DODDLE-OWL is re-designed based on DODDLE-II [1], our former study. DODDLE-OWL has the following five modules: Input Module, Construction Module, Refinement Module, Visualization Module, and Translation Module. DODDLE-OWL supports the construction of both taxonomic relationships and non-taxonomic relationships in ontologies. Since DODDLE-II has been built for ontology construction not for the Semantic Web but for typical knowledge systems, it needs some extensions for the Semantic Web such as OWL export facility. DODDLE-OWL contributes the evolution of ontology construction and the Semantic Web.

2 The DODDLE-OWL Architecture

Figure 1 shows the overview of DODDLE-OWL. First, as input of DODDLE-OWL, several domain specific terms are selected by a user in the Input Module. DODDLE-OWL shows a list of noun concepts in the document as candidates of input concept. At this phase, a user also identifies the sense of terms to map those terms to concepts in WordNet. In the Construction Module, DODDLE-OWL generates the basis of the ontology, an initial concept hierarchy and set of concept pairs, by referring to WordNet and documents. The detail of the Construction Module is described in section 2.1. In the Refinement Module, the initial ontology produced by the Construction Module is refined by the user through interactive support by DODDLE-OWL. The detail of the Refinement Module is described in section 2.2. The ontology constructed by DODDLE-OWL can be exported with the representation of OWL. Finally, the Visualization Module (MR^3 [2]) is connected with DODDLE-OWL and works with an RDF graphical editor.

2.1 Construction Module

In the Construction Module, DODDLE-OWL generates the basis of output ontology for further modification by a user. This module consists of two sub-modules: Hierarchy Construction Module and Relationship Construction Module. For building taxonomic relationship of an ontology, DODDLE-OWL attempts to extract "best-matched concepts". That is, "concept matching" between input concepts and WordNet concepts is done, and matched nodes are extracted, and then merged at each root nodes. To extract related concept pairs from domain specific text corpus as a basis of identifying non-taxonomic relationships, statistic methods are applied. In particular, WordSpace and an association rule algorithm are used in this part and these methods attempt to identify significantly related concept pairs.

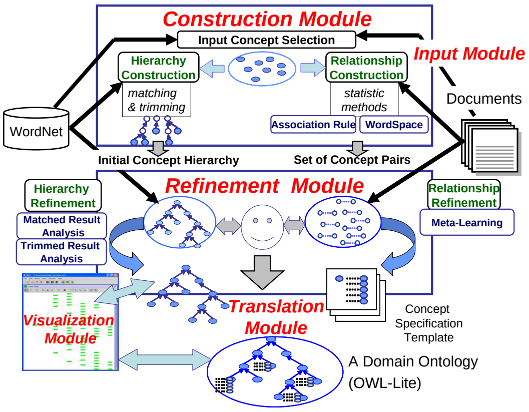

Figure 1: DODDLE-OWL overview

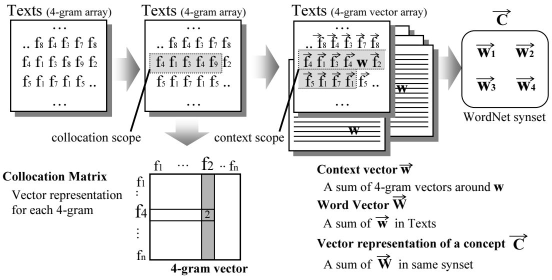

Figure 2: Construction flow of WordSpace

Construction of WordSpace

WordSpace is constructed as shown in Figure 2.

1. Extraction of high-frequency 4-grams

Since letter-by-letter co-occurrence information becomes too much and so often irrelevant, we take term-by-term co-occurrence information in four words (4-gram) as the primitive to make up co-occurrence matrix useful to represent context of a text based on experimented results. We take high frequency 4-grams in order to make up WordSpace.

2. Construction of collocation matrix

A *collocation matrix* is constructed in order to compare the context of two 4-grams. Element $a_{i,j}$ in this matrix is the number of 4-gram f_i which comes up just before 4-gram f_j (called *collocation area*). The collocation matrix counts how many other 4-grams appear before the target 4-gram. Each column of this matrix is the *4-gram vector* of the 4-gram f.

3. Construction of context vectors

A *context vector* represents context of a word or phrase in a text. A sum of 4-gram vectors around appearance place of a word or phrase (called *context area*) is a context vector of a word or phrase in the place.

4. Construction of word vectors

A word vector is a sum of context vectors at all appearance places of a word or phrase within texts, and can be expressed with Eq.1. Here, $\tau(w)$ is a vector representation of a word or phrase w, $C(w)$ is appearance places of a word or phrase w in a text, and $\varphi(f)$ is a 4-gram vector of a 4-gram f. A set of vector $\tau(w)$ is WordSpace.

$$\tau(w) = \sum_{i \in C(w)} \Big(\sum_{f \text{ close to } i} \varphi(f) \Big) \tag{1}$$

5. Construction of vector representations of all concepts

The best matched "synset" of each input terms in WordNet is already specified, and a sum of the word vector contained in these synsets is set to the vector representation of a concept corresponding to an input term. The concept label is the input term.

6. Construction of a set of similar concept pairs

Vector representations of all concepts are obtained by constructing WordSpace. Similarity between concepts is obtained from inner products in all the combination of these vectors. Then we define certain threshold for this similarity. A concept pair with similarity beyond the threshold is extracted as a similar concept pair.

Finding Association Rules between Input Terms

The basic association rule algorithm is provided with a set of transactions, $T := \{t_i \mid i = 1..n\}$,where each transaction t_i consists of a set of items, $t_i = \{a_{i,j} \mid j = 1..m_i, a_{i,j} \in C\}$ and each item $a_{i,j}$ is a set of concepts C. The algorithm finds association rules $X_k \Rightarrow Y_k : (X_k, Y_k \subset C, X_k \cap Y_k = \{\})$ such that measures for support and confidence exceed user-defined thresholds. Thereby, support of a rule $X_k \Rightarrow Y_k$ is the percentage of transactions that contain $X_k \cup Y_k$ as a subset (Eq.2) and confidence for the rule is defined as the percentage of transactions that Y_k is seen when X_k appears in a transaction (Eq.3).

$$support(X_k \Rightarrow Y_k) = \frac{\mid \{t_i \mid X_k \cup Y_k \subseteq t_i\} \mid}{n} \tag{2}$$

$$confidence(X_k \Rightarrow Y_k) = \frac{\mid \{t_i \mid X_k \cup Y_k \subseteq t_i\} \mid}{\mid \{t_i \mid X_k \subseteq t_i\} \mid} \tag{3}$$

As we regard input terms as items and sentences in text corpus as transactions, DODDLE-OWL finds associations between terms in text corpus. Based on experimented results, we define the threshold of support as 0.4% and the threshold of confidence as 80%. When an association rule between terms exceeds both thresholds, the pair of terms are extracted as candidates for non-taxonomic relationships.

2.2 Refinement Module

In order to improve the quality of the initial ontology, the Refinement Module works interactively with a user. Since the initial taxonomy is constructed from a general ontology, we need to adjust the taxonomy to the specific domain considering an issue called Concept Drift. It means that the position of particular concepts changes depending on the domain. For concept drift management, DODDLE-OWL applies two strategies: Matched Result Analysis and Trimmed Result Analysis. In Matched Result Analysis, DODDLE-OWL divides the taxonomy into PABs (PAths including only Best matched concepts) and STMs (SubTrees that includes best-matched concepts and other concepts and so can be Moved) and indicates on the screen. PABs are paths that include only best-matched concepts that have senses suitable for the given domain. STMs are subtrees of which root is an internal concept of WordNet and its subordinates are all best-matched concepts. Since the sense of an internal concept has not been identified by a user yet, STMs may be moved to other places for the concept adjustment to the domain. In addition, for Trimmed Result Analysis, DODDLE-OWL counts the number of internal concepts when the part was trimmed. By considering this number as the original distance between those two concepts, DODDLE-OWL indicates to move the lower concept to other places. As a facility for related concept pair discovery, there are functions that allow users to attempt some ways to improve the quality of extracted concept pairs through trial and error by changing parameters of statistic methods. Users can re-adjust the parameters of WordSpace and association rule algorithm and check the result. After that, DODDLE-OWL generates "Concept Specification Templates" by using the results. It consists of some concept pairs which have considerable relationship found from the result value of statistic methods. By referring to the constructed domain specific taxonomic relationship and the "Concept Specification Templates", a user constructs a domain ontology.

3 Future Work

In order to evaluate how DODDLE-OWL is doing in a practical field, case studies have been done in particular field of law and business [1]. A future work is to apply meta-learning scheme to the Relationship Refinement Module of DODDLE-OWL.

References

[1] Naoki Sugiura, Masaki Kurematsu, Naoki Fukuta, Noriaki Izumi, and Takahira Yamaguchi. *A Domain Ontology Engineering Tool with General Ontologies and Text Corpus*. Proceedings of the 2nd Workshop on Evaluation of Ontology based Tools, pp.71–82, 2003.

[2] Takeshi Morita, Noriaki Izumi, Naoki Fukuta and Takahira Yamaguchi. *MR^3: Meta-Model Management based on RDFs Revision Reflection*. Proceedings of the 6th Joint Conference on Knowledge-Based Software Engineering (JCKBSE 2004), pp.228–236, 2004.