Classification by Selecting Plausible Formal Concepts in a Concept Lattice

Madori Ikeda and Akihito Yamamoto

Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
m.ikeda@iip.ist.i.kyoto-u.ac.jp, akihiro@i.kyoto-u.ac.jp

Abstract. We propose a classification method using a concept lattice, and apply it to thesaurus extension. In natural language processing, solving a practical task by extending many thesauri with a corpus is time-consuming. The task can be represented as classifying a set of test data for each of many sets of training data. The method enables us to decrease the time-cost by avoiding feature selection, which is generally performed for each pair of a set of test data and a set of training data. More precisely, a concept lattice is generated from only a set of test data, and then each formal concept is given a score by using a set of training data. The score represents plausibleness as neighbors of an unknown object, and the unknown object classified into classes to which its neighbors belong. Therefore, once we make the lattice, we can classify test data for each set of training data by only scoring, which has a small computational cost. By experiments using practical thesauri and corpora, we show that our method classifies more accurately than k-nearest neighbor algorithm.

Keywords: Formal Concept, Concept Lattice, Classification, Thesaurus Extension

1 Introduction

In this paper, we propose a method for classifying data that generates a *concept lattice* and selects appropriate *formal concepts* in the lattice. The method enables us to avoid *feature selection* superficially in classification by securing storage enough to maintain both the selected concepts and redundant concepts. This contributes to saving time in solving practical problems concerning with a great variety of large data, e.g. *thesaurus extension*.

Classification can be divided into *feature selection* and classification with the selected features. Selecting features, which affects classification results, is very important, and many methods have been proposed for it [6, 14]. Generally, the selection is executed for a pair of a set of test data and a set of training data. Moreover, the selection is time-consuming when the size of these data is large or many noise are contained in them, as well as when test data are assumed to be classified into multi-classes. Therefore, classifying raw and large test data for each of many sets of training data can be costly from a computational point of view. Our method can overcome the problem. The method generates a concept lattice from a set of test data in advance by following the mathematical definitions naïvely. When a set of training data is given, the method gives each formal concept a *score* by simple calculation using the training data. The score represents plausibleness of the concept as a set of *neighbors* of an unknown object to be classified. Then the method selects some of the concepts based on the score and finds the neighbors. Each unknown object is classified into classes to which its neighbors belong. The method is thus faster because it uses the same concept lattice for every set of training data without feature selection. In addition, we can easily maintain a concept lattice updated with novel test data using well known methods [4, 17, 19]. Storing such a lattice can be costly, since the lattice must store both concepts that end up being selected or not. We claim this disadvantage can be mitigated by the low cost of memory storage.

We apply our method to the problem of thesaurus extension. *Thesauri* are semantic dictionaries of *terms*, and many kinds of thesauri are available now. In almost of all thesauri, each terms have several semantic definitions, and the definitions of terms often vary from a thesaurus to another thesaurus. *Corpora* are also linguistic resources of another type that consist of sentences in a natural language, and recently some of them contain huge amount of sentences with many kinds of characteristics generated and attached to them by applying parsers and syntactic analyzers. *Thesaurus extension* can be regarded as classification and has been researched in the area of NLP (natural language processing) [1, 9, 18]. As classification, a thesaurus is a set of training data, and a corpus is a set of test data. Many proposed methods calculate similarities among terms by using features of them that are selected from the characteristics contained in a corpus. Then an unregistered term is put into the original thesaurus properly by finding registered terms similar to the unregistered term. This is a practical way of the extension because it is so easy to acquire many syntactic characteristics of terms for classifying unregistered terms semantically. However, many thesauri to be extended exist, and the selected features for a thesaurus might not be useful for another thesaurus. Moreover, these linguistic resources are often updated. These methods do not take the practical problems in NLP into account. Our method does not depend on feature selection, but generates a concept lattice without training data, and is robust to update. We apply thesaurus extension to two thesauri and two corpora that are freely available.

This paper is organized as follows. First, we introduce the definitions of formal concepts and concept lattices in the next section. In Section 3, we explain our classification method based on the definitions, and we compare the method with related works in Section 4. In Section 5, we define thesaurus extension in terms of classification, and show our experimental results. Conclusions are placed in Section 6.

2 Formal Concepts and Concept Lattices

We introduce the definitions of *formal concepts* and *concept lattices* according to [5, 7] with a running example.

Table 1. An example context $K_0 = (G_0, M_0, I_0)$

	m_1	m_2	m_3	m_4	m_5	m_6	m_7
g_1	×	×					
g_2	×	×		×			
g_3	×	×		×			
g_4		×		×		×	
g_5		×			×	×	
g_6		×			×	×	
g_7			×		×		×

Definition 1 ([7]) Let G and M be mutually disjoint finite sets, and $I \subseteq G \times M$. Each element of G is called an *object*, and each element of M is called an *attribute*, and $(g, m) \in I$ is read as the object g has the attribute m. A triplet (G, M, I) is called a *formal context* (context for short).

Example 1 We introduce a context $K_0 = (G_0, M_0, I_0)$ as a running example where $G_0 = \{g_1, g_2, ..., g_7\}$, $M_0 = \{m_1, m_2, ..., m_7\}$, and every element of I_0 is represented with a cross in Table 1. For example, the object g_4 has the attributes m_2, m_4, and m_6.

Definition 2 ([7]) For a context (G, M, I), subsets $A \subseteq G$ and $B \subseteq M$, we define $A^I = \{ m \in M \mid \forall g \in A, (g, m) \in I \}$, $B^I = \{ g \in G \mid \forall m \in B, (g, m) \in I \}$. A *formal concept* of the context is a pair (A, B) where $A^I = B$ and $A = B^I$.

Definition 3 ([7]) For a formal concept $c = (A, B)$, A and B are called the *extent* and the *intent* respectively, and we let $\mathrm{Ex}(c) = A$ and $\mathrm{In}(c) = B$. For arbitrary formal concepts c and c', there is an order $c \leq c'$ iff $\mathrm{Ex}(c) \subseteq \mathrm{Ex}(c')$ (or equally $\mathrm{In}(c) \supseteq \mathrm{In}(c')$).

Definition 4 ([7]) The set of all formal concepts of a context $K = (G, M, I)$ with the order $\leq$ is denoted by $\underline{\mathfrak{B}}(G, M, I)$ (for short, $\underline{\mathfrak{B}}(K)$) and is called the *concept lattice* of K. For a concept lattice, the least concept is called the *bottom* and is denoted by $\perp$, and the greatest concept is called the *top* and is denoted by $\top$.

Example 2 The concept lattice $\underline{\mathfrak{B}}(K_0)$ of the context $K_0 = (G_0, M_0, I_0)$ given in Table 1 is shown in Figure 1. Each circle represents a formal concept $c \in \underline{\mathfrak{B}}(K_0)$ with $\mathrm{Ex}(c)$ and $\mathrm{In}(c)$ on its side. The gray concept and numbers called *scores* beside concepts are explained in Section 3. In the figure, each edge represents an order $\leq$ between two concepts, and the greater concept is drawn above, and transitional orders are omitted. In the lattice, the concept c_1 is the top and the concept c_{11} is the bottom.

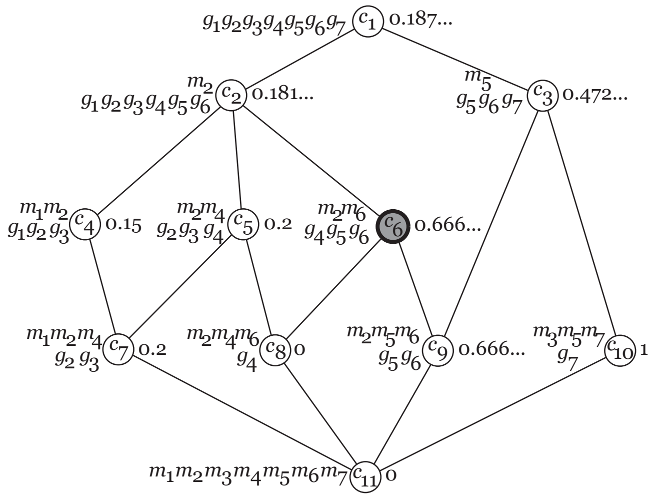

Fig. 1. A concept lattice $\underline{\mathfrak{B}}(K_0)$ with scores

Definition 5 ([7]) For a concept lattice $\underline{\mathfrak{B}}(G, M, I)$, the formal concept $\gamma g = (\{g\}^{II}, \{g\}^I)$ of an object $g \in G$ is called the *object concept*.

Definition 6 ([7]) For every formal concept $c \in \underline{\mathfrak{B}}(K)$, the subset of formal concepts $\{ c' \in \underline{\mathfrak{B}}(K) \mid c' \geq c \}$ is denoted by $\uparrow c$ and called the *principal filter* generated by c.

Definition 7 ([5]) Let S be an ordered set and let $x, y \in S$. We say x is *covered* by y if $x < y$ and $x \leq z < y$ implies $x = z$.

Definition 8 For a concept lattice $\underline{\mathfrak{B}}(K)$, a *path* is a string of formal concepts $c_0, c_1, ..., c_n$ satisfying that c_i is covered by c_{i+1} for every $i \in [0, n-1]$, and its *length* is n.

Example 3 For the concept lattice $\underline{\mathfrak{B}}(K_0)$ shown in Figure 1, $\gamma g_4 = c_8$ and $\uparrow \gamma g_4 = \{ c_1, c_2, c_5, c_6, c_8 \}$. Length of the longest path from $\perp$ to $\top$ is four, and length of the shortest path from $\perp$ to $\top$ is three.

In addition, several algorithms have been proposed [4, 17, 19] in order to update a concept lattice $\underline{\mathfrak{B}}(K)$ when a new object or a new attribute is added to a context $K = (G, M, I)$, e.g. K turns into (G', M', I') where $G' \supset G$, $M' \supset M$, and $I' \supset I$. Thus we can easily modify a concept lattice by using these algorithms.

3 Classification with a Concept Lattice

We illustrate our classification method after formalizing classification problems.

Table 2. An example training set $\tau_0 = (T_0, \mathcal{L}_0)$

$g \in T_0$	g_1	g_2	g_3	g_5	g_6	g_7
$\mathcal{L}_0(g)$	$\{ l_1, l_2 \}$	$\{ l_2, l_3, l_4 \}$	$\{ l_4, l_5, l_6 \}$	$\{ l_1, l_6, l_7 \}$	$\{ l_6, l_7 \}$	$\{ l_1, l_7, l_8 \}$

Definition 9 We let L be a finite set that is disjoint with both G and M, and every element of L is called a *label*. We assume a function $\mathcal{L}_* : G \to 2^L$ as a target classification rule, and $\mathcal{L}_*(g)$ is called the set of *labels* of g.

Each label $l \in L$ represents a class. Note that, for every object $g \in G$, the value of $\mathcal{L}_*(g)$ might not be a singleton and might share some labels with the value of $\mathcal{L}_*(g')$ of another object $g' \in G$, i.e. $\mathcal{L}_*(g) \cap \mathcal{L}_*(g') \neq \emptyset$. Therefore this is a multi-class classification problem and is regarded as an extension of the binary-class classification problems [11–13].

Definition 10 For a subset $T \subseteq G$ and a function $\mathcal{L} : T \to 2^L$ satisfying that $\mathcal{L}(g) = \mathcal{L}_*(g)$ if $g \in T$, a pair $(T, \mathcal{L})$ is called a *training set*. For a training set $(T, \mathcal{L})$, every object $g \in G$ is called an *unknown object* if $g \notin T$, otherwise it is called a *known object*.

Example 4 A training set $\tau_0 = (T_0, \mathcal{L}_0)$ where $T_0 = \{ g_1, g_2, g_3, g_5, g_6, g_7 \}$ and $\mathcal{L}_0 : T_0 \to 2^{\{ l_1, l_2, \ldots, l_8 \}}$ is shown in Table 2. The object g_4 is excluded from G_0 of the context $K_0 = (G_0, M_0, I_0)$ given in Example 1 and is unknown for τ_0.

Classification problems can be defined as obtaining a function $\hat{\mathcal{L}} : G \to 2^L$, and a classification is *successful* when a function $\hat{\mathcal{L}}$ such that $\forall g \in G. \hat{\mathcal{L}}(g) = \mathcal{L}_*(g)$ is obtained from a given training set $(T, \mathcal{L})$.

Our method is designed for classifying only test data that can be expressed as a context (G, M, I) and consists of the following steps.

1. constructing the concept lattice $\mathfrak{B}(K)$ of a context $K = (G, M, I)$,
2. calculating *scores* of formal concepts using a given training set $\tau = (T, \mathcal{L})$,
3. finding the set of *neighbors* for each unknown object $u \in G \setminus T$ based on the scores, and
4. deciding a function $\hat{\mathcal{L}}$ by referring $\mathcal{L}(g)$ for every known object $g \in T$.

The first step is achieved by simply following the definitions of formal concepts.

In order to find the neighbors, formal concepts are given *scores*, and some of the scored concepts are extracted based on their score. The score of every formal concept is a real number calculated with a training set, and it changes for another training set.

Definition 11 For every formal concept $c \in \mathfrak{B}(K)$ and a training set $\tau = (T, \mathcal{L})$, we define $\mathrm{Ex}(c, \tau) = \mathrm{Ex}(c) \cap T$.

Definition 12 For every formal concept $c \in \underline{\mathfrak{B}}(K)$ and a training set $\tau = (T, \mathcal{L})$, we define $\sigma(c, \tau)$ as a real number in $[0, 1]$ and call it the *score* of the concept c under the training set τ. The value $\sigma(c, \tau)$ is calculated as follows:

$$\sigma(c,\tau) = \begin{cases} 0 & \text{if } |\mathrm{Ex}(c,\tau)| = 0, \\ 1 & \text{if } |\mathrm{Ex}(c,\tau)| = 1, \text{ and} \\ \dfrac{\sum_{i=1}^{|\mathrm{Ex}(c,\tau)|-1} \sum_{j=i+1}^{|\mathrm{Ex}(c,\tau)|} \mathrm{sim}(g_i, g_j)}{\dbinom{|\mathrm{Ex}(c,\tau)|}{2}} & \text{otherwise,} \end{cases}$$

where $\mathrm{sim}(g_i, g_j) = \dfrac{|\mathcal{L}(g_i) \cap \mathcal{L}(g_j)|}{|\mathcal{L}(g_i) \cup \mathcal{L}(g_j)|}$.

The function sim calculates similarity between know objects g_i and g_j, and the function σ calculates the average of similarities among objects in $\mathrm{Ex}(c, \tau)$. The purpose of defining the score $\sigma(c, \tau)$ is to estimate similarity among all objects in $\mathrm{Ex}(c)$ that includes not only known objects but also unknown objects.

After scoring formal concepts, the set of *neighbors* is found for each unknown object based on the scores. The neighbors are known objects extracted from the extent of *plausible* concepts.

Definition 13 For every object $g \in G$ in a concept lattice $\underline{\mathfrak{B}}(G, M, I)$ and a training set $\tau = (T, \mathcal{L})$, a formal concept $c \in \uparrow\gamma g$ is called *plausible* w.r.t. g and τ if $\sigma(c, \tau) \geq \sigma(c', \tau)$ for any other concept $c' \in \uparrow\gamma g$ and $|\mathrm{Ex}(c)| \geq |\mathrm{Ex}(c'')|$ for any other concept $c'' \in \uparrow\gamma g$ such that $\sigma(c, \tau) = \sigma(c'', \tau)$. The set of plausible formal concepts w.r.t. g and τ is denoted by $\mathrm{p}(g, \tau) \subseteq \uparrow\gamma g$.

We intend the score of a formal concept c to represent similarities among objects in $\mathrm{Ex}(c)$. We therefore define objects in $\mathrm{Ex}(c) \ni u$ of the concept c that has the highest score as neighbors of an unknown object u. However, it sometimes happens that some formal concepts have the highest score at the same time. In this case, we define a concept c consisting of the largest size $\mathrm{Ex}(c)$ as a plausible concept. This is based on our policy that, among formal concepts that have the same score, the larger the size of $\mathrm{Ex}(c)$ is, the less the concept c has noises in $\mathrm{Ex}(c)$.

Definition 14 For every unknown object $u \in G \setminus T$ under a concept lattice $\underline{\mathfrak{B}}(G, M, I)$ and a training set $\tau = (T, \mathcal{L})$, a set $\mathrm{N}(u, \tau)$ of *neighbors* is defined as

$$\mathrm{N}(u, \tau) = \bigcup_{c \in \mathrm{p}(u,\tau)} \mathrm{Ex}(c, \tau).$$

At the last step, a function $\hat{\mathcal{L}}$ is constructed as

$$\hat{\mathcal{L}}(g) = \begin{cases} \bigcup_{g' \in \mathrm{N}(g,\tau)} \mathcal{L}(g') & \text{if } g \text{ is unknown for } \tau = (T, \mathcal{L}), \\ \mathcal{L}(g) & \text{otherwise.} \end{cases}$$

In this paper, we employed this simple definition although it could be defined by many ways.

Example 5 Suppose that we obtained the context $K_0 = (G_0, M_0, I_0)$ given in Example 1 and constructed the concept lattice $\underline{\mathfrak{B}}(K_0)$ as shown in Figure 1 at the first step. Then, suppose that the training set $\tau_0 = (T_0, \mathcal{L}_0)$ shown in Example 4 was given. The score $\sigma(c, \tau_0)$ of every formal concept $c \in \underline{\mathfrak{B}}(K_0)$ under the training set τ_0 can be calculated at the second step and is shown as the number beside each formal concept c in Figure 1. Plausible formal concepts of the unknown object g_4 decided as the third step are represented as gray and bold circles in Figure 1. There is only one plausible concept c_6, and thus $\mathrm{N}(g_4, \tau_0) = \{ g_5, g_6 \}$. Finally, we can obtain a function $\hat{\mathcal{L}}_0$ at the last step, and $\hat{\mathcal{L}}_0(g_4) = \{ l_1, l_6, l_7 \}$.

4 Comparison with Related Works

We concern a task classifying objects in a context for each of many training sets, and we assume that training sets have multi-classes, and that an object might be classified into several classes. This is a practical task in NLP (natural language processing) that is extending many thesauri by using a corpus. Classification results required by every pair of the context and a training set must be different each other. Classification thus needs to be executed for each of the pairs, and the greater the number of the pairs is, the more solving the task is time-consuming. Our research is motivated to save the time required for the task, and our classification method can overcome the problem. The proposed method constructs a concept lattice from a context in advance, and then the method classifies unknown objects of a training set given later. The same concept lattice is used repeatedly in order to classify unknown objects of each training set, and each classification is performed by scoring formal concepts and finding neighbors of unknown objects. This is based on an idea that as many processes requiring no training sets as possible should be executed before training sets are given. Because of this idea, our method is different from some researches.

Learning models based on a concept lattice are proposed in [11–13]. In these researches, a hypothesis (a set of attributes) is constructed from positive and negative examples (objects) by using a formal concept lattice, and it is determined whether an unknown object is positive or not when the hypothesis is acquired. This is a binary-class classification problem, but unknown objects sometimes are not classified when hypotheses are not constructed appropriately in these approaches. Our method certainly classifies unknown objects into malt-classes by scoring formal concepts. In order to classifying data, some approaches generate decision trees from concept lattices [2, 12]. The decision tree is generated for a pair of a context and a training set. Our method however is not intended to generate decision trees because manipulating a concept lattice that is already constructed is not preferable in order to reduce the time for the task we concern.

As classification for a pair of a context and a training set, our method is similar to k-NN (k-nearest neighbor algorithm) [3] on the point that they find and refer neighbors of an unknown object in order to classify it. However, our method often decreases the number of the neighbors and does not need to be

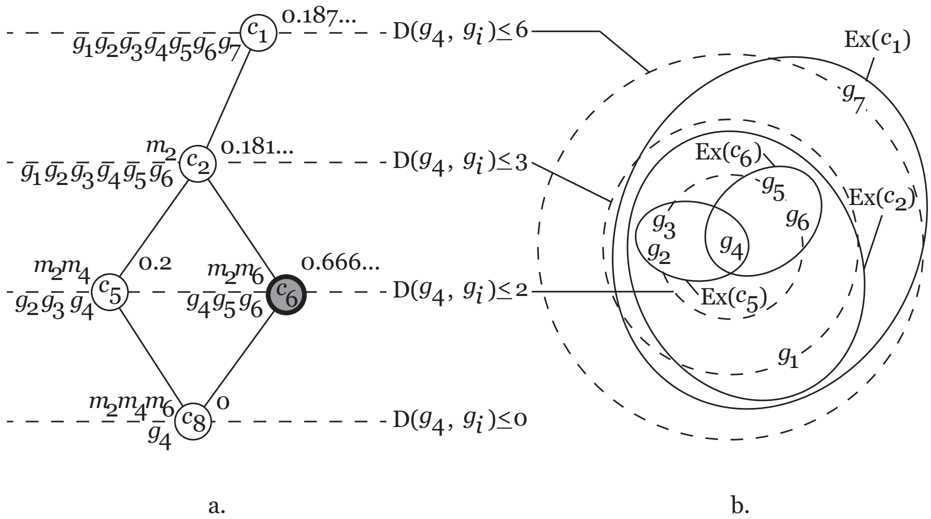

Fig. 2. Formal concepts $\uparrow\gamma g_4$ in the concept lattice $\underline{\mathfrak{B}}(K_0)$ with distance

given the number. In this section, we illustrate such differences between the two methods with an example, and we describe that the differences cause our method to classify more accurately.

In the illustration, we use the *symmetric difference* for measuring dissimilarity between two objects.

Definition 15 For two objects $g, g' \in G$ of a context (G, M, I), we define a distance $\mathrm{D}(g, g')$ between g and g' as

$$\mathrm{D}(g, g') = |\{g\}^I \cup \{g'\}^I| - |\{g\}^I \cap \{g'\}^I|.$$

This distance is also known as the *Hamming distance* between bit vectors in information theory. Figure 2.a shows a part of the concept lattice $\underline{\mathfrak{B}}(G_0, M_0, I_0)$ that is a set of formal concepts $\uparrow\gamma g_4$ with the order $\leq$, and Figure 2.b shows a space that every object $g \in G_0$ is placed according to the distance $\mathrm{D}(g_4, g)$ from the unknown object g_4. From these figures, we observe that each formal concept $c \in \uparrow\gamma g$, as the extent $\mathrm{Ex}(c)$, represents a set of objects located within a certain distance from an object g. It is also found that the greater a concept is, the more the concept includes many dissimilar objects, i.e. for every $g \in G$ and $c', c'' \in\uparrow\gamma g$, $\max(\{\mathrm{D}(g, g') \mid g' \in \mathrm{Ex}(c')\}) \leq \max(\{\mathrm{D}(g, g'') \mid g'' \in \mathrm{Ex}(c'')\})$ if $c' \leq c''$.

Suppose that we have the context $K_0 = (G_0, M_0, I_0)$ given in Example 1 and the training set $\tau_0 = (T_0, \mathcal{L}_0)$ given in Example 4, and that we have to find neighbors $\mathrm{N}(g_4, \tau_0)$ in order to complete a function $\hat{\mathcal{L}_0}$. As Figure 2.b shows, the known objects g_2, g_3, g_5, and g_6 are nearest to the unknown object g_4. In adopting k-NN, each of the four objects is equally a candidate of a neighbor of g_4.

When $k > 4$, we have to find $k-4$ more candidates that are less similar to g_4 than g_2, g_3, g_5, g_6, and such candidates might be noises and might decrease accuracy of obtained function $\hat{\mathcal{L}}_0$. Otherwise, we need to reject $4-k$ candidates from g_2, g_3, g_5, g_6 according to some policy. The rejection also affects the accuracy if values of the function $\mathcal{L}_0$ for the candidates are different. In this case, the values of $\mathcal{L}_0(g_2)$, $\mathcal{L}_0(g_3)$, $\mathcal{L}_0(g_5)$, and $\mathcal{L}_0(g_6)$ are mutually distinct, and the value of $\hat{\mathcal{L}}_0(g_4)$ varies depending on remaining candidates. Therefore, the fixed number k of k-NN may lead accuracy of obtained function $\hat{\mathcal{L}}$ to be decreased. Generally, for the purpose of avoiding such problems, feature selection is repeated for each training set in advance.

By contrast, the nearest objects g_2, g_3, g_5, and g_6 are divided into two extents $\mathrm{Ex}(c_5) = \{ g_2, g_3, g_4 \}$ and $\mathrm{Ex}(c_6) = \{ g_4, g_5, g_6 \}$ in our method. The concepts c_5 and c_6 are discriminated by their scores, and c_6 is more plausible than c_5. Consequently, the objects $g_2, g_3 \in \mathrm{Ex}(c_5)$ are neighbors of the unknown object g_4. Therefore, the number of the neighbors is often less than one in k-NN. Moreover, the number is depends on the size of the extent of every plausible concept, so the number k is not necessary in our method.

We claim that the process of our method, dividing candidates of neighbors into extents and discriminating them by scores, improves both the *precision* and the *recall* of an obtained function $\hat{\mathcal{L}}$.

Definition 16 Under a target function $\mathcal{L}_*$ and an obtained function $\hat{\mathcal{L}}$, the *precision* $\mathrm{prec}(g)$ and the *recall* $\mathrm{rec}(g)$ for every object $g \in G$ is defined as

$$\mathrm{prec}(g) = \frac{|\hat{\mathcal{L}}(g) \cap \mathcal{L}_*(g)|}{|\hat{\mathcal{L}}(g)|}, \quad \mathrm{rec}(g) = \frac{|\hat{\mathcal{L}}(g) \cap \mathcal{L}_*(g)|}{|\mathcal{L}_*(g)|}.$$

Generally, a larger number k of candidates of neighbors results in a lower precision and a higher recall. While k is fixed for all unknown objects in k-NN, it is flexible for each unknown object in our method. More precisely, our method tries to make a precision higher by making k for an unknown object less, but the method also tries to make a recall higher by making k greater when it can keep a precision high. Thus, the two values are better than ones in k-NN. We confirm this assertion by experiments in the next section.

5 Thesaurus Extension and Experiments

We cast the problem of *thesaurus extension* as a classification problem to which we apply the method proposed in this paper. A *thesaurus* is a dictionary registering *terms* based on their *senses*. It is common to all thesauri available now that every registered term (known object) is sorted according to some senses (classes). Extending a thesaurus is putting an unregistered term (unknown object) on some proper positions corresponding to senses in the thesaurus. It is a classification problem defined in Section 3 when we regard a training set $\tau = (T, \mathcal{L})$ as an original thesaurus, T as a set of registered terms, $\mathcal{L}(g)$ for every term $g \in T$ as a set of labels identifying its senses, and unknown objects as unregistered terms.

In this section, we compare our method with k-NN (k-nearest neighbor algorithm) [3] on the point of accuracy by experiments before illustrating resources used in the experiments.

5.1 Contexts and Training Sets for Experiments

We prepared two Japanese corpora, a case frame corpus published by Gengo-Shigen-Kyokai (which means "Language Resource Association" in English) [8] and the Google 4-gram [10], and two Japanese thesauri, Japanese WordNet 1.0 [15] and Bunruigoihyo [16]. We generated the set G_1 of 7,636 nouns contained by all of these linguistic resources. For our experiments, we constructed a context $K_1 = (G_1, M_1, I_1)$ from the case frame corpus and a context $K_2 = (G_1, M_2, I_2)$ from the Google 4-gram so that they satisfy $I_1 \cap I_2 = \emptyset$. Additionally, we construct the third context $K_3 = (G_1, M_3, I_3)$ where $M_3 = M_1 \cup M_2$ and $I_3 = I_1 \cup I_2$. We also constructed a pair $(G_1, \mathcal{L}_{1*})$ from Japanese WordNet 1.0, and $(G_1, \mathcal{L}_{2*})$ from Bunruigoihyo. We generated ten training sets from each pair in each experiment adopting 10-fold cross validation.

Table 3 shows the statistics for the three concept lattices of the three contexts. Numbers on the row beginning with "max $|\{g\}^{I_i}|$", "min $|\{g\}^{I_i}|$", and "mean $|\{g\}^{I_i}|$" respectively represent the maximum, the minimum, and the mean of the numbers of attributes that an object $g \in G_1$ has. Numbers on the row of "mean $|\mathrm{Ex}(c)|$" represent the mean of $|\mathrm{Ex}(c)|$ of every formal concept $c \in \underline{\mathfrak{B}}(K_i)$ for $i \in \{1, 2, 3\}$. Every numbers on the row beginning with "max height" and "min height" respectively indicate length of the longest and the shortest path from $\perp$ to $\top$. Observing the table, we find that every object has a few attributes in all of the three contexts, and that every formal concept $c \in \underline{\mathfrak{B}}(K_i)$ splits the objects into small groups. In other words, formal contexts constructed from the practical corpora are very sparse, and, for all concepts, not so many objects are needed to score it.

Table 4 shows the statistics for the pairs. Numbers on the row beginning with "max $|\mathcal{L}_{i*}(g)|$", "min $|\mathcal{L}_{i*}(g)|$", and "mean $|\mathcal{L}_{i*}(g)|$" respectively indicate the maximum, the minimum, and the mean of $|\mathcal{L}_{i*}(g)|$ of every object $g \in G_1$ for $i \in \{1, 2\}$. Note that, in both of the two thesauri, many terms have several senses, and many senses are represented by several terms, i.e. $|\mathcal{L}_{i_*}(g)| > 1$ and $\mathcal{L}_{i_*}(g) \cap \mathcal{L}_{i_*}(g') \neq \emptyset$ for many terms $g, g' \in G_1$. They have quite different definitions of terms. Moreover, we have to note that the two thesauri share no identifiers of senses, i.e. $\mathcal{L}_{1_*}(g) \cap \mathcal{L}_{2_*}(g) = \emptyset$ for every term $g \in G_1$. In the remains of this subsection, we describe contents of the contexts and the pairs in detail.

The case frame corpus, which is used for construct the context K_1, consists of *case frame structures* that are acquired from about 1.6 billion Japanese sentences on the Web by syntactic analysis. In Japanese, every predicate relates to some nouns with some case terms in a sentence, and such relations are called case frame structures. In $K_1 = (G_1, M_1, I_1)$, every element of M_1 is a pair of a predicate and a case term that the predicate relates to a noun in G_1 with the case term in the corpus, and every element of I_1 is such a relation between a noun in G_1 and

Table 3. Statistics for the concept lattices

	$\mathfrak{B}(K_1)$	$\mathfrak{B}(K_2)$	$\mathfrak{B}(K_3)$
$\lvert G_1 \rvert$	7,636	7,636	7,636
$\lvert M_i \rvert$	19,313	7,135	26,448
max $\lvert \{g\}^{I_i} \rvert$	17	27	32
min $\lvert \{g\}^{I_i} \rvert$	1	1	2
mean $\lvert \{g\}^{I_i} \rvert$	3.85	4.70	8.55
$\lvert \mathfrak{B}(K_i) \rvert$	11,960	20,066	30,540
mean $\lvert \mathrm{Ex}(c) \rvert$	2.55	6.04	4.89
max height	6	9	10
min height	2	2	2

Table 4. Statistics for the thesauri

	$(G_1, \mathcal{L}_{1*})$	$(G_1, \mathcal{L}_{2*})$
$\lvert G_1 \rvert$	7636	7636
$\lvert \bigcup_{g \in G_1} \mathcal{L}_{i*}(g) \rvert$	9560	595
max $\lvert \mathcal{L}_{i*}(g) \rvert$	19	9
min $\lvert \mathcal{L}_{i*}(g) \rvert$	1	1
mean $\lvert \mathcal{L}_{i*}(g) \rvert$	2.19	2.89

Table 5. A context from case frame structures

	⟨hoeru,ga⟩	⟨hoeru,ni⟩
inu	×	
otoko		×

Table 6. A context from 4-grams

	ga	otoko	ni	hoete	iru
inu	×	×	×		
otoko			×	×	×

a pair in M_1. For example, in a Japanese sentence "inu ga otoko ni hoete iru(in English, A dog is barking to a man)", the predicate "hoeru(bark)" relates to the noun "inu(dog)" with the case term "ga(be/do)" and also relates to the noun "otoko(man)" with the case term "ni(to)". These relations are represented as (inu,⟨hoeru,ga⟩) and (otoko,⟨hoeru,ni⟩) respectively and are shown in Table 5. Note that the corpus also has the frequency of each relation, and we used only relations satisfying $0.05 \leq (f/n) \leq 0.95$ where f is the frequency of a relation and n is the sum of the frequencies of relations including the same noun that the relation holds.

The Google 4-gram, which is used to construct the context K_2, is acquired from about 20 billion Japanese sentences on the Web. A Japanese sentence can be regarded as a string of POSs (part of speech) that are words included in the sentence, and a 4-gram is a string of POSs whose length is four. In $K_2 = (G_1, M_2, I_2)$, every element of G_1 is the first POS, and every element of M_2 is POS following the first in a sentence, and every element of I_2 is a relation of "following". For example, from the same sentence "inu ga otoko ni hoete iru" that is a string of six POSs, we can obtain two 4-grams starting with a noun, "inu ga otoko ni" and "otoko ni hoete iru", and the context shown in Table 6 is obtained. This corpus also has the frequency of each 4-gram, and, in order to construct (G_1, M_2, I_2), we use only 4-grams satisfying the condition $0.05 \leq (f/n) \leq 0.95$ (f is the frequency of a 4-gram and n is the sum of the frequencies of 4-grams containing the same noun the 4-gram holds).

Japanese WordNet 1.0 is used to construct the pair $(G_1, \mathcal{L}_{1*})$, and we use values called *lemmas* in the thesaurus as terms. Bunruigoihyo is used to construct the pair $(G_1, \mathcal{L}_{2*})$, and we use values called *midashi-honntai* (entry) in

Table 7. Accuracies of obtained functions $\hat{\mathcal{L}}_1$ and $\hat{\mathcal{L}}_2$

		$(G_1, \mathcal{L}_{1*})$		$(G_1, \mathcal{L}_{2*})$	
	method	precision	recall	precision	recall
$K_1 = (G_1, M_1, I_1)$	our method	0.039	0.274	0.164	0.533
	1-NN	0.026	0.024	0.103	0.103
	5-NN	0.007	0.036	0.031	0.150
	10-NN	0.004	0.038	0.016	0.169
$K_2 = (G_1, M_2, I_2)$	our method	0.007	0.079	0.028	0.248
	1-NN	0.007	0.007	0.027	0.027
	5-NN	0.002	0.013	0.014	0.070
	10-NN	0.002	0.018	0.010	0.100
$K_3 = (G_1, M_3, I_3)$	our method	0.030	0.072	0.132	0.250
	1-NN	0.009	0.009	0.039	0.039
	5-NN	0.004	0.018	0.017	0.085
	10-NN	0.002	0.024	0.011	0.116

the thesaurus as terms and use values called *bunrui-bango* (class number) in it as senses.

5.2 Experimental Results

We carried out experiments to compare our method with k-NN (k-nearest neighbor algorithm).

In the experiments, both our method and other methods were executed with six combinations of the three concept lattices $\underline{\mathfrak{B}}(K_i)$ for $i \in \{1, 2, 3\}$ and the two pairs $(G_1, \mathcal{L}_{j*})$ for $j \in \{1, 2\}$. We adopted 10-fold cross validation on all combinations. On each combination, the set of objects G_1 was split into ten sets U_l for $l \in \{1, 2, ..., 10\}$ that are almost equal about their size. The l-th classification was executed for a concept lattice $\underline{\mathfrak{B}}(K_i)$ and a training set $(G_1 \setminus U_l, \mathcal{L}_{j*})$. We used the precision and the recall in order to evaluate accuracy of both methods. The precision and the recall of a method are mutually defined as the means of precisions and recalls of the ten classifications, and the precision and the recall of the l-th classification are defined as the means of the values of the obtained function $\hat{\mathcal{L}}_j$ for unknown objects U_l. We have to note that our research is intended to solve a task classifying unknown objects in a contest for every one of many training sets, but each classification in the experiments was executed for a pair of a context and a training set.

We compare our method with k-NN for $k \in \{1, 5, 10\}$ and show the results in Table 7. In the table, every value is rounded off to the third decimal place. The results shows our method is better than k-NN in both the precision and the recall over the combinations. The results shows that we can conclude that this is due to the fact that our method is more flexible than the others on the numbers of neighbors.

6 Conclusions

We proposed a classification method that uses a concept lattice and introduces scores of concepts in order to find neighbors of each unknown object. The method tries to make a precision higher by making the number of the neighbors less, but the method also tries to make a recall higher by making the number greater when it can keep a precision high. Moreover, the method does not need feature selection by using the lattice and the scores. We intend the method to apply a task that classifies a set of test data for every one of many sets of training data. The task is practical in NLP, e.g. thesaurus extension By avoiding feature selection, the method has an advantage of time-cost in the task. We made sure that our method classifies unknown objects more accurately than k-NN (nearest neighbor algorithm) [3] by experiments applying both of the methods to thesaurus extension.

The classification results given in Figure 7 shows that the accuracy of our method is not enough although it is better than ones of the other method. The results also show that the accuracies vary over combinations of concept lattices and training sets. It is expected that the variation depends especially on the structure of a concept lattice because the structure affects directly what kind of object a formal concept contains in its extent. Therefore analyzing relations among structures of the lattices and classification results would be a future work for improvement in the accuracy.

References

1. Agirre, E., Ansa, O., Hovy, E., Martinez, D.: Enriching Very Large Ontologies Using the WWW. In: Proc. of the ECAI 2000 Workshop gOntology Learningh (2000)
2. Belohlavek, R., De Baets, B., Outrata, J., Vychodil, V.: Inducing Decision Trees via Concept Lattices. International Journal of General Systems, vol. 38, Num. 4, pp. 455–467 (2009)
3. Belur, V., D.: Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos, CA (1991)
4. Choi, V., Huang, Y.: Faster Algorithms for Constructing a Galois Lattice, Enumerating All Maximal Bipartite Cliques and Closed Frequent Sets. In: SIAM Conference on Discrete Mathematics (2006)
5. Davey, B., A., Priestly, H., A.: Introduction to Lattice and Order. Cambridge University Press, Cambridge (2002)
6. Deng, H., Runger, G.: Feature Selection via Regularized Trees. In: Neural Networks (IJCNN), Proc. of the 2012 International Joint Conference on, pp. 1–8. IEEE (2012)
7. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc., Secaucus, NJ (1999)
8. Gengo Shigen Kyokai, `http://www.gsk.or.jp`
9. Jun Wang, J., Ge, N.: Automatic Feature Thesaurus Enrichment: Extracting Generic Terms From Digital Gazetteer. In: Digital Libraries, 2006. JCDL '06. Proc. of the 6th ACM/IEEE-CS Joint Conference on, pp. 326–333. IEEE (2006)
10. Kudo, T., Kazawa, H.: Web Japanese N-gram Version 1, Gengo Shigen Kyokai

11. Kuznetsov, S., O.: Complexity of Learning in Concept Lattices from Positive and Negative Examples. Discrete Applied Mathematics, vol. 142, issues. 1–3, pp. 111–125 (2004)
12. Kuznetsov, S., O.: Machine Learning and Formal Concept Analysis. Lecture Notes in Computer Science, vol. 2961, pp. 287–312 (2004)
13. Kuznetsov, S., O.: Mathematical Aspects of Concept Analysis. Journal of Mathematical Sciences, vol. 80, no. 2, pp. 1654–1698 (1996)
14. Lopez, F., G., Torres, M., G., Melian, B., Perez, J., A., M., Moreno-Vega, J., M.: Solving Feature Subset Selection Problem by a Parallel Scatter Search. European Journal of Operational Research, vol. 169, no. 2, pp. 477–489 (2006)
15. Mok, S., W., H., Gao, H., E., Bond, F.: Using Wordnet to Predict Numeral Classifiers in Chinese and Japanese. In: Proc. of the 6th International Conference of the Global WordNet Association (GWC-2012), Matsue (2012)
16. National Institute for Japanese Language and Linguistics, `http://www.ninjal.ac.jp/archives/goihyo`
17. Soldano, H., Ventos, V., Champesme, M., Forge, D.: Incremental Construction of Alpha Lattices and Association Rules. In: Proc. of the 14th International Conference on Knowledge-based and Intelligent Information and Engineering Systems: Part II (KES'10), pp. 351–360. Springer, Heidelberg (2010)
18. Uramoto, N.: Positioning Unknown Words in a Thesaurus by Using Information Extracted from a Corpus. In: Proc. of the 16th Conference on Computational Linguistics (COLING '96), vol. 2, pp. 956–961. Association for Computational Linguistics, Stroudsburg, PA (1996)
19. Valtchev, P., Missaoui, R.: Building Concept (Galois) Lattices from Parts: Generalizing the Incremental Methods. In: Proc. of the ICCS'01, pp. 290–303. Springer, Heidelberg (2001)