Toward Domain Specific Thesaurus Construction: Divide-and-Conquer Method

Pum-Mo Ryu and **Jae-Ho Kim** and **Yoonyoung Nam** and **Jin-Xia Huang** and **Saim Shin**
Sheen-Mok Lee and **Key-Sun Choi**
Computer Science Division, KAIST, KORTERM/BOLA
373-1 Guseong-dong Yuseong-gu Daejeon
305-701, Korea
{pmryu,jjaeh,yynam,hgh,mirror,smlee}@world.kaist.ac.kr, kschoi@cs.kaist.ac.kr

Abstract

This paper describes new thesaurus construction method in which class-based, small size thesauruses are constructed and merged as a whole based on domain classification system. This method has advantages in that 1) taxonomy construction complexity is reduced, 2) each class-based thesaurus can be reused in other domain thesaurus, and 3) term distribution per classes in target domain is easily identified. The method is composed of three steps: term extraction step, term classification step, and taxonomy construction step. All steps are balanced approaches of automatic processing and manual verification. We constructed Korean IT domain thesaurus based on proposed method. Because terms are extracted from Korean newspaper and patent corpus in IT domain, the thesaurus includes many Korean neologisms. The thesaurus consists of 81 upper level classes and over 1,000 IT terms.

1 Introduction

A thesaurus is a controlled vocabulary arranged in a known order and structured so that equivalence and hierarchical relationships among terms are displayed clearly and identified by standardized relationship indicators. The primary purposes of a thesaurus are to facilitate retrieval of documents, and to achieve consistency in the indexing of written or otherwise recorded documents and other items, mainly for post-coordinate information storage and retrieval systems (ANSI/NISO, 2003).

Our interest in this paper is the construction of domain specific thesaurus by divide-and-conquer method which minimizes human labor based on step-by-step automatic procedures. Brewster emphasized the problems of thesaurus construction and maintenance (Christopher et al. 2004). First, there is the high initial cost in terms of human labor in performing the editorial task of writing the thesaurus and maintaining it. Secondly, the knowledge which the thesaurus attempts to capture is changing and developing continuously. So thesaurus tends to be out of date as soon as it is published or made available to its intended audience. Thirdly, thesauruses need to be very domain specific. Particular subject areas whether in the engineering or business world have their own technical terminology, thus making a general thesaurus is inappropriate without considerable pruning and editing. So we propose new thesaurus construction process which handles above problems. Firstly, we extract domain terms from domain corpus. This process satisfies the third problem because the terms extracted from domain corpus are mostly composed of technical terminology of the domain. Secondly, we classify the extracted terms using predefined domain classification system and construct class based, small thesauruses. The classification system connects the small thesauruses as a whole. We can reduce

Petr Sojka, Key-Sun Choi, Christiane Fellbaum, Piek Vossen (Eds.): GWC 2006, Proceedings, pp. 69–83.

complexity of thesaurus construction by this divide-and-conquer method.

Especially this method is useful in that we can effectively reuse parts of current thesaurus in the construction of other domain thesaurus when two domains share common areas. So we can tackle out of dated thesaurus problem by rapid construction. Thirdly, we adopted balanced approach of automatic process and manual process in every thesaurus construction steps: term extraction, term classification, relation construction. The problem of high cost of human labor is decreased by automatic procedures, and the inconsistency in manual work is reduced by procedural manuals in each step. It is hard to believe the fully automatic ontology/thesaurus construction without any user involvement (Cimiano et al., 2005). Our balanced approach can be considered as beginning point for effective and practical ontology/thesaurus construction.

The remainder of this paper is organized as follows: Section 1 describes the overview of proposed method which consists of three steps. Section 2 describes the automatic term extraction method and verification guidelines for descriptors. Section 3 describes the automatic term classification and manual verification method. Section 4 also describes the automatic taxonomic relation extraction method and manual verification method. Before concluding, we discuss some related works in section 5.

1 Overview of Methods

Our thesaurus construction process is composed of three steps as shown in Fig. 1: term extraction and descriptor verification step, term classification step, and finally taxonomy construction step.

In the first step, terms are automatically extracted from a domain corpus, and the extracted terms are classified into terms for descriptors and non-descriptors manually based on predefined guidelines. Our term extractor uses many information sources to extract domain terms: existing domain term dictionary, English-Korean transliteration information, term statistics such as term frequency and term temporal salience value.

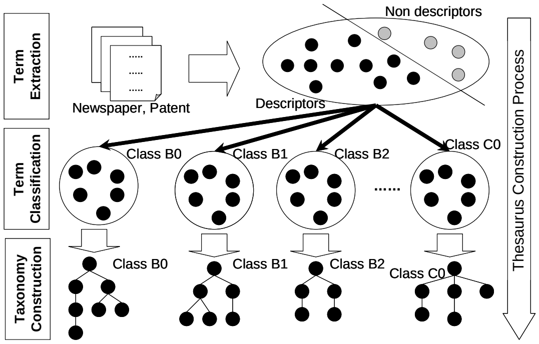

Fig. 1. Overview of thesaurus construction. This method consists of term extraction step, term classification step, and taxonomy construction step.

In the second step, the descriptor terms are classified into predefined classification system. In this process, our classifier assigns most probable semantic classes to the terms automatically, and domain experts verify whether the assigned classes are relevant or not to the terms. The reasons for term classification are simplicity and reusability. In the view of simplicity, it is easier to construct number of small-sized, class-based thesauruses than to construct one large-sized thesaurus at once. In the view of reusability, class-based thesauruses are easily reused to other domain thesauruses because some classes in a domain are also related to other domains. For example *electronic business* class is a part of *information technology* as well as a part of *economics*. So a thesaurus for *electronic business* class in IT domain is also can be used as a part of *economics* thesaurus.

In the third step, our taxonomy prediction system present possible taxonomic relations among terms, and the domain experts validate the presented relations. This step is processed by the unit of classes. The prediction system uses vertical relation method, definition pattern based method, reference thesaurus based method and statistics based method. Domain experts also add relation types to all valid relations. The relation types are abstractions of possible taxonomic relations between terms.

2 Term Extraction and Verification

In this section, we describe a sequence of processes for the construction of Korean IT (Information Technology) domain thesaurus: term extraction, scope note annotation and descriptor selection for thesaurus construction.

2.1 Automatic Term Extraction

In this section, we describe automatic term extraction method from corpus.

Neologisms are rapidly increasing due to the explosion of new domain knowledge. However the neologisms are major hurdles in automatic creation of domain thesaurus. Also the terms which are rarely used currently make the thesaurus construction complex. For this reason, an automatic method for term extraction from corpus is needed in domain specific thesaurus construction process. We use Oh *et al.*'s term extractor (Oh et al., 2000), which is based on domain term dictionary, English-Korean transliteration information and term frequency. The method is usually composed of candidate extraction step and filtering candidates step. Candidate term expressions in texts are usually captured by the shallow syntactic technique called a linguistic filter that describes term formation patterns; morphologically or syntactically parsed sentences are scanned for term formation patterns, which are usually noun phrases (NPs) consisting of at least one noun (Justeson et al., 1995; Frantzi et al., 1999; Maynard et al., 1998). From the analysis of entry words in domain dictionaries – chemical, computer science, and economy – terms are usually noun phrases with the constituents: noun, postposition, and suffix (about 96%) and the rest being composed of verbs, adverbs and so on. Based on the analysis, a linguistic filter that describes candidate noun phrases is used for candidate extraction as shown in Eq. 1.

$$\text{NP} = (\text{Adj} \mid \text{Noun})^{*}\ \text{Noun} \qquad (1)$$

After candidates are extracted from corpus, we use a filtering method of *relevant* terms for a given domain (Oh et al., 2001). The filtering method is based on three scoring function, called dictionary weight (W_{Dic}), transliterated word weight (W_{Trl}), statistical weight (W_{Stat}) each of which support certain characteristics of terms.

Dictionary weight (W_{Dic}) enables the system to extract new terms which are extended from dictionary terms. For example, we can give high scores to a new term '멀티미디어 오브젝트' (*multimedia object*) because it was extended from '오브젝트' (*object*) which is in existing domain term dictionary.

Transliterated word weight (W_{Trl}) is for dealing with terms containing transliterations. In Korean, transliterations and English words are important clues to identify the *relevant* terms because many Korean terms, which come from English terms, contain transliterations as their constituents. When we observe computer science domain dictionary and chemical engineering domain dictionary to investigate the ratio of transliterations in Korean terms, about 53% of entries in the computer science domain dictionary and about 48% of those in the chemical engineering domain dictionary contain transliterations. Therefore, the number of transliterated constituents in candidate terms is one of important clues for identifying "relevant" terms. Transliterated word weight is measured as Eq. 2.

$$W_{Trl}(tc_i) = \frac{\sum_{j=1}^{|tc_i|} trans(tc_{ij})}{|tc_i|} \qquad (2)$$

where tc_i is a candidate term and tc_{ij} is a component word of tc_i. *trans*(tc_{ij}) is the binary function which outputs 1 when tc_{ij} is transliteration or 0 otherwise.

In the Statistical Weight (W_{Stat}), frequencies and *Term Temporal Saliency Value* (*TTSV*) of candidate terms are considered. High frequency terms in domain corpus represent dominant concepts in the domain. We also apply *TTSV* to select terms where annual usage is increasing as

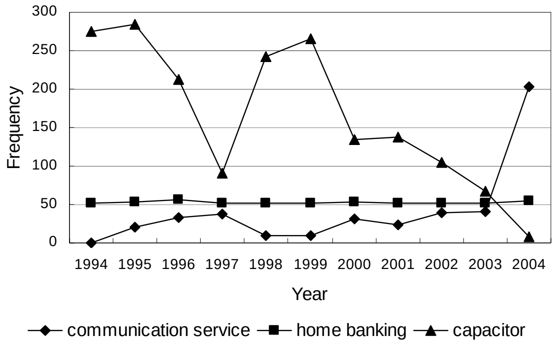

Fig. 2. Annual usage trend of three terms '*통신 서비스*' (*communication service*), '*홈 뱅킹*' (*home banking*), and '*커패시터*' (*capacitor*).

year goes on. *TTSV* is a variation of *TDV* introduced in (Koo et al., 2005) and the value for a term *t* is calculated using Eq. 3.

$$TTSV(t) = \sum_{i=first\_year}^{last\_year} (TF(t,i) - ATF(t)) * w_i \quad (3)$$

where *TF*(*t*,*i*) is term frequency of *t* at year *i*, *ATF*(*t*) is average term frequency of *t* from *first_yrear* to *last_year* and w_{i} is weight of year *i* where recent years get higher weight than past years. Because our corpus consists of newspaper articles and patents from 1994 to 2004, we assigned from 0.0 to 1.0 to the w_{i} s to each year. Intuitively, a term get high *TTSV* when term frequencies of recent years are higher than that of old years. Annual usage trend of three terms are shown in Fig. 2. TTSV of '*통신 서비스*' (*communication servic*) is higher than that of other terms, because annual usage of the term is increasing and that of other terms are decreasing or uniform as year goes on.

W_{Dic}, W_{Stat}, and W_{Trl} described above are combined according to the Eq. 4 called Term Weight (W_{Term}). Because W_{Dic}, W_{Stat}, and W_{Trl} deal with different kinds of terminological characteristic, one of them alone may show limitation on filtering relavent terms. In Eq. 4, W_{Dic}, W_{Stat}, and W_{Trl} are normalized by the functions *f*, *g*, and *h*, because the range of values assigned by W_{Dic}, W_{Stat}, and W_{Trl} is different. α, β, and γ are weighting parameters for W_{Dic}, W_{Stat}, and W_{Trl}, respectively. Though determination of the weighting parameters depends on user preference and domain property, experiments with various settings of weighting parameters (Oh et al., 2001) show that high performance can be acquired when the weighting parameters are between 0.3 and 0.4. Table 1 shows the first part of the *relevant* terms ordered by W_{Term}, when α, β, and γ are the same value.

Table 1. The first 10 terms ordered by W_{Term}

Meaning	Extracted Terms
slot cycle index	*슬롯 사이클 인덱스* *'seul-lot sa-i-keul in-dek-seu'*
roaming service center	*로밍 서비스 서버* *'lo-ming seo-bi-seu seo-beo'*
roaming server	*로밍 서버* *'lo-ming seo-beo'*
data cell	*데이터 셀* *'de-i-teo sel'*
device test program	*디바이스 테스트 프로그램* *'di-ba-i-seu te-seu-teu peu-lo-geu-raem'*
slave	*슬레이브* *'seul-le-i-beu'*
digital signature	*디지털 시그너처* *'di-ji-teol si-geu-ne-cheo'*
service primitive	*서비스 프리미티브* *'seo-bi-seu peu-li-mi-ti-beu'*
frame buffer	*프레임 버퍼* *'peu-le-im beo-peo'*
digital contents server	*디지털 콘텐츠 서버* *'di-ji-tal kon-ten-cheu seo-beo'*

$$\begin{aligned} W_{Term}(tc_i) = {} & \alpha \times f(W_{Dic}(tc_i)) \\ & + \beta \times g(W_{Stat}(tc_i)) \\ & + \gamma \times h(W_{Trl}(tc_i)) \end{aligned} \quad (4)$$
$$\alpha + \beta + \gamma = 1, \quad \alpha, \beta, \gamma \in [0,\ 1]$$

2.2 Descriptor Selection

Concept is a unit of thought, formed by mentally combining some or all of the characteristics of a concrete or abstract, real or imaginary object.

Concepts included in a doain thesaurus are temporally or spatially invariant and represent domain specific knowledge.

Descriptor is a term chosen as the expression of a concept in a thesaurus. Descriptors in a thesaurus should represent a single concept or unit of thought. We classified extracted terms as non descriptors using following criteria. Because the criteria are not always explicit to all terms, the terms are classified by voting of three experts' decisions.

- Proper nouns like person names, location names, and organization names are non descriptors for they denote instances rather concepts. For example, '마이크로소프트' (*Microsoft*) is the name of organization, so we excluded this from set of descriptors.
- Terms having temporal or spatial meaning are non descriptors. For example, '토종 리눅스' (*native Linux*) has spatial information '*native*' and '최근 컴퓨터' (*recent computer*) also has temporal meaning like '*recent*'. Therefore the terms are classified as non descriptors.
- Terms that do not represent domain concepts are non descriptors. For example, '지르코늄' (*Zirconium*) is classified as non descriptor in IT thesaurus because it is a chemistry-domain term.

We made synonym groups of the descriptors based on English translations information and experts' decision. Many Korean domain specific terms are transliteration of English terms. A English term can be expressed to one or more transliterations. For example 'computer' has many Korean transliterations such as '컴퓨터' (keom-pyu-teo), '콤퓨터'(kom-pyu-teo). If two or more descriptors are transliterated from same English term, we group them in a synonym group. Other synomyms taht cannot found the transliteration information are identified by the domain experts. Table 2 shows some synonym groups of descriptors. We select the most frequent descriptor in a synonym set as USE and others as UF. The USE is preferred term in a synonym group, and the UF is synonymous or variant forms of the UF.

Table 2. Examples of synonym groups.

Meaning	**USE** (Freq)	**UF** (Freq)
clocking	클럭킹 (6) '*keul-leog-king*'	클로킹 (1) '*keul-ro-king*'
reboot	재부팅 (24) '*jae-bu-ting*'	리부트 (3) '*ri-bu-teu*'
proxy server	프록시서버 (16) '*peu-rok-si seo-beo*'	대리서버 (1) '*dae-ri seo-beo*'
data bus	데이터버스 (128) '*de-i-teo beo-seu*'	자료 버스 (7) '*ja-ryo beo-seu*'
flash memory cell	플래시메모리소자 (70) '*peul-rae-si me-mo-ri so-ja*'	플래쉬메모리셀 (5) '*peul-rae-swi me-mo-ri sel*'

The scope of a descriptor is restricted to selected meanings within the domain of the thesaurus. Scope note is a piece of text that helps to clarify the meaning of a term. We use term definition from existing domain term dictionary if available, otherwise usages extracted from corpus as scope notes.

2.3 Experiments and Analysis

We extracted terms automatically from IT corpus that consists of Korean patents and newspaper. The size of patent and newspaper are 14 million and 15 million words respectively from year 1994 to 2004. We extracted 765,468 distinct relevant terms which are noun phrases from the corpus. We sorted the relevant terms in the decreasing order of term weight (W_{term}). We selected top 3,688 terms which are up to 10% accumulation frequency of the relavant terms. Three experts classified the terms to descriptors and non-descriptors based on the decision criteria. We classified 3,023 terms as descriptors which are 82.0% of selected terms. 627 terms (17.0%) of 3,023 descriptors are the terms not included in existing IT dictionary. For example,

‘와이브로’ (*WiBro*) is a neologism representing new mobile internet service.

3 Term Classification

It is complex and time-consuming work to construct a large taxonomy using all selected descriptors at once. If terms are grouped by their semantic classes, we can easily build a sub-thesaurus for each class, and we can reuse or share the class-based sub-thesaurus in other domain thesaurus. We classify the terms extracted in section 2 to the classes in the subset of Inspec[1] classification system. After the terms are automatically classified, they are verified manually. We use top three levels of Inspec classification system which consists of 81 classes. If we expand to deeper levels of the classification system, we cannot guarantee the correctness for the data sparseness problem.

3.1 Automatic Classification with the *k*-NN (*k*-Nearest Neighborhood) Method

Our automatic classification system proposes the most suitable *k* classes using the *k*-NN classification method (Hwang et al. 1998). *k*-NN method is a supervised statistical classification method. This method classifies a term to *k* most similar classes measuring distances from the term to all classes. *k*-NN method is well known for its good stability and noise rejection properties. It is important to select a good distance function in this method. The distance between a term and a class is measured using the cosine similarity metric of two feature vectors. The followings are available feature vectors for a term, *t*, and a class *c*:

- The feature vectors for a term *t*
  - V_{tw} : Vector of words which constitutes *t*
  - V_{td} : Vector of nouns in the definition of *t*
  - V_{tu} : Vector of nouns in the usages of *t*
- The feature vectors for a class *c*
  - V_{cw} : Vector of words which constitutes terms in *c*
  - V_{cd} : Vector of nouns in the definitions for terms in *c*
  - V_{cu} : Vector of nouns in the usages for terms in *c*

The weights of each vector is the frequencies of words or nouns in the terms, definitions, and usages depending on the types of vectors. The similarity between *t* and *c* is calculated by Eq. 5. α , β , and γ are weighting schemes and we fix these values with 0.6, 0.3 and 0.1 respectively acquired from the repeated experiments.

$$\begin{aligned} Sim(t,\ c) \;=\; & \alpha \cdot Sim(V_{tl},\ V_{cl}) \\ & + \beta \cdot Sim(V_{td},\ V_{cd}) \\ & + \gamma \cdot Sim(V_{tu},\ V_{cu}) \end{aligned} \tag{5}$$

We decide the value *k*, the number of proposed classes from the *k*-NN for a term, based on sample experiment. Because we found at least one correct class within top 12.98 classes on average for randomly selected 40 terms, we decided *k* as 13.

3.2 Term Class Verification by Experts

Domain experts verify the classes of terms proposed by our automatic classification system. In the verification process, it is more effective to judge whether a term is related to a class or not than to judge a class can have a term as its element or not. Therefore, we rearrange the classification result to class-to-term format as shown in Fig. 3. The domain experts verify the terms in a class whther they can be member of the class or not.

The most important fact in this verification process is that the experts should fully understand the scope of classes and terms. Reviewers catch the scope of classes by referring the class name or the terms included in

[1] Inspec is the English, bibliographic information service providing access to the scientific and technical literature produced by the IEE. We use ‘electrical engineering & electronics’, ‘computer & control’ and ‘information technology’ classes in Inspec classification system.

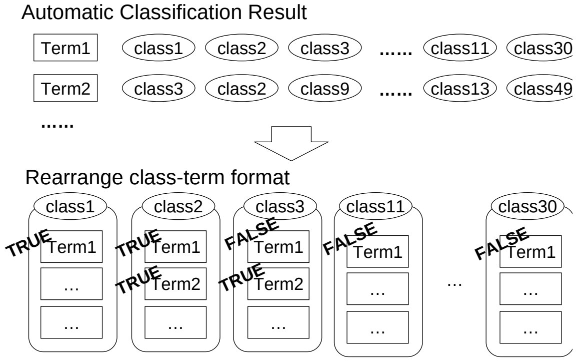

Fig. 3. Result of term classification. Class-term format makes easy the experts' decision.

the classes. Reviewers catch the meaning of a term, *t*, by referring 1) component words of term *t*, especially head word among the component words, 2) definition of term *t*, especially genus term extracted from the definition, and 3) usages of term *t* extracted from corpus. We present all the information in a view so that reviewers easily reference the information.

3.3 Experiments and Analysis

We assigned classes to 2,470 terms among 3,023 terms which were identified as descriptors in section 2. For 553 terms, we didn't find correct classes among 13 classes the system proposed. We assigned 2.99 classes to each term on average.

Fig. 4 shows the number of classified terms to the second level classes. Class *B* (*Electrical Engineering & Electronics*) and class *C* (*Computers & Control*) include more terms than class *D* (*Information Technology*). This means that the corpus from which we extracted terms is concentrated on the two areas.

4 Taxonomy Construction

A taxonomy is a collection of controlled vocabulary terms organized into a hierarchical structure. Terms have one or more hypernym-hyponym relationships to other terms in a taxonomy. There may be different types of taxonomic relationships in a taxonomy such as *is-a*, *part-of*, *instance-of* and other *broader/narrower* relationships.

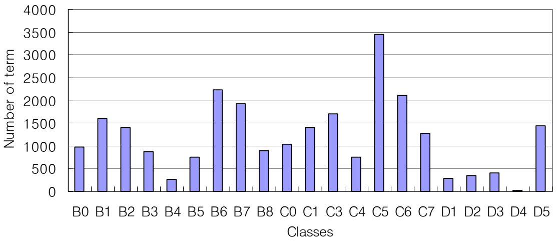

Fig. 4. The distribution of classified terms to second level classes

In this step, small-sized and class-based taxonomies are constructed for each term class. We model taxonomy construction process as a sequential insertion of new terms to current taxonomy. The taxonomy starts with empty state, and changes to rich taxonomic structure with the repeated insertion of terms as depicted in Fig. 5. Terms to be inserted are sorted by term specificity values. Term specificity is a measure for domain specific information for terms under a domain (Ryu et al, 2004). More specific terms usually locate lower part of taxonomy than less specific term. Inserting terms based on the incremental order of term specificity is natural, because the taxonomy increases from top to down under the process of term insertion in this specificity sequence.

Taxonomy construction process is basically composed of following steps:

Repeat the following step until term sequence, *TS*, is empty or no more terms are added to taxonomy, *T*.

1. Sort terms in *TS* in ascending order of term specificity.
2. Assign the first term in *TS* to t_{new}.
3. System proposes possible taxonomic relations for t_{new}, i.e. system find possible hypernyms of t_{new} in *T*.
4. Experts select one or more valid hypernyms and insert t_{new} as a hyponym of the hypernyms to *T*. Go to step 2.
5. If proper hypernyms of t_{new} are not found in the result of step 3, experts manually search hypernyms of t_{new} in *T*.

6. If proper hypernyms of t_{new} are found in step 5, the experts insert t_{new} as a hyponym of the hypernyms to *T*. Go to step 2.
7. If proper hypernyms for t_{new} are not found in the result of step 5, t_{new} goes to the end of *TS*. Go to step 2.

The system's prediction mechanisms minimize the experts' manual task and provide consistency of result taxonomic relations.

4.1 Automatic Taxonomic Relation Extraction

In this section, we illustrate our automatic taxonomic relation extraction method using vertical relation, defintion patterns, reference thesaurus, and term specificity-similarity to extract taxonomic relations between new term and terms in current taxonomy.

4.1.1 Method based on Vertical Relation

When domain specific concepts are embodied into terms, many new terms are created by adding modifiers to existing terms (ISO, 2000). For example '*read only memory*' was created by adding the modifier '*read only*' to its hypernym '*memory*'. Vertical relation is useful taxonomic evidence among terms. For two given terms t_1 and t_2, if t_2 matches t_1 and t_1 is additionally modified by other terms or adjectives, they derive the relation *is-a*(t_1,t_2) (Cimiano et al. 2004; Velardi et al., 2001). However, this method does not always produce correct *is-a* relation. For example, two terms '*exclusive OR gate*' and '*OR gate*' do not related by *is-a* relation rather they are in a sibeling relation.

4.1.2 Method based on Definition Patterns

We apply term definition patterns to extract taxonomic relation from World Wide Web. We firstly search definitions of terms from World Wide Web, and secondly, we extract genus term from the definitions, and finally we generate *is-a* relation between search term and genus term. This method is different from that of Hearst's research (Hearst, 1992) in that our method focuses on term definition patterns. Definitions occur frequently in many types of scientific writing because it is often necessary to define certain operations, substances, objects or machines. We applied definition patterns described in (Pearson, 1998). The most common definition pattern is as follows:

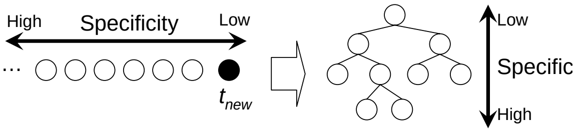

Fig. 5. The terms classified as a class are sequentially inserted to the taxonomy for the class. The system suggests possible locations of a new term, t_{new}, and experts verify the locations.

– A(An) *term* is a(an) *genus term verb*+ed …

For example, we send a query 'a *support vector machine* is a' to Web search engine[2] to find definitions of '*support vector machine*'. One of the searched definition is as follows:

– A *support vector machine* is a *supervised learning algorithm* developed over the past decade by Vapnik and others.

We analyze this definition by applying above definition pattern. '*supervised learning algorithm*' is a genus term of the definition. Finally we make a relation, *is-a*('*support vector machine*', '*supervised learning algorithm*') when '*supervised learning algorithm*' is in current taxonomy. Because the definition patterns for Korean terms are less explicit than English definition patterns, we use English translations to apply this method.

4.1.3 Method based on Reference Thesaurus

The other information source of taxonomic relations is existing thesaurus, such as WordNet[3]. Altohugh WordNet is a domain independent thesaurus, it contains reasonable amount of taxonomic relations for specific domain terms.

[2] We used Google (http://www.google.com) to search definitions of terms.

[3] http://www.cogsci.princeton.edu/~wn

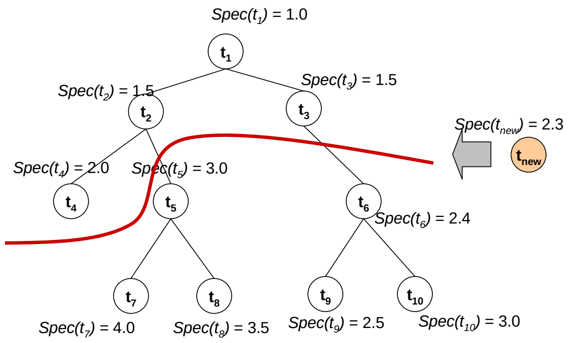

Fig. 6. Selection of candidate hypernyms of t_{new} from current taxonomy using term specificity

For example, '*symbolic logic*' is hypernym of '*Boolean logic*' in WordNet. So if current new term is '*Boolean logic*' and '*symbolic logic*' exists in taxonomy, then '*symbolic logic*' is a possible hypernym of '*Boolean logic*'. Because Korean WordNet is not available, we use English translations to apply this method.

4.1.4 Method based on Term Specificity & Similarity

Specificity is the measure of information quantity that is contained in each term. Because term specificity is the ability of a term to describe topics precisely, it has mainly been discussed in information retrieval in the context of selection of accurate index terms (Aizawa, 2003; Wong et al., 1992). Term specificity can also be applied in the task of taxonomy/thesaurus learning. Because specific terms cover a narrow range in conceptual space and tend to be located at deep levels in a term taxonomy, term specificity is a necessary condition for taxonomic relations, such as *is-a* or *part-of* relations, among terms in a domain (say *D*). That is, if a term t_2 is an ancestor of another term t_4 in a taxonomy, T_D, derived from the domain *D*, then the specificity of t_2 is lower than that of t_4 in *D*. Based on this condition, it is highly probable that t_2 is an ancestor of t_4 in T_D, when t_2 and t_4 are semantically similar enough and the specificity of t_2 is lower than that of t_4 in *D* as in Fig. 6. However, the specificity is not a sufficient enough condition for taxonomic relations, because, for example, t_2 is not similar to t_6 on the semantic level, and t_2 is not an ancestor of t_6 even though the specificity of t_2 is lower than that of t_6 as shown in Fig. 6. According above assumption, our system selects possible hypernyms of a new term, t_{new}, in current taxonomy as following steps:

1. Select candidate hypernyms for a new term, t_{new}, in current taxonomy using term specificity
2. Select *n*-best hypernyms of new term, t_{new}, among the candidate hypernyms selected in step 1 based on term similarity

In Fig. 6, the possible hypernyms of t_{new} are t_1, t_2, t_3 and t_4 because specificity values of the hypernyms are less than that of t_{new}. The possible hypernyms are sorted based on the similarity with t_{new}. Similarity between two terms is the degree of semantic intersection. Term similarity is measured based on compositionality assumption and distributional hypothesis of contextual words. Compositionality assumption refers to the idea that the meaning of a term can be derived from the meaning of its constituent words plus the way these words are combined. Because many domain specific terms are multiword terms, compositional information is useful to measure term similarity. When two terms have many common words, they can be said semantically similar to each other. Distributational hypothesis refers to the idea that the meaning of a term can be derived from the cooccurring words of the term. Therefore if two terms share many common context words, they can be said semantically similar to each other.

4.1.5 Combination of Methods

The taxonomic relation extraction methods have their own pros and cons. We combined the methods to maximize the pros and minimize the cons. We evaluated the methods using precision and recall measures to know the characteristics of the methods. We simulated the our taxonomy construction process using the terms in a part of Inspec thesaurus which consists of 212 terms. Firstly, we assigned speicificity values to the terms according to their levels in thesaurus tree. Terms in high levels have low specificity values,

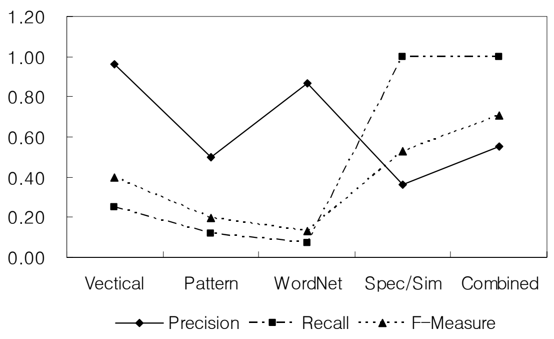

Fig. 7. Precision and recall of the suggested methods in sample test.

and vice versa. Secondly, we reprated term insertion step described at the start of this section. We evaluated each method using precision, recall and F-measure. Precision is the ratio of correct relations over 1-best system suggested relations, and recall is the ratio of correct system suggested relations over all possible taxonomic relations in current taxonomy. We say a relation is *correct* when the system suggested hypernym of new term is real hypernym in the test thesaurus. F-measure is harmonic mean of precision and recall. Fig. 7 showes precision, recall and F-measure of suggested methods in sample experiment. Vertical relation based method showed the best precision in overall methods. Pattern based method and WordNet based method showed relatively high precision. Specificity and similarity based method showed low precision but high recall. We made a pipeline with which we extract hypernym of new term by sequentially applying the methods in the order of vertical relation based method, WordNet based method, pattern based method and specificity and similarity based method. When we cannot find hypernym by current method, we apply next method in pipeline. The combined method showd the best F-measure among all methods.

4.2 Taxonomy Verification

Since system generated taxonomic relations are not always correct, the relations are verified by domain experts. We made a guideline to help verification process. The guideline composed of abstract level relation types compiled from existing Inspec thesaurus. Experts verify extracted taxonomies as following steps:

1. Decide the main facet of suggested relations. A facet is a defining property of a term that distinguishes it from others. The possible facets in IT domain are as follows:

   - Object (A): A view that the relation is between *object* and *object plus other attributes*.
   - Action (B): A view that the relation is between *action* and *action plus other attributes*.
   - Attribute (C): A view that the relation is between *attribute* and *attribute related to other objects*.
   - Technology (D): A view that the relation is between *technology* and *technology plus other attributes*.

For example, a main facet of a taxonomic relation, '*Network*←*Computer network*', is *Object.*

2. Decide specific relation types. A taxonomic relation is composed of the added attributes or object to main facets which make taxonomic relations between two terms. Possible relation types for *Object* and *Action* are as follows. The symbols *A01-B04* represent relation types. The left hand side of ← is an abstraction of hypernym, and right hand side of ← is an abstraction of hyponym in a taxonomic relation. For example, the relation, *A01*, represent a taxonomic relation between an object and constrained form of the object. '*Computer network*' is constrained form of '*Network*', and the formal is hyponym of the latter.

   - *A01* : Object ← *Constraint* on Object
     Ex) network ← *computer* network
   - *A02*: Object ← *Action* of/to Object
     Ex) network ← network *management*
   - *A03*: Object ← *Attribute* of Object

Ex) network ← network *reliability*

- *A04*: Object ← *Instance* of Object
Ex) digital computer← *IBM computer*
- *A05*: Object ← *Part* of Object
Ex) database management system ← *database indexing*
- *A06*: Object ← *Application* of Object
Ex) Internet ← Internet telephony

- *B01* : *Action* ← *Part* of Action
Ex) pattern recognition ← feature extraction
- *B02* : *Action* ← *Constraint* on *Action*
Ex) optimization ← Pareto optimization
- *B03* : *Action* ← *Tool or system* related to Action
Ex) education ← intelligent tutoring system
- *B04* : *Action* ← *Applied technique* of Action
Ex) image processing ← computerized tomography

3. When we can not verify a given relation in step 1 and step 2, we 1) reject the relation or 2) add new guideline appropriate to the relation and accept the relation.

4.3 Experiments and Analysis

We constructed IT domain taxonomy which consists of 1,042 terms. Among the term, 330 terms (31.7%) were inserted to taxonomy based on system suggested relations, whereas 712 terms (68.3%) were inserted to taxonomy entirely based on experts' decision as shown in Fig 8.

Fig. 9 shows the number of relation types for the 568 *Object* based relations. 358 relations (63.0%) are between *Object* and *Constraint on Object* which is a kind of *is-a* relations. The next popular relation type is the relations between *Object* and *Part of Object*. (110 relations, 10.6%)

5. Related Work

In this section, we discuss some work related to thesuaurus construction. Many works have focued on learning method of taxonomic relations based on the distributional hypothesis and lexico-syntactic patterns which convey a certain relation.

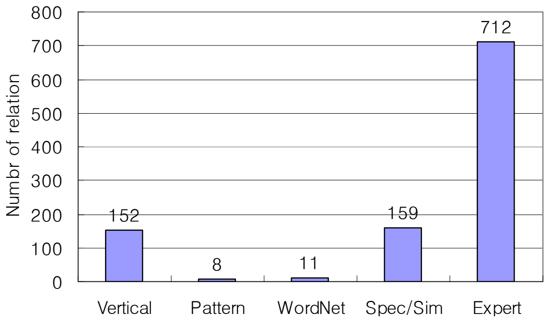

Fig 8. Number of taxonomic relations determined by systems (Vertical relation, Pattern, WordNet, Spec/Sim) and experts (Expert).

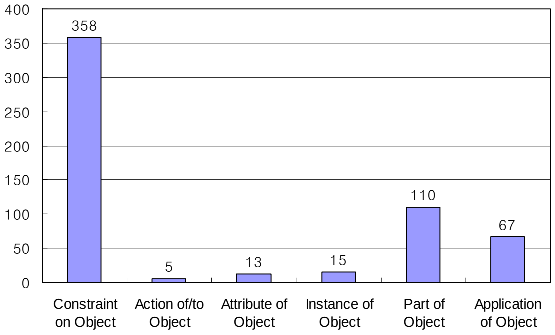

Fig. 9. Number of relation types of *Object* based relations

Pereira (Pereira et al. 1993) present a top-down clustering approach to build an unlabeled hierarchy of nouns. They present an entropy-based evalutaion of their approach. Grefenstette has addressed the automatic construction of thesaurus using SEXTANT system (Grefenstette, 1994). The system used weak syntactic analysis methods on texts to generate thesaurus under the assumption that similar terms will appear in similar syntactic relationships. Terms are then grouped according the grammatical context in which they appear. He presents results on different and various domains. He showed that for frequent words, the syntactic-based approaches are better, while for rare words the window-based approaches are preferable. This method is viable approaches but still do not address the specific relationships between terms, such as *is-a* or *part-of* relations. Faure & Nedellec (Faure et al., 1998) suggested an iterative bottom-up clustering approach of nouns appearing in similar contexts. In each step, they culster the two most similar extents of some argument position of two verbs. Their method is semi-automaitc similar to our method that in that it involves users in the validation of the clusters at each step. Caraballo (Caraballo, 1999)[13] uses clustering methods to derive an unlabeled hierarchy of nouns by using data about conjunctions of nouns and appositions collected from the Wall Street Journal corpus. The final tree is evaluated by presenting a random choice of clusters and the corresponding hypernym to three human judges for validation. This method is also based on distributional hypothesis. Cimiano et al. have presented an approach to the automatic acquisition of taxonomies or concept hierarchies from a text corpus (Cimiano et al., 2004). The approach is based on Formal Concept Analysis (FCA). They followed the distributional hypothesis and modeled the context of a term as a vector. They have also analyzed the impact of a smoothing technique in order to cope with data sparseness and found that it doesn't improve the results of the FCA-based approach. Yamamoto et al. (Yamamoto et al., 2004; Yamomoto et al., 2005) proposed a method of automatically extracting word hierarchies based on the inclusion relations of word appearance patterns in corpora. They applied the complementary simialrity measure (CSM) to determine a hierarchical structure of word meaning. The CSM determines the inclusion between two feature vectors which represent the characteristics of two words respectively. They applied the measure to extract hierarchies of Japanese abstract nouns and evaluated the result by comparing to the hierarchy of EDR electronic dictionary[4]. The approaches based on distributionl hypothesis have some drawbacks in nature. For example, many unrelated terms might co-occur if they are very frequently used. Data sparseness is another problem when we apply the methods to specific domains. Because many domain terms are multi-word terms and they appear in domain corpus relatively low frequency, it is difficult to collect statistically meaningful information from corpus.

There have been many works related to the use of linguistic patterns to discovr certain relations from corpus. Hearst (Hearst, 1992) aimed to discover taxonomic realtions from electronic dictionaries using lexico-syntactic patterns. Her idea has been replied by different researchers with either slight variations in the patterns used (Iwanska et al., 2000), or to discover other kinds of semantic relations such as part-of relations (Charniak & Berland, 1999) or causation relations (Girju & Moldovan, 2002). The pattern based approaches are characterized by a high precision in the sense that the quality of the learned relations is very high compared to the approaches based on distributional hypothesis. However, these approaches suffer from a very low recall due to the fact that the patterns are very rare in real corpus.

Recently researches, covering all the processes of thesaurus/ontology building, have been proposed in the view of ontology engineering. Navigli and Velardi (Navigli et al., 2004) presented a method and a tool, *OntoLearn*, aimed at the extraction of domain ontologies from Web sites, and more generally from documents shared among the members fo virtual organizations. *OntoLearn* first extracts a domain terminology from available documets. Then, complex domain terms are semantically interpreted and arranged in a hierarchical fashion. Finally a general-purpose ontology, WordNet, is trimmed and enriched with the detected domain concepts. The major aspect of this approach is semantic interpretation, that is, the association of a complex concept with a complex term.

[4] EDR Electronic Dictionary (http://www2.nict.go.jp/kk/e416/EDR/index.html)

Conclusions

We have presented a novel approach to acquire domain thesaurus using divide-and-conquer method. This method is composed of three steps: term extraction step, term classification step, and taxonomy construction step. This method has advantages in that 1) taxonomy construction complexity is reduced, 2) each class-based thesaurus can be reused in other domain thesaurus, and 3) term distribution to target domain is easily identified. Though many related works is fully automatic, it is important to mention that it is hard to believe in fully automatic thesaurus construction without any user involvement. In this sense, our approach is balanced in that automatic processing and manual verification do their roles interactively in the construction steps. We have constructed Korean IT domain thesaurus based on proposed method. Because terms are extracted from Korean newspaper and patent corpus in IT domain, the thesaurus includes many neologisms created in Korea. The thesaurus consists of 81 upper level classes and over 1,000 terms.

Though our approach is well organized, there are still many points to be improved in all construction steps. Firstly, a large-scale evaluttion is still to be done. As many researchers have already pointed out, evaluation of ontologies/thesauruses is reconized as an open problem, and few results are available, mostly on the procedural side. So objective, quantitative and procedural evaluation method is needed in future. Secondly, because the reviewer's guidelines in each step have many inconsistent points, it is needed to update the guidelines removing conflicts. We will construct different domain thesaurus in order to validate and update our approach.

References

ANSI/NISO (2003) *Guidelines for the Construction, Format, and Management of Monolingual Thesauri*. ANSI/NISO Z39.19-2003, NISO Press, Bethesda, Mariland, U.S.A.

Christopher, B. and Wilks, Y. (2004) *Ontologies, Taxonomies, Thesauri: Learning from Texts*. In Proceedings of The Use of Computational Linguistics in the Extraction of Keyword Information from Digital Library Content Workshop, Kings College, London, UK

Oh, J., Lee, K., and Choi, K. (2000) *Term Recognition Using Technical Dictionary Hierarchy*. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, pp 496-503

Hwang, W. and Wen K. (1998) *Fast kNN classification algorithm based on partial distance search*. Electronics Letters, Vol. 34, Issue 21, pp. 2062-2063

Ryu, P., Choi K. (2004) *Measuring the Specificity of Terms for Automatic Hierarchy Construction*. In Proceedings of ECAI-2004 Workshop on Ontology Learning and Population

Cimiano, P., Pivk, A., Schmidt-Thieme, L. and Staab, S. (2004) *Learning Taxonomic Relations from Heterogeneous Evidence*. In Proceedings on ECAI-2004 Workshop on Ontology Learning and Population

Velardi, P., Fabriani, P., and Missikoff, M. (2001) *Using Text Processing Techniques to Automatically enrich a Domain Ontology*. In Proceedings of the ACM International Conference on Formal Ontology in Information Systems

Hearst, M. (1992) *Automatic Acquisition of Hyponyms from Large Text Corpora*. In Proceedings of the 14th International Conference on Computational Linguistics

Pearson, J. (1998) *Analysis of Definitions in Text*, Terms in Context (Studies in Corpus Linguistics), Vol. 1, John Benjamins Publishing Company, pp.89-104

Cimiano, P., Hotho, A., Staab, S. (2005) *Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis*. Journal of AI Research, Vol. 24, pp. 305-339

Grefenstette, G. (1994) *Explorations in Automatic Thesaurus Construction*. Kluwer Academic Publishers, Boston, USA

ISO (2000) *Terminology work-Principles and methods*. ISO 704:2000(E)

Caraballo, S. (1999) *Automatic construction of a hypernym-labeled noun hierarchy from text*. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 120-126

Koo, H., Jung, H., Lee, B. and Sung, W. (2005) *Term Extraction and Ranking for Building Term Dictionary*. Proceedings of the 23th Conference of Korea Information Processing (written in Korean)

Aizawa, A. (2003) *An information-theoretic perspective of tf-idf measures*, Journal of Information Processing and Management, Vol. 39

Wong S.K.M., and Yao, Y.Y. (1992) *An Information-Theoretic Measure of Term Specificity*. Journal of the American Society for Information Science, Vol. 43, Num. 1

Yamamoto, E., Kanzaki, K. and Isahara, H. (2005) *Extraction of Hierarchies Based on Inclusion of Co-occurring Words with Frequency Information*. Proceedings of 9th International Joint Conference on Artificial Intelligence, pp. 1160-1167

Yamamoto, E., Kanzaki, K. and Isahara, H. (2004) *Hierarchy Extraction based on Inclusion of appearance*. Proceedings of ACL04 Companion Volume to the Proceedings of the Conference, pp. 149-152

Navigli, R., Velardi, P. (2004) *Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites*. Computational Linguistics Vol. 30, Num. 2, pp. 151-179

Iwanska, L. Mata, N., and Kruger, K. (2000) *Fully automatic acquisition of taxonomic knowledge from large corpora of texts*. In Iwanska, L. & Shapiro, S. (Eds.), Natural Language Processing and Knowledge Processing, pp. 335-345, MIT/AAAI Press.

Charniak, E. and Berland, M. (1999) *Finding parts in very large corpora*. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 57-64

Girju, R. and Moldovan, M. (2002) *Text mining for causal relations*. In Proceeding of the FLAIRS conference, pp. 360-364

Pereira, F., Tishby, N., and Lee, L. (1993) *Distributional clustering of English words*. Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, pp. 183-190

Faure, D. and Nedellec, C. (1998) *A Corpus-based Conceptual Clustering Method for Verb Frames and Ontology*. In Proceedings of the LREC Workshop on Adapting lexical and corpus resources to sublanguages and applications, pp. 5-12

Justeson, J.S. and S.M. Katz (1995) *Technical terminology: some linguistic properties and an algorithm for identification in text*. Natural Language Engineering, 1(1) pp. 9-27

Frantzi, K.T. and Ananiadou S. (1999) *The C-value/NC-value domain independent method for multi-word term extraction*. Journal of Natural Language Processing, 6(3) pp. 145-180

Maynard, D. and Ananiadou, S. (1998) *Acquiring Context Information for Term Disambiguation*. In Proceedings of the First Workshop on Computational Terminology Computerm?8, pp 86-90

Appendix A

Part of constructed taxonomy for class B61 (*Information and Communication Theory*)

Taxonomic Codes	Term	English Translation	Type
533	패턴 인식	pattern recognition	
533.107	특징 추출	feature extraction	B01
533.107.a00	에지 검출	edge detection	B02
8eb	신호 처리	signal processing	
8eb.001	영상 신호 처리	video signal processing	B02
8eb.002	영상 처리	image processing	B02
8eb.002.002	영상 인식	image recognition	B02
8eb.002.002.001	영상 정합	image matching	B01
8eb.002.002.002	에지 검출	edge detection	B01
8eb.002.002.003	지문 인식	fingerprint	B02
8eb.002.003	컴퓨터 비전	computer vision	B04
8eb.002.003.001	머신 비전	machine vision	B02
8eb.002.004	영상 부호화	image coding	B02
8eb.002.006	입체 영상 처리	stereo image processing	B02
8eb.002.007	영상 개선	image enhancement	B02
8eb.002.008	컴퓨터 단층 촬영	computerised tomography	B06
8eb.002.009	영상 표현	image representation	B02
8eb.002.00a	렌더링	rendering	B02
8eb.002.00a.001	광선 추적법	ray tracing	B02
8eb.002.00a.002	볼륨 렌더링	volume rendering	B02
8eb.002.00b	화상 분석	image analysis	B02
8eb.002.00c	영상 변환	image transformation	B02
8eb.002.00d	세선화	thinning	B02
8eb.004	의학 신호 처리	medical signal processing	B02
8eb.005	데이터 압축	data compression	B02
8eb.005.001	벡터 양자화	vector quantization	B02
8eb.015	광 정보 처리	optical information processing	B02
8eb.016	음향 신호 처리	acoustic signal processing	B02
8eb.016.001	음향 합성	acoustic convolution	B02
8eb.023	신호 검출	signal detection	B02
8eb.023.001	차등 검파	differential detection	B02
8eb.023.002	헤테로다인 검파	heterodyne detection	B02
8eb.023.003	동기 검파	homodyne detection	B02
8eb.02c	음성처리	speech processing	B02
8eb.02c.001	음성 인식	speech recognition	B02
8eb.02c.001.001	화자 인식	speaker recognition	B02
8eb.02c.001.002	연속 음성 인식	continuous speech recognition	B02
8eb.02c.001.003	화자 적응	speaker adaptation	B02
8eb.02c.001.004	자동 음성 인식	automatic speech recognition	B02
8eb.02c.003	음성 부호화	speech coding	B02
8eb.02c.004	음성 압축	speech compression	B02
8eb.02c.005	음성 합성	speech synthesis	B02
8eb.02c.006	음성 분석	speech analysis	B02

- Taxonomic codes represent hierarchical structure of terms.