Analysis of an Iterative Algorithm for Term-Based Ontology Alignment

Shisanu Tongchim, Canasai Kruengkrai, Virach Sornlertlamvanich,
Prapass Srichaivattana, and Hitoshi Isahara

Thai Computational Linguistics Laboratory,
National Institute of Information and Communications Technology,
112,Paholyothin Road, Klong 1, Klong Luang, Pathumthani 12120, Thailand
{shisanu, canasai, virach, prapass}@tcllab.org, isahara@nict.go.jp

Abstract. This paper analyzes the results of automatic concept alignment between two ontologies. We use an iterative algorithm to perform concept alignment. The algorithm uses the similarity of shared terms in order to find the most appropriate target concept for a particular source concept. The results show that the proposed algorithm not only finds the relation between the target concepts and the source concepts, but the algorithm also shows some flaws in the ontologies. These results can be used to improve the correctness of the ontologies.

1 Introduction

To date, several linguistic ontologies in different languages have been developed independently. The integration of these existing ontologies is useful for many applications. Aligning concepts between ontologies is often done by humans, which is an expensive and time-consuming process. This motivates us to find an automatic method to perform such task. However, the hierarchical structures of ontologies are quite different. The structural inconsistency is a common problem [1]. Developing a practical algorithm that is able to deal with this problem is a challenging issue.

The objective of this research is to investigate an automated technique for ontology alignment. The proposed algorithm links concepts between two ontologies, namely the MMT semantic hierarchy and the EDR concept dictionary. The algorithm finds the most appropriate target concept for a given source concept in the top-down manner. The experimental results show that the algorithm can find reasonable concept mapping between these ontologies. Moreover, the results also suggest that this algorithm is able to detect flaws and inconsistency in the ontologies. These results can be used for developing and improving the ontologies by lexicographers.

The rest of this paper is organized as follows: Section 2 discusses related work. Section 3 provides the description of the proposed algorithm. Section 4 presents experimental results and discussion. Finally, Section 5 concludes our work.

R. Dale et al. (Eds.): IJCNLP 2005, LNAI 3651, pp. 346–356, 2005.

2 Related Work

Daudé *et al.* [2] used a relaxation labeling algorithm – a constraint satisfaction algorithm – to map the verbal, adjectival and adverbial parts between two different WordNet versions, namely WordNet 1.5 and WordNet 1.6. The structural constraints are used by the algorithm to adjust the weights for the connections between WN1.5 and WN1.6. Later, some non-structural constraints are included in order to improve the performance [3].

Asanoma [4] presented an alignment technique between the noun part of WordNet and Goi-Taikei 's Ontology. The proposed technique utilizes sets of Japanese and/or English words and semantic classes from dictionaries in an MT system, namely ALT-J/E.

Chen and Fung [5] proposed an automatic technique to associate the English FrameNet lexical entries to the appropriate Chinese word senses. Each FrameNet lexical entry is linked to Chinese word senses of a Chinese ontology database called HowNet. In the beginning, each FrameNet lexical entry is associated with Chinese word senses whose part-of-speech is the same and Chinese word/phrase is one of the translations. In the second stage of the algorithm, some links are pruned out by analyzing contextual lexical entries from the same semantic frame. In the last stage, some pruned links are recovered if their scores are greater than the calculated threshold value.

Ngai *et al.* [6] also conducted some experiments by using HowNet. They presented a method for performing alignment between HowNet and WordNet. They used a word-vector based method which was adopted from techniques used in machine translation and information retrieval. Recently, Yeh *et al.* [7] constructed a bilingual ontology by aligning Chinese words in HowNet with corresponding synsets defined in WordNet. Their alignment approach utilized the co-occurrence of words in a parallel bilingual corpus.

Khan and Hovy [8] presented an algorithm to combine an Arabic-English dictionary with WordNet. Their algorithm also tries to find links from Arabic words to WordNet first. Then, the algorithm prunes out some links by trying to find a generalization concept.

Doan *et al.* [9] proposed a three steps approach for mapping between ontologies on the semantic web. The first step used machine learning techniques to determine the joint distribution of any concept pair. Then, a user-supplied similarity function is used to compute similarity of concept pairs based on the joint distribution from the first step. In the final step, a relaxation labeling algorithm is used to find the mapping configuration based on the similarity from the previous step.

3 Proposed Algorithm

In this section, we describe an approach for ontology alignment based on term distribution. To alleviate the structural computation problem, we assume that the considered ontology structure has only the hierarchical (or taxonomic) relation. One may simply think of this ontology structure as a general tree, where each node of the tree is equivalent to a concept.

Given two ontologies called the source ontology $\mathcal{T}_s$ and the target ontology $\mathcal{T}_t$, our objective is to align all concepts (or semantic classes) between these two ontologies. Each ontology consists of the concepts, denoted by $\mathcal{C}_1, \ldots, \mathcal{C}_k$. In general, the concepts and their corresponding relations of each ontology can be significantly different due to the theoretical background used in the construction process. However, for the lexical ontologies such as the MMT semantic hierarchy and the EDR concept dictionary, it is possible that the concepts may contain shared members in terms of English words. Thus, we can match the concepts between two ontologies using the similarity of the shared words.

In order to compute the similarity between two concepts, we must also consider their related child concepts. Given a root concept $\mathcal{C}_i$, if we flatten the hierarchy starting from $\mathcal{C}_i$, we obtain a nested cluster, whose largest cluster dominates all sub-clusters. As a result, we can represent the nested cluster with a feature vector $\mathbf{c}_i = (w_1, \ldots, w_{|\mathcal{V}|})^T$, where features are the set of unique English words $\mathcal{V}$ extracted from both ontologies, and w_j is the number of the word j occurring the nested cluster i. We note that a word can occur more than once, since it may be placed in several concepts on the lexical ontology according to its sense.

After concepts are represented with the feature vectors, the similarity between any two concepts can be easily computed. A variety of standard similarity measures exists, such as the *Dice coefficient*, the *Jaccard coefficient*, and the *cosine* similarity [10]. In our work, we require a similarity measure that can reflect the degree of the overlap between two concepts. Thus, the Jaccard coefficient is suitable for our task. Recently, Strehl and Ghosh [11] have proposed a version of the Jaccard coefficient called the *extended Jaccard similarity* that can work with continuous or discrete non-negative features. Let $\|\mathbf{x}_i\|$ be the L_2 norm of a given vector $\mathbf{x}_i$. The extended Jaccard similarity can be calculated as follows:

$$JaccardSim(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - \mathbf{x}_i^T \mathbf{x}_j} . \tag{1}$$

We now describe an iterative algorithm for term-based ontology alignment. As mentioned earlier, we formulate that the ontology structure is in the form of the general tree. Our algorithm aligns the concepts on the source ontology $\mathcal{T}_s$ to the concepts on the target ontology $\mathcal{T}_t$ by performing search and comparison in the top-down manner.

Given a concept $\mathcal{C}_i \in \mathcal{T}_s$, the algorithm attempts to find the most appropriate concept $\mathcal{B}^* \in \mathcal{T}_t$, which is located on an arbitrary level of the hierarchy. The algorithm starts by constructing the feature vectors for the current root concept on the level l and its child concepts on the level $l+1$. It then calculates the similarity scores between a given source concept and candidate target concepts. If the similarity scores of the child concepts are not greater than the root concept, then the algorithm terminates. Otherwise, it selects a child concept having the maximum score to be the new root concept, and iterates the same searching procedure. Algorithms 1 and 2 outline our ontology alignment process.

Algorithm 1. OntologyAlignment

```
input  : The source ontology $\mathcal{T}_s$ and the target ontology $\mathcal{T}_t$.
output : The set of the aligned concepts $\mathcal{A}$.

begin
    Set the starting level, $l \leftarrow 0$;
    while ${\mathcal{T}_s}^{\langle l \rangle} \leq {\mathcal{T}_s}^{\langle max \rangle}$ do
        Find all child concepts on this level, $\{\mathcal{C}_i\}_{i=1}^{k} \in {\mathcal{T}_s}^{\langle l \rangle}$;
        Flatten $\{\mathcal{C}_i\}_{i=1}^{k}$ and build their corresponding feature vectors, $\{\mathbf{c}_i\}_{i=1}^{k}$;
        For each $\mathbf{c}_i$, find the best matched concepts on $\mathcal{T}_t$,
            $\mathcal{B} \leftarrow$ FindBestMatched($\mathbf{c}_i$);
            $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathcal{B}, \mathcal{C}_i\}$;
        Set $l \leftarrow l + 1$;
    end
end
```

Algorithm 2. FindBestMatched($\mathbf{c}_i$)

```
begin
    Set the starting level, $l \leftarrow 0$;
    $BestConcept \leftarrow \mathcal{T}_t$(root concept);
    repeat
        $s_{tmp} \leftarrow JaccardSim(\mathbf{c}_i, BestConcept)$;
        if ${\mathcal{T}_t}^{\langle l \rangle} > {\mathcal{T}_t}^{\langle max \rangle}$ then
            return $BestConcept;$

        Find all child concepts on this level, $\{\mathcal{B}\}_{j=1}^{h} \in {\mathcal{T}_t}^{\langle l \rangle}$;
        Flatten $\{\mathcal{B}_j\}_{j=1}^{h}$ and build corresponding feature vectors, $\{\mathbf{b}_j\}_{i=1}^{h}$;
        $s_{j^*} \leftarrow \text{argmax}_j JaccardSim(\mathbf{c}_i, \{\mathbf{b}_j\}_{j=1}^{h})$;

        if $s_{j^*} > s_{tmp}$ then
            $BestConcept \leftarrow \mathcal{B}_{j^*}$;

        Set $l \leftarrow l + 1$;

    until $BestConcept$ does not change;

    return $BestConcept$;
end
```

Figure 1 shows a simple example that describes how the algorithm works. It begins with finding the most appropriate concept on $\mathcal{T}_t$ for the root concept $1 \in \mathcal{T}_s$. By flattening the hierarchy starting from given concepts ('1' on $\mathcal{T}_s$, and 'a', 'a-b', 'a-c' for $\mathcal{T}_t$), we can represent them with the feature vectors and measure their similarities. On the first iteration, the child concept 'a-c' obtains the maximum score, so it becomes the new root concept. Since the algorithm cannot find improvement on any child concepts in the second iteration, it stops the loop and the target concept 'a-c' is aligned with the source concept '1'. The algorithm proceeds with the same steps by finding the most appropriate concepts on $\mathcal{T}_t$ for the concepts '1-1' and '1-2'. It finally obtains the resulting concepts 'a-c-f' and 'a-c-g', respectively.

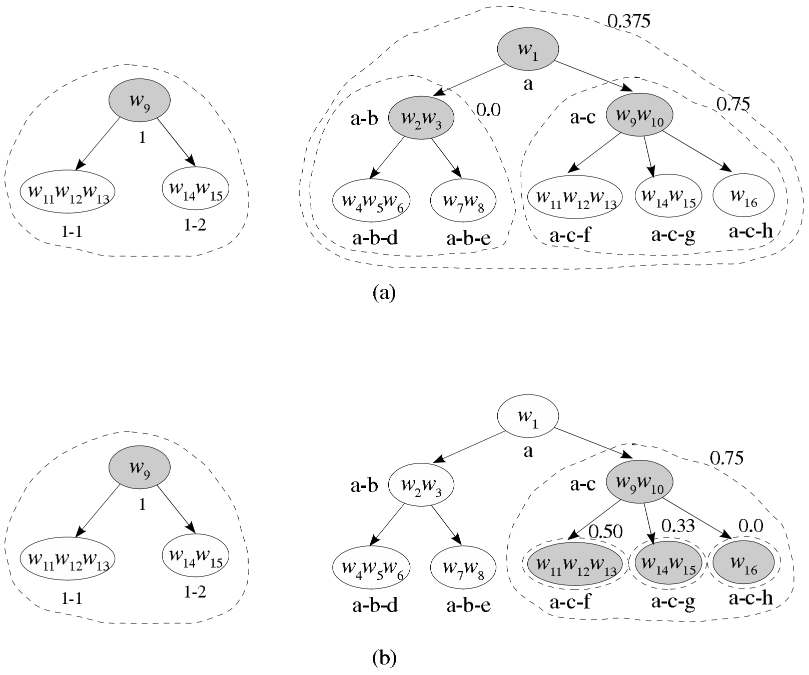

Fig. 1. An example of finding the most appropriate concept on $\mathcal{T}_t$ for the root concept $1 \in \mathcal{T}_s$

4 Experiments and Evaluation

4.1 Data Sets

Two dictionaries are used in our experiments. The first one is the EDR Electronic Dictionary [12]. The second one is the electronic dictionary of Multilingual Machine Translation (MMT) project [13].

The EDR Electronic Dictionary consists of lexical knowledge of Japanese and English divided into several sub-dictionaries (e.g., the word dictionary, the bilingual dictionary, the concept dictionary, and the co-occurrence dictionary) and the EDR corpus. In the revised version (version 1.5), the Japanese word dictionary contains 250,000 words, while the English word dictionary contains 190,000 words. The concept dictionary holds information on the 400,000 concepts that are listed in the word dictionary. Each concept is marked with a unique hexadecimal number.

For the MMT dictionary, we use the Thai-English Bilingual Dictionary that contains around 60,000 lexical entries. The Thai-English Bilingual Dictionary also contains semantic information about the case relations and the word concepts. The word concepts are organized in a manner of semantic hierarchy. Each word concept is a group of lexical entries classified and ordered in a hierarchical level of meanings. The MMT semantic hierarchy is composed of 160 concepts.

In our experiments, we used a portion of the MMT semantic hierarchy and the EDR concept dictionary as the source and the target ontologies, respectively. We considered the 'animal' concept as the root concepts and extracted its related concepts. In the EDR concept dictionary, however, the relations among concepts are very complex and organized in the form of the semantic network. Thus, we pruned some links to transform the network to a tree structure. Starting from the 'animal' concept, there are more than 200 sub-concepts (containing about 7,600 words) in the EDR concept dictionary, and 14 sub-concepts (containing about 400 words) in the MMT semantic hierarchy. It is important to note that these two ontologies are considerably different in terms of the number of concepts and words.

4.2 Experimental Results

The proposed algorithm is used to find appropriate EDR concepts for each one of 14 MMT concepts. The results are shown in Table 1. From the table, there are 6 relations (marked with the symbol '*') that are manually classified as *exact* mapping. This classification is done by inspecting the structures of both ontologies by hand. If the definition of a given MMT concept appears in the EDR concept and the algorithm seems to correctly match the most suitable EDR concept, this mapping will be classified as exact mapping. The remaining 8 MMT concepts, e.g. 'cold-blood' and 'amphibian', are mapped to closely related EDR concepts, although they are not considered to be exact mapping. The EDR concepts found by our algorithm for these 8 MMT concepts are considered to be only the subset of the source concepts. For example, the 'amphibian' concept of the MMT is mapped to the 'toad' concept of the EDR. The analysis in the later section will explain why some MMT concepts are mapped to specific sub-concepts.

Our algorithm works by flattening the hierarchy starting from the considered concept in order to construct a word list represented that concept. The word lists are then compared to match the concepts. In practice, only a portion of word list is intersected. Figure 2 illustrates what happens in general. Note that the EDR concept dictionary is much larger than the MMT semantic

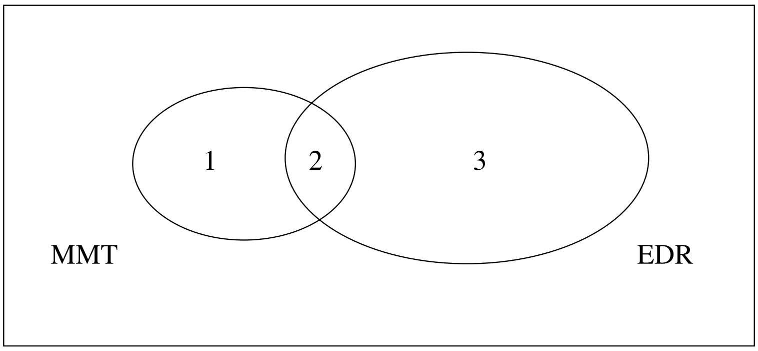

Fig. 2. A schematic of aligned concepts

Table 1. Results of aligned concepts between the MMT and the EDR

<table>
<tr><th>MMT concept</th><th>EDR concept</th></tr>
<tr><td><b>vertebrate</b></td><td>vertebrate *</td></tr>
<tr><td>| → warm-blood</td><td>mammal</td></tr>
<tr><td>| | → mammal</td><td>mammal *</td></tr>
<tr><td>| | → bird</td><td>bird *</td></tr>
<tr><td>|</td><td></td></tr>
<tr><td>| → cold-blood</td><td>reptile</td></tr>
<tr><td>| → fish</td><td>fish *</td></tr>
<tr><td>| → amphibian</td><td>toad</td></tr>
<tr><td>| → reptile</td><td>reptile *</td></tr>
<tr><td>| → snake</td><td>snake *</td></tr>
<tr><td></td><td></td></tr>
<tr><td><b>invertebrate</b></td><td>squid</td></tr>
<tr><td>| → worm</td><td>leech</td></tr>
<tr><td>| → insect</td><td>hornet</td></tr>
<tr><td>| → shellfish</td><td>crab</td></tr>
<tr><td>| → other sea creature</td><td>squid</td></tr>
</table>

* These concepts are manually classified as *exact* mapping.

hierarchy. Thus, it always has EDR words that are not matched with any MMT words. These words are located in the section 3 of the figure 2. The words in the section 1 are more important since they affects the performance of the algorithm. We assume that the EDR is much larger than the MMT. Therefore, most MMT words should be found in the EDR. The MMT words that cannot found any related EDR words may be results of incorrect spellings, specific words (i.e. only found in Thai language). In case of incorrect spelling and other similar problems, the results of the algorithm can be used to improve the MMT ontology.

By analyzing the results, we can classify the MMT words that cannot find any associated EDR words into 4 categories.

1. *Incorrect spelling or wrong grammar :* Some English words in the MMT semantic hierarchy are simply incorrect spelling, or they are written with wrong grammar. For example, one description of a tiger species is written as 'KIND A TIGER'. Actually, this instance should be 'KIND OF A TIGER'. The algorithm can be used to find words that possible have such a problem. Then, the words can be corrected by lexicographers.
2. *Inconsistency :* The English translation of Thai words in the MMT semantic hierarchy was performed by several lexicographers. When dealing with Thai words that do not have exact English words, lexicographers usually enter phrases as descriptions of these words. Since there is no standard of writing the descriptions, these is incompatibility between descriptions that explain the same concept. For example, the following phrases are used to describe fishes that their English names are not known.

- *Species of fish*
- *A kind of fish*
- *Species of fresh water fish*

3. *Thai specific words :* The words that we used in our experiments are animals. Several animals are region specific species. Therefore, they may not have any associated English words. In this case, some words are translated by using short phrases as English descriptions of these Thai words. Another way to translate these words is to use scientific names of species.

The problems mentioned earlier make it more difficult to match concepts by the algorithm. However, we can use the algorithm to identify where the problems occur. Then, we can use these results to improve the MMT ontology.

The proposed algorithm works in the top-down manner. That is, the algorithm attempts to find the most appropriate concept from the top level, and it will move down if the lower concepts yield better scores. In order to analyze the algorithm, we trace the algorithm during moving through the EDR concepts. The first example of the bird concept alignment is shown in Table 2. The concept alignment of this example is considered to be exact mapping. The first column indicates the level of EDR concepts. The second and third columns indicate the number of MMT words and the number of EDR words after flattening respectively. The fourth column shows the number of intersected words between the MMT and the EDR. From the table, the algorithm moves through the EDR concepts in order to find the most specific concept that still maintains shared terms. This example shows that the algorithm passes through 3 concepts until it stops at the 'bird' concept of the EDR. At the final step, the algorithm decides to trade few shared terms for a more specific EDR concept. Note that the MMT is not completely cleaned. When moving down to the EDR bird concept, three shared terms are lost. Our analysis shows that these terms are bat species. They are all wrongly classified to the MMT bird concept by some lexicographers. Thus, these shared terms will not intersect with any words in the EDR bird concept when the algorithm proceeds to the lower step. This result suggests that our algorithm is quite robust. The algorithm still finds an appropriate concept even the MMT ontology has some flaws.

Another analysis of exact mapping is shown in Table 3. The algorithm moves through 4 concepts until matching the EDR snake concept with the MMT snake concept. In this example, the number of members in the MMT snake concept is quite small. However, the number of shared terms is sufficient to correctly locate the EDR snake concept.

Table 2. Concept alignment for the 'bird' concept

Level	MMT words	EDR words	Intersected words
1	67	2112	26
2	67	1288	26
3	67	373	23

Table 3. Concept alignment for the 'snake' concept

Level	MMT words	EDR words	Intersected words
1	17	2112	8
2	17	1288	8
3	17	71	8
4	17	26	8

The third example shown in Table 4 illustrates the case that is considered to be subset mapping. That is, the EDR concept selected by the algorithm is subconcept of the MMT concept. This case happens several times since the EDR is more fine-grained than the MMT. If the members of MMT concept do not cover enough, the algorithm tends to return only sub-concepts. From the table, the MMT amphibian concept covers only toad and frog species (3 members). Thus, the algorithm moves down to a very specific concept, namely the EDR toad concept. Another example of subset mapping is shown in Table 5. This example also shows that the members of MMT concept do not cover enough. These results can be used to improve the MMT ontology. If the MMT concepts are extended enough, we expect that the correctness of alignment should be improved.

Table 4. Concept alignment for the 'amphibian' concept

Level	MMT words	EDR words	Intersected words
1	3	2112	2
2	3	1288	2
3	3	23	2
4	3	16	2
5	3	2	1

Table 5. Concept alignment for the 'other sea creature' concept

Level	MMT words	EDR words	Intersected words
1	17	2112	5
2	17	746	5
3	17	78	3
4	17	3	2

5 Conclusion

We have proposed an iterative algorithm to deal with the problem of automated ontology alignment. This algorithm works in the top-down manner by using the similarity of the terms from each ontology. We use two dictionaries in our experiment, namely the MMT semantic hierarchy and the EDR concept dictionary.

The results show that the algorithm can find reasonable EDR concepts for given MMT concepts. Moreover, the results also suggest that the algorithm can be used as a tool to locate flaws in the MMT ontology. These results can be used to improve the ontology.

There are several possible extensions to this study. The first one is to examine this algorithm with larger data sets or other ontologies. The second one is to improve and correct the ontologies by using the results from the algorithm. Then, we plan to apply this algorithm to the corrected ontologies, and examine the correctness of the results. The third one is to use structural information of ontologies in order to improve the correctness.

References

1. Ide, N. and Véronis, J.: Machine Readable Dictionaries: What have we learned, where do we go?. Proceedings of the International Workshop on the Future of Lexical Research, Beijing, China (1994) 137–146
2. Daudé, J., Padró, L. and Rigau, G.: Mapping WordNets Using Structural Information. Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, (2000)
3. Daudé, J., Padró, L. and Rigau, G.: A Complete WN1.5 to WN1.6 Mapping. Proceedings of NAACL Workshop "WordNet and Other Lexical Resources: Applications, Extensions and Customizations", Pittsburg, PA, United States, (2001)
4. Asanoma, N.: Alignment of Ontologies: WordNet and Goi-Taikei. Proceedings of NAACL Workshop "WordNet and Other Lexical Resources: Applications, Extensions and Customizations", Pittsburg, PA, United States, (2001) 89–94
5. Chen, B. and Fung, P.: Automatic Construction of an English-Chinese Bilingual FrameNet. Proceedings of Human Language Technology conference, Boston, MA (2004) 29–32
6. Ngai, G., Carpuat , M. and Fung, P.: Identifying Concepts Across Languages: A First Step towards a Corpus-based Approach to Automatic Ontology Alignment. Proceedings of the 19th International Conference on Computational Linguistics, Taipei, Taiwan (2002)
7. Yeh, J.-F., Wu, C.-H., Chen, M.-J. and Yu, L.-C.: Automated Alignment and Extraction of a Bilingual Ontology for Cross-Language Domain-Specific Applications. International Journal of Computational Linguistics and Chinese Language Processing. **10** (2005) 35–52
8. Khan, L. and Hovy, E.: Improving the Precision of Lexicon-to-Ontology Alignment Algorithms. Proceedings of AMTA/SIG-IL First Workshop on Interlinguas, San Diego, CA (1997)
9. Doan, A., Madhavan, J., Domingos, P., and Halevy, A.: Learning to Map Between Ontologies on the Semantic Web. Proceedings of the 11th international conference on World Wide Web, ACM Press (2002) 662–673
10. Manning, C. D., and Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press. Cambridge, MA (1999)

11. Strehl, A., Ghosh, J., and Mooney, R. J.: Impact of Similarity Measures on Webpage Clustering. Proceedings of AAAI Workshop on AI for Web Search (2000) 58–64
12. Miyoshi, H., Sugiyama, K., Kobayashi, M. and Ogino, T.: An Overview of the EDR Electronic Dictionary and the Current Status of Its Utilization. Proceedings of the 16th International Conference on Computational Linguistics (1996) 1090–1093
13. CICC: Thai Basic Dictionary. Center of the International Cooperation for Computerization, Technical Report 6-CICC-MT55 (1995)