A Domain Ontology Development Environment Using a MRD and Text Corpus

Naomi Nakaya\textsuperscript{1} and Masaki Kurematsu\textsuperscript{2} and Takahira Yamaguchi\textsuperscript{1}

\textsuperscript{1} Faculty of Information, Shizuoka University
3-5-1 Johoku Hamamatsu Shizuoka 432-8011 Japan
\{nakaya, yamaguti\}@ks.cs.inf.shizuoka.ac.jp

\textsuperscript{2} Faculty of Software and Information Science, Iwate Prefectural University
152-52 Takizawasugo Takizawa Iwate 020-0193 Japan
kure@soft.iwate-pu.ac.jp

Abstract. In this paper, we describe how to exploit a machine-readable dictionary (MRD) and domain-specific text corpus in supporting the construction of domain ontologies that specify taxonomic and non-taxonomic relationships among given domain concepts. A) In building taxonomic relationships (hierarchically structure) of domain concepts, some hierarchically structure can be extracted from a MRD with marked sub-trees that may be modified by a domain expert, using matching result analysis and trimmed result analysis. B) In building non-taxonomic relationships (specification templates) of domain concepts, we construct concept specification templates that come from pairs of concepts extracted from text corpus, using WordSpace and an association rule algorithm. A domain expert modifies taxonomic and non-taxonomic relationships later. Through the case study with the Contracts for the International Sale of Goods (CISG), we make sure that our system can work to support the process of constructing domain ontologies with a MRD and text corpus.

1 Introduction

Although ontologies have been very popular in many application areas, we still face the problem of high cost associated with building them manually. In particular, since domain ontologies have the meaning specific to application domains, human experts have to make huge efforts for constructing them entirely by hand.

In order to reduce the costs, automatic or semi-automatic methods have been proposed using knowledge engineering techniques and natural language processing ones (cf. Ontosaurus [1]). The authors have also developed a domain ontology rapid development environment called DODDLE [2], using a machine-readable dictionary. However, these environments facilitate the construction of only a hierarchically structured set of domain concepts, in other words, taxonomic conceptual relationships.

As domain ontologies have been applied to widespread areas, such as knowledge sharing, knowledge reuse and software agents, we need software environments that support a human expert in constructing domain ontologies with not only taxonomic conceptual relationships but also non-taxonomic ones. In order to develop environments, it seems better to put together two or more techniques such as knowledge engineering, natural language processing, machine learning and data engineering (e.g. [3], [4]).
Here in this paper, we extend DOODLE into DOODLE-II that constructs both taxonomic and non-taxonomic conceptual relationships, exploiting WordNet [5] and domain-specific text corpus (text corpus) with the automatic analysis of lexical co-occurrence statistics and an association rule algorithm[6]. Furthermore, we evaluate how DOODLE-II works in the field of law, the Contracts for the International Sale of Goods(CISG).

constructing domain ontologies.

2 DOODLE-II : A Domain Ontology Rapid Development Environment

Figure 1 shows an overview of DOODLE-II , “a Domain Ontology rapid Development Environment” that has the following two components: Taxonomic relationship acquisition module using WordNet and Non-taxonomic relationship learning module using text corpus.

A domain expert who is a user of DOODLE-II gives a set of domain terms to the system. A) The taxonomic relationship acquisition module (TRA module) does “spell match” between input domain terms and WordNet. The “spell match” links these terms to WordNet. Thus the initial model from the “spell match” results is a hierarchically structured set of all the nodes on the path from these terms to the root of WordNet. However the initial model has unnecessary internal terms (nodes). They do not contribute to keeping topological relationships among matched nodes, such as parent-child relationship and sibling relationship. So we get a trimmed model by trimming the unnecessary internal nodes from the initial model. In order to refine the trimmed model, we have the following two strategies that we will describe later: Matched result analysis and Trimmed result analysis.

B) The non-taxonomic relationship learning module (NTRL module) extracts the pairs of terms that should be related by some relationship from text corpus, analyzing lexical co-occurrence statistics, based on WordSpace (WS) [7] and an associate rule algorithm(AR). Thus the pairs of terms extracted from text corpus are the candidates for non-taxonomic relationships. NTRL module also extracts candidates for taxonomic relationships from these pairs, analyzing the distance between terms in a document.

We can built concept specification templates by putting together taxonomic and non-taxonomic relationships for the input domain terms. The relationships should be identified in the interaction with a human expert.

3 Taxonomic Relationship Acquisition

After getting the trimmed model, TRA module refines it by interaction with a domain expert, using Matched result analysis and Trimmed result analysis.

First, TRA module divides the trimmed model into a PAB (a Path including only Best spell-matched nodes) and a STM (a Sub-Tree that includes best spell-matched nodes and other nodes and so can be Moved) based on the distribution of best-matched nodes. A PAB is a path that includes only best-matched nodes that have the senses good for given domain specificity. Because all nodes have already been adjusted to the domain in PABs, PABs can stay in the trimmed model. A STM is such a sub-tree that an internal node is a root and the subordinates are only best-matched nodes. Because internal nodes have not been confirmed to have the senses good for a given domain, a STM can be moved in the trimmed model.

In order to refine the trimmed model, DOODLE-II can use trimmed result analysis. Taking some sibling nodes with the same parent node, there may be many differences about the number of trimmed nodes between them and the parent node. When such a big difference comes up on a sub-tree in the trimmed model, it is be better to change the structure of it.
DODDLE-II asks a human expert if the sub-tree should be reconstructed or not. Based on the empirical analysis, the sub-trees with two or more differences may be reconstructed.

Finally DODDLE-II completes taxonomic relationships of the input domain terms manually from the user.

4 Non-Taxonomic Relationship Learning

NTRL module almost comes from WS. WS derives lexical co-occurrence information from a large text corpus and is a multi-dimension vector space (a set of vectors). The inner product between two word vectors works as the measure of their semantic relatedness. When two words’ inner product is beyond some upper bound, there are possibilities to have some non-taxonomic relationships between them.

NTRL module also use an AR algorithm to find associations between terms in text corpus. When an AR between terms exceeds user-defined thresholds, there are possibilities to have some non-taxonomic relationships between them.

4.1 Construction of WordSpace

WS 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 WS.
2. construction of context vectors  A context vector represents context of a word or phrase in a text. Element $a_{i,j}$ in a context vector $\overrightarrow{w}_i$ is the number of 4-gram $g_j$ which comes up around appearance place of a word or phrase $w_i$ (called context scope). The concept vector counts how many other 4-grams come up in context scope of a word or phrase. The size of context scope is decided by users, depending on the length of sentences included in documents.

3. construction of word vectors  A word vector $\overrightarrow{W}_i$ is a sum of context vectors $\overrightarrow{w}_i$ at all appearance places of a word or phrase $w_i$, within texts. A set of word vectors is WS.

4. construction of vector representations of all concepts  The best matched “synset” of each input term in WordNet is already specified, and a sum of the word vector contained in these synsets is concept vector set to the vector representation of a concept corresponding to a input term. The concept label is the input term. A concept vector $\overrightarrow{C}$ can be expressed with the following formula. Here, $A(w)$ is all appearance places of a word or phrase $w$ in a text, and $\overrightarrow{w}(i)$ is a context vector at a appearance place $i$ of a word or phrase $w$.

$$\overrightarrow{C} = \sum_{w \in \text{synset}(C)} \left( \sum_{i \in A(w)} \overrightarrow{w}(i) \right)$$

5. construction of a set of similar concept pairs  Vector representations of all concepts are obtained by constructing WS. 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.

4.2 Finding Association Rules between Input Terms

The basic AR 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, $t_i = \{a_{i,j} \mid j = 1..m, a_{i,j} \in CS\}$ and each item $a_{i,j}$ is from a set of concepts $CS$. The algorithm finds ARs $X_k \Rightarrow Y_k : \{X_k, Y_k \subset CS, X_k \cap Y_k = \{\}\}$ 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 and confidence for the rule is defined as the percentage of transactions that $Y_k$ is seen when $X_k$ appears in a transaction.

$$\text{support}(X_k \Rightarrow Y_k) = \frac{\left| \{t_i \mid X_k \cup Y_k \subseteq t_i\} \right|}{n}$$

$$\text{confidence}(X_k \Rightarrow Y_k) = \frac{\left| \{t_i \mid X_k \cup Y_k \subseteq t_i\} \right|}{\left| \{t_i \mid X_k \subseteq t_i\} \right|}$$
As we regard input terms as items and sentences in text corpus as transactions, DODDLE-II 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 rules between terms exceeds thresholds, the pair of terms are extracted as candidates for non-taxonomic relationships.

4.3 Constructing and Modifying Concept Specification Templates

A set of similar concept pairs from WS and term pairs from the AR algorithm becomes concept specification templates. Both of the concept pairs, whose meaning is similar (with taxonomic relation), and has something relevant to each other (with non-taxonomic relation), are extracted as concept pairs with above-mentioned methods. However, by using taxonomic information from TRA module with co-occurrence information, DODDLE-II distinguishes the concept pairs which are hierarchically close to each other from the other pairs as TAXONOMY.

A user constructs a domain ontology by considering the relation with each concept pair in the concept specification templates, and deleting unnecessary concept pairs.

4.4 Extracting Taxonomic Relationships from Text Corpus

NTRL module tries to extract pairs of terms which form part of a candidate for domain-specific hierarchy structure. Because we suppose that there are taxonomic relationships in text corpus. In order to do that, we pay attention to the distance between two terms in a document. In this paper, the distance between two terms means the number of words between them. If the distance between two terms is small and the similarity between them is close, we suppose that one term explains the other. If the distance is large and the similarity is close, we suppose that they have taxonomic relationships. According to above-mentioned idea, we calculate the proximally rate between two terms within a certain scope. It is the number of times both terms occur within the scope divided by the number of times only one term occurs within it. We define certain threshold for this proximally rate. Pairs of terms whose proximally rate is within this threshold and the similarity between them is beyond the threshold are extracted as part of a candidate for non-taxonomic relationships. DODDLE-II asks the domain expert if the hierarchy structure from TRA module should be changed into unified one or not.

5 Case Studies for Taxonomic Relationship Acquisition

In order to evaluate how DODDLE is doing in practical fields, case studies have been done in a particular field of law called Contracts for the International Sale of Goods (CISG)[8]. Two lawyers joined the case studies. The details of the case studies are described in [2].

Generally speaking, in constructing legal ontologies, 70 % or more support comes from DODDLE. About half portion of the final legal ontology results in the information extracted from WordNet. Because the two strategies just imply the part where concept drift may come up, the part generated by them has low component rates and about 30 % hit rates. So one out of three indications based on the two strategies work well in order to manage concept drift. Because the two strategies use such syntactical feature as matched and trimmed results, the hit rates are not so bad. In order to manage concept drift smartly, we may need to use more semantic information that is not easy to come up in advance in the strategies.
6 Case Studies for Non-Taxonomic Relationship Learning

DODDLE-II is being implemented on Perl/Tk now. Figure 3 shows the ontology editor.

Subsequently, as a case study for non-taxonomic relationship acquisition, we constructed the concept definition for significant 46 concepts of having used on the first case study (Table 1) with editing the concept specification template using DODDLE-II, and verified usefulness. The concept hierarchy, which the lawyer actually constructed using DODDLE in the first case study was used here (Figure 4).

6.1 Construction of WordSpace

High-frequency 4-grams were extracted from CISG (about 10,000 words) and 543 kinds of 4-grams were obtained. The extraction frequency of 4-grams must be adjusted according
to the scale of text corpus. As CISG is the comparatively small-scale text, the extraction frequency was set as 7 times this case. In order to construct a context vector, a sum of 4-gram around appearance place circumference of each of 46 concepts was calculated. One article of CISG consists of about 140 4-grams. In order to construct a context scope from some 4-grams, it consists of putting together 60 4-grams before the 4-gram and 10 4-grams after the 4-gram independently of length of a sentence. For each of 46 concepts, the sum of context vectors in all the appearance places of the concept in CISG was calculated, and the vector representations of the concepts were obtained. The set of these vectors is used as WS to extract concept pairs with context similarity.

Having calculated the similarity from the inner product for the 1035 concept pairs which is all the combination of 46 concepts, and having used threshold as 0.87, 77 concept pairs were extracted.

6.2 Finding Associations between Input Terms

In this case, DODDLE-II extracted 55 pairs of terms from text corpus using the above-mentioned AR algorithm. There are 15 pairs out of them in a set of similar concept pairs extracted using WS.

6.3 Constructing and Modifying Concept Specification Templates

Concept specification templates were constructed from two sets of concept pairs extracted by WS and AR. Table 2 is the list of the extracted similar concepts corresponding to each concept. In Table 2, a concept in bold letters is either an ancestor, descendant or a sibling to the left concept in the concept hierarchy constructed using DODDLE in the first case study.
### Table 2: The Concept Pairs Extracted According to Context Similarity (threshold 0.87)

<table>
<thead>
<tr>
<th>CONCEPT</th>
<th>CONCEPT LIST IN SIMILAR CONTEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>acceptance</td>
<td>communication, offer, indication, telex</td>
</tr>
<tr>
<td>act</td>
<td>offeror, assent, effect, payment, person, quantity, time, goods, delivery, dispatch, price, contract, delay, withdrawal, offeror, place, quality</td>
</tr>
<tr>
<td>assent</td>
<td>offeror, act, effect, offer, person, offerree, withdrawal, time, proposal</td>
</tr>
<tr>
<td>communication</td>
<td>acceptance, offer, telex, conduct, indication</td>
</tr>
<tr>
<td>conduct</td>
<td>party, telex, communication</td>
</tr>
<tr>
<td>contract</td>
<td>effect, act, person, delivery, payment, quantity</td>
</tr>
<tr>
<td>delay</td>
<td>delivery, offer, act, payment</td>
</tr>
<tr>
<td>delivery</td>
<td>payment, quantity, goods, place, act, delay, time, contract, person, effect, quality</td>
</tr>
<tr>
<td>dispatch</td>
<td>goods, price, act, person, quantity, offeror</td>
</tr>
<tr>
<td>effect</td>
<td>person, assent, act, offeror, contract, proposal, payment, time, withdrawal, party, delivery</td>
</tr>
<tr>
<td>goods</td>
<td>dispatch, quantity, delivery, payment, act, person, price, quality</td>
</tr>
<tr>
<td>indication</td>
<td>intention, acceptance, communication</td>
</tr>
<tr>
<td>intention</td>
<td>indication</td>
</tr>
<tr>
<td>offer</td>
<td>acceptance, assent, communication, delay</td>
</tr>
<tr>
<td>offerree</td>
<td>withdrawal, offeror, assent, act, price</td>
</tr>
<tr>
<td>offeror</td>
<td>act, assent, withdrawal, offeror, person, effect, time, price, dispatch</td>
</tr>
<tr>
<td>party</td>
<td>conduct, effect, place, person</td>
</tr>
<tr>
<td>payment</td>
<td>quantity, delivery, place, act, goods, quality, delay, effect, person, contract, time</td>
</tr>
<tr>
<td>person</td>
<td>effect, offeror, act, proposal, goods, assent, withdrawal, contract, dispatch, payment, delivery, party, place, price</td>
</tr>
<tr>
<td>place</td>
<td>payment, delivery, time, quantity, party, act, person</td>
</tr>
<tr>
<td>price</td>
<td>dispatch, act, offeror, goods, withdrawal, offerree, person</td>
</tr>
<tr>
<td>proposal</td>
<td>person, effect, withdrawal, offeror</td>
</tr>
<tr>
<td>quality</td>
<td>quantity, payment, goods, act, delivery</td>
</tr>
<tr>
<td>quantity</td>
<td>payment, delivery, goods, act, quality, dispatch, place, contract, time</td>
</tr>
<tr>
<td>telex</td>
<td>conduct, communication, acceptance</td>
</tr>
<tr>
<td>time</td>
<td>act, offeror, delivery, place, effect, payment, quantity, assent</td>
</tr>
<tr>
<td>withdrawal</td>
<td>offeror, offeror, person, price, act, assent, effect, proposal</td>
</tr>
</tbody>
</table>

**Figure 5:** The Concept Specification Templates for “goods”

**Figure 6:** The Concept Definition for “goods” with Editing the Templates

concept specification templates, such a concept is distinguished as TAXONOMY relation. As taxonomic and non-taxonomic relationships may be mixed in the list based on only context similarity, the concept pairs which may be concerned with non-taxonomic relationships are obtained by removing the concept pairs with taxonomic relationships.

Figure 5 shows concept specification templates extracted about the concept “goods”. The final concept definition is constructed from consideration of concept pairs in the templates. Figure 6 shows the definition of the concept “goods” constructed from the templates.

### 6.4 Extracting Taxonomic Relationships from Text Corpus

In this case, we defined the threshold for the proximally rate as 0.78 and the certain scope as the same sentence. DODDLE-II extracted 128 pairs of concepts regarded as having taxonomic relationships from text corpus. 8 pairs out of them have occurred in the concept hierarchy constructed by the user and have not occurred in the trimmed model. That is, they are the same as modifications by the user. It shows that DODDLE-II can extract taxonomic relationships, which are not included in a MRD, from text corpus. But the rate of accepted taxonomic relationships is about 6%(8/128) and is not good. So, we have to improve do that.
Table 3: Evaluation by the User with Legal Knowledge

<table>
<thead>
<tr>
<th></th>
<th># Extracted concept pairs</th>
<th># Accepted concept pairs</th>
<th># Rejected concept pairs</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS</td>
<td>77</td>
<td>18</td>
<td>59</td>
<td>0.23(18/77)</td>
<td>0.38(18/48)</td>
</tr>
<tr>
<td>AR</td>
<td>55</td>
<td>13</td>
<td>42</td>
<td>0.24(13/55)</td>
<td>0.27(13/48)</td>
</tr>
<tr>
<td>The join of WS and AR</td>
<td>117</td>
<td>27</td>
<td>90</td>
<td>0.23(27/117)</td>
<td>0.56(27/48)</td>
</tr>
</tbody>
</table>

Figure 7: Three Different Sets of Concept Pairs from User, WS and AR

6.5 Results and Evaluation

The user evaluated the following two sets of concept pairs: ones extracted by WS and ones extracted by AR. Figure 7 shows three different sets of concept pairs from user, WS and AR. Table 3 shows the details of evaluation by the user, computing precision and recall with the numbers of concept pairs extracted by WS and AR, accepted by the user and rejected by the user.

Looking at the field of Precision in Table 3, there is almost no differences among three kinds of results from WS, AR and the join of WS and AR. However, looking at the field of Recall in Table 3, the recall from the join WS and AR is higher than that from each WS and AR, and then goes over 0.5.

Generating non-taxonomic relationships of concepts is harder than modifying and deleting them. Therefore, taking the join of WS and AR with high recall, it supports the user in constructing non-taxonomic relationships.

7 Related Work

In the research using verb-oriented method, the relation of a verb and nouns modified with it is described, and the concept definition is constructed from these information (e.g. [10]). In [9], taxonomic relationships and Subcategorization Frame of verbs (SF) are extracted from technical texts using a machine learning method. The nouns in two or more kinds of different SF with a same frame-name and slot-name is gathered as one concept, base class. And ontology with only taxonomic relationships is built by carrying out clustering of the base class further. Moreover, in parallel, Restriction of Selection (RS) which is slot-value in SF is also replaced with the concept with which it is satisfied instantiated SF. However, proper evaluation is not yet done. Since SF represents the syntactic relationships between verb and noun, the step for the conversion to non-taxonomic relationships is necessary.

On the other hand, in ontology learning using data-mining method, discovering non-taxonomic relationships using a AR algorithm is proposed by [3] and [4]. They extract concept pairs based on the modification information between terms selected with parsing, and made the concept pairs a transaction. By using heuristics with shallow text processing, the generation of a transaction more reflects the syntax of texts. Moreover, RLA, which is their
original learning accuracy of non-taxonomic relationships using the existing taxonomic rela-
tions, is proposed. The concept pair extraction method in our paper does not need parsing, 
and it can also run off context similarity between the terms appeared apart each other in texts 
or not mediated by the same verb.

8 Conclusions

In this paper, we discussed how to construct a domain ontology using existing MRD and 
text corpus. In order to acquire taxonomic relationships, two strategies have been proposed: 
matched result analysis and trimmed result analysis. Furthermore, in order to learn non-
taxonomic relationships, concept pairs have been extracted from text corpus with WS and 
AR. Taking the join of WS and AR, the recall goes over 0.5 and so it works to support the 
user in constructing non-taxonomic relationships of concepts.

The future work comes as follows: the system integration of taxonomic relationship ac-
quision module and non-taxonomic relationship learning module and the application to large 
size domains.

Acknowledgments

We would like to express our thanks to Mr. Takamasa Iwade (a graduate student of Shizuoka 
university) and Mr. Takuya Miyabe (a student of Shizuoka university) and the members in the 
Yamaguchi-Lab.

References

[1] Bill Swartout, Ramesh Patil, Kevin Knight and Tom Russ: “Toward Distributed Use of Large-Scale Ontolo-
gies”, Proc. of the 10th Knowledge Acquisition Workshop (KAW’96), (1996)


(2000)

72–79 (2001)


Processing for Lexical Acquisition edited by Branimir Boguraev & James Pustejovsky, 77–96

Seirin-Shoin(1993)

[9] David Faure, Claire Nédellec, “Knowledge Acquisition of Predicate Argument Structures from Technical 

[10] Udo Hahn, Klemens Schnattinger : “Toward Text Knowledge Engineering”, AAAI98, IAAAI-98 proceed-
ings, 524-531 (1998)