Mining Preorder Relation between Knowledge Units from Text

Jun Liu
MOE KLINNS Lab and SKLMS Lab
Department of Computer Science and Technology
Xi'an Jiaotong University
Xi'an 710049, P.R.China
liukeen@mail.xjtu.edu.cn

Lu Jiang
MOE KLINNS Lab and SKLMS Lab
Department of Computer Science and Technology
Xi'an Jiaotong University
Xi'an 710049, P.R.China
roadjiang@yahoo.com

Zhao Hui Wu
MOE KLINNS Lab and SKLMS Lab
Department of Computer Science and Technology
Xi'an Jiaotong University
Xi'an 710049, P.R.China
wzh@stu.xjtu.edu.cn

Qinghua Zheng
MOE KLINNS Lab and SKLMS Lab
Department of Computer Science and Technology
Xi'an Jiaotong University
Xi'an 710049, P.R.China
qhzheng@mail.xjtu.edu.cn

Yanan Qian
MOE KLINNS Lab and SKLMS Lab
Department of Computer Science and Technology
Xi'an Jiaotong University
Xi'an 710049, P.R.China
yaya213@gmail.com

ABSTRACT
Preorder relation between Knowledge Units (KU) is the precondition for navigation learning. Although possible solutions, existing link mining methods lack the ability of mining preorder relation between knowledge units which are linearly arranged in text. Through the analysis of sample data, we discovered and studied two characteristics of knowledge units: the locality of preorder relation and the distribution asymmetry of domain terms. Based on these two characteristics, a method is presented for mining preorder relation between knowledge units from text documents, which proceeds in three stages. Firstly, the associations between text documents are established according to the distribution asymmetry of domain terms. Secondly, candidate KU-pairs are generated according to the locality of preorder relation. Finally, the preorder relations between KU-pairs are identified by using classification methods. The experimental results show the method can efficiently extract the preorder relation, and reduce the computational complexity caused by the quadratic problem of link mining.

Categories and Subject Descriptors
H.3.6 [Library Automation]: Large text archives

General Terms

Keywords
Knowledge Unit, Text, Preorder Relation, Locality

1. INTRODUCTION
Presently, 80% of domain knowledge is stored in the form of semi-structured or unstructured texts [1] such as TXT, DOC and HTML. As the smallest and integral knowledge object (e.g. definition, theorem, rule and algorithm) in a special domain, knowledge unit exists widely in domain texts within a linear structure [2]. There are potential associations among knowledge units, and the association information usually indicates the knowledge dependence in human being’s cognitive process. For instance, in “Computer Networks”, the preorder relation between knowledge units “Definition of Subnet Mask” and “Subnetting” indicates that a person should learn “Definition of Subnet Mask” before he or she learns “Subnetting”. According to the Constructivism, the structural knowledge organization based on Knowledge units and their relations is an effective way to lessen the cognitive load of individuals [3]. Although a possible approach, annotating potential preorder relations manually not only costs considerable time and effort, but also requests annotators to be experts of the domain.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC’10, March 22-26, 2010, Sierre, Switzerland.
Copyright 2010 ACM 978-1-60558-638-0/10/03…$10.00.
To solve this problem, the paper proposed a text-oriented knowledge unit preorder relation mining method. It is based on the two characteristics of Knowledge units we found in domain corpus: the locality of preorder relation and the distribution asymmetry of domain terms. According to the experimental evaluations, the method can efficiently recognize the preorder relations among knowledge units, and overcome the quadratic problem of link mining.

The remainder of the paper is organized as follows. Section 2 discusses the related work. Section 3 defines the formal representation of knowledge unit preorder relation mining problem. Section 4 presents two characteristics of preorder relation among knowledge units. Section 5 describes the preorder relation mining algorithm. Experimental results are discussed in Section 6. Finally, in section 7, conclusions are drawn and some further research directions are provided.

2. RELATED WORK
To the best of our knowledge, there has been no previous work on knowledge unit relation mining. Many researchers have been engaged in mining other types of relations in text. For example, Ontology Learning or KAT (Knowledge Acquisition from Text) aims to extract concepts of a specific domain and taxonomic relations between concepts from text, while the goal of RDC (Relation Detection and Characterization) is to identify relationship between named entities. In the literature [4], there are mainly three kinds of methods for mining relation, including template-based, clustering-based and classification-based methods [4].

Template-based methods attempt to discover the relations between entities by frequent-used language patterns. For instance, Timothy introduced a method to recognize five types of semantic relations between verbs using non-ambiguous language template [5]. For certain types of relations, the method works well. However, it has the following deficiencies: difficulties in defining templates, dependence on domains, poor scalability.

Clustering-based methods, such as [6], try to cluster entities according to their semantic similarity distances before recognizing relations within the same cluster. The computational complexity is considerably reduced in the way. Nevertheless, since the semantic similarity distances cannot reflect relation types, it simply manages to obtain the anonymous relations.

Classification-based methods employ the grammar rules and statistic models in its framework, in which classifier is built to identify the semantic relations. For example, Fleischman and Hovy described a method for recognizing hyponymy relations by a decision tree classifier in which each relation pair is represented in terms of their template and adjacent features [7]. Zhou et al. proposed entities a relation recognition method based on SVM (Support Vector Machine) [8].

Previous work helps to enlighten the idea of mining knowledge unit preorder. However, due to the characteristics of the issue itself, we are still confronted with two major problems:

1) Knowledge units expressed in natural language are of ambiguity and ill-formedness [9]. Moreover, the structures of knowledge units are far more complex than those of concepts and named entities, resulting in difficulties in feature extraction and representation.

2) Unlike the relations between concepts or named entities, the preorder relations between knowledge units have the characteristic of long distance dependence, indicating that related knowledge units are usually distributed in different paragraphs or even different text documents. Consequently, mining relations of knowledge units requires detecting KU-pairs in a larger context size and leads to the quadratic problem of link mining [10], in which the complexity of mining the knowledge unit relations is proportional to the square of knowledge unit’s number.

Because of these characteristics, the previous work on concept and named entities relation mining can hardly be applied to the issue of knowledge unit preorder mining. As a result, a novel method which takes into account the inherent characteristics is needed.

3. PROBLEM FORMULATION
We now formalize the problem of knowledge unit preorder relation mining.

Given an input of text document set \( T = \{t_k\} \ (1 \leq k \leq m) \), and knowledge unit set \( U = \{u_i\} \ (1 \leq i \leq n) \), the preorder relation mining process will output the preorder relation set \( A \subseteq U \times U \).

Each \( u_i \in U \) can be further represented as a triple (name, type, content), where name denotes name of \( u_i \), such as “definition of subnet mask”; type denotes semantic type of \( u_i \), such as definition, property and method; and content represents the text content of \( u_i \). \((u_i, u_j) \in A\) indicates that \( u_i \) is an immediate precursor of \( u_j \).

<table>
<thead>
<tr>
<th>Table 1. Examples for KU and Preorder Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ID</strong></td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>
4. CHARACTERISTICS ANALYSIS OF PREORDER RELATION BETWEEN KNOWLEDGE UNITS

Through observation and analysis of the preorder relation in our sample data, we have fortunately found some helpful characteristics for mining preorder relation. First, two knowledge units of preorder relation are often close to each other in distance, and by quantitative analysis we find that the distribution of preorder relation under different distance is approximately exponential. Second, the distribution of domain term frequency is asymmetrical in associated text documents in which there are knowledge units of preorder relations. These two characteristics will be studied in the following two sections respectively.

4.1 Locality of Preorder Relation

Using the knowledge unit extraction method we have proposed in [11], we acquired 4318 knowledge units from 336 text documents in “Computer Networks” and “Advanced Mathematics”. Among these knowledge units, 5404 preorder relations were labeled manually. Based on the labeled data set, we counted the preorder relations under different distance in the two domains respectively. Figure 2 shows the distribution of preorder relation pairs’ number under different distance. The x-axis d denotes the distance of two knowledge units of preorder relation, and the y-axis s denotes the ratio of preorder relations under different distance.

The distance of a preorder relation pair \((u_i, u_j)\) can be calculated through the following two rules:

1) If \(u_i\) and \(u_j\) are in the same document, their distance \(d_{ij} = |j' - i'|\). Here \(i'\) or \(j'\) is the ordinal ID of \(u_i\) or \(u_j\) in the document.

2) If \(u_i\) and \(u_j\) are in the document \(t_a\) and \(t_b\), respectively, their distance \(d_{ij} = |j' - i' + n_o|\). Here \(i'\) or \(j'\) is the ordinal identity of \(u_i\) in \(t_a\) or \(u_j\) in \(t_b\), and \(n_o\) is the number of knowledge units in \(t_a\).

Let \(s_d\) denote the proportion of preorder relations of distance \(d\). By curve fitting, we find that \(s_d\) is inversely proportional to exponential function of \(d\), that is, \(s_d \propto e^{-\beta d}\). Here, \(\beta > 0\) is the distribution coefficient of preorder relation pairs. The smaller \(\beta\) is, the more preorder relations will be distributed between closer knowledge units, and the more sharply the curve will decrease as the distance \(d\) enlarges. In figure 2, the distribution coefficient of “Computer Network” and “Advanced Mathematics” are 0.4 and 0.43 respectively. However, this coefficient depends on the specific data set.

We name this characteristic the locality of preorder relation. Since almost all preorder relations are located in knowledge units from the same document or from documents with similar topic. In our data set, the max \(d\) is 10. Thus, when mining preorder relation, we can limit the distance of candidate knowledge unit pairs. If there is total of \(n\) knowledge units and we set the max \(d\) to be \(k\), usually \(k\) is much smaller than \(n\), so the candidate pairs of preorder relation can be reduce from...
\(O(n^2)\) to \(O(n)\), and it will be notably helpful for improving the mining efficiency.

4.2 Distribution Asymmetry of Domain Term

Usually, a knowledge unit is named as “definition case/property of \(A\)”, where “\(A\)” is a term in the related domain. Here, we call “\(A\)” the core term of a knowledge unit. Intuitively, if \(u_i\) is a precursor of \(u_j\), \(u_i\’s\) core term may be more possibly appearing in \(u_i\). Table 1 shows some good examples. Taking the preorder relation pair (‘definition of Internet Protocol’, ‘definition of IP address’) as an example, we can see the core term “Internet Protocol” has appeared in the content of ‘definition of IP address’, but “IP address” is not present in the content of ‘definition of Internet Protocol’.

Let \(C_i\) be the set of knowledge units’ core term in the document \(t_j\). We define the average term frequency of \(C_i\) as 
\[
f(C_i, t_j) = \frac{\sum_{c_k \in C_i} f(c_k, t_j)}{|C_i|},
\]
where \(c_k\) represents the \(k\)th term in \(C_i\), and \(f(c_k, t_j)\) denotes the term frequency of \(c_k\) in \(t_j\).

Through analysis on our data sets, we found the asymmetry distribution of core term: if knowledge units in \(t_j\) are precursors of knowledge units in \(t_i\), then \(f(C_i, t_i) < f(C_i, t_j)\). We take a collection of text documents to analyze the distribution of core terms in two associated text documents. The result is shown in Figure 3. The x-axis indicates the text document’s ID, and the y-axis represents the value of 
\[
F(i) = f(C_{i+1}, t_i)/f(C_i, t_{i+1}).
\]
The blue horizontal line is \(F(i) = 0.73\), which could be considered as a threshold for selecting associated text documents. By calculation, we get \(P(F < 0.73) = 0.92\) and \(P(F = 0) = 0.24\). This implies that by setting a proper threshold we can get a satisfactory precise for acquiring associations of text documents.

![Figure 3. Ratio of Term Frequency](image)

5. MINING PROCESS

According to the locality of preorder relation studied in section 4.1, the preorder relation exists mainly in knowledge units from the same text document or documents of similar topic. Hence, we need only to find preorder relations in these documents, namely associated documents. The mining process is shown in Figure 4, which is divided into three phases: firstly, find the text documents of similar topic through text clustering and distribution asymmetry of term frequency, i.e., text association mining; secondly, generate candidate KU-pairs according to locality of preorder relation in these documents; finally, build a binary classifier to recognize the preorder relation based on features of term, distance and type feature of candidate KU-pairs.

5.1 Text Association Mining

The input of text document set \(T\) is a group of text documents in a specific domain, which lacks explicit association. Thus, text association mining aims at finding the documents of similar topic, and then ranking them in pairs to make the association in accordance with preorder relation of knowledge units.

Let \(R\) be the set of text document associations from \(T\). \((t_i, t_j) \in R \) represents that \(t_i\) and \(t_j\) are associated text documents. \(R\) is initialized as \(\emptyset\). First, each text document in \(T\) is vectorized based on VSM (vector space model) using TF-IDF and the distance of two documents is computed by Euclidean distance, and then the hierarchy clustering algorithm AGNES is applied [12]. The clustering process includes three kinds of conditions:

1) Two documents \(t_i\) and \(t_j\) are gathered into a cluster;
2) A document \(t_i\) is gathered into the cluster \(S\) (assume \(t_i\) in \(S\) are closest to \(t_j\));
3) Cluster \(S\) and cluster \(S’\) are gathered into a new cluster (assume \(t_i\) in \(S\) and \(t_j\) in \(S’\) are closest document pair in \(S\) and \(S’\));

For each pair \((t_i, t_j)\) in either of the above conditions, we need to set a proper threshold \(F_0 < 1\) (0.73 in our experiment), and establish the association of documents according to the characteristic of distribution asymmetry of terms.

\[
\begin{align*}
&\text{If } f(C_i, t_i)/f(C_i, t_j) < F_0, R = R \cup \{(t_i, t_j)\}; \\
&\text{If } f(C_i, t_i)/f(C_i, t_j) > 1/F_0, R = R \cup \{(t_j, t_i)\}; \\
&\text{If } f(C_j, t_i)/f(C_i, t_j) \in [F_0, 1/F_0], t_i \text{ and } t_j \text{ are not associated with each other.}
\end{align*}
\]

Once the clustering is finished, the directed graph \(< T, R >\) is also generated.
5.2 Candidate KU-Pair Generation

According to the locality of preorder relation, the preorder relation can be mined in the associated text documents or a group of text clusters. For each node \( t_i \) in the generated directed graph \( \mathcal{T}, R \succ \), the corresponding text cluster can be defined as \( \mathcal{T}_i = \{ t_i \} \cup \{ t_j \mid (t_i, t_j) \in R \} \). For each knowledge unit in a text cluster, we need only to find whether it has preorder relation with knowledge units in the text cluster, but not all the knowledge units in the whole text document set.

We call a KU-pair in the text cluster a candidate KU-pair if its two knowledge units may have preorder relation. The candidate KU-pair set in text cluster \( \mathcal{T}_i \) can be divided into two subsets:

1) One subset is the candidate KU-pair set \( \mathcal{A}_i \), denoted by \( A_i \). Let \( U_i \) be the knowledge unit set of \( t_i \), then \( A_i = \{ (u_{ix}, u_{iy}) \mid u_{ix}, u_{iy} \in U_i \land x < y \} \), where \( x \) and \( y \) are ID of \( u_{ix} \) and \( u_{iy} \) in \( t_i \).

2) The other subset is the candidate KU-pair set \( A'_i \) where a KU-pair is partnered by a knowledge unit in \( t_i \) and a knowledge unit in another text document. Let \( U'_{i} \) be the knowledge unit set of other text documents from \( T_i - \{ t_i \} \), then \( A'_i = \{ (u_{ix}, u_{iy}) \mid u_{ix} \in U_i \land u_{iy} \in U'_{i} \} \), where \( x \) and \( y \) are ID of \( u_{ix} \) and \( u_{iy} \) respectively.

To sum up, the candidate KU-pair set of a text cluster \( \mathcal{T}_i \) is \( A_i \cup A'_i \). Therefore, candidate KU-pair set of the whole text document set can be presented as \( A_{\text{cand}} = \bigcup_{t_i \in \mathcal{T}} (A_i \cup A'_i) \).

Assume the number of knowledge units in the text document \( t_i \) is \( n_i \), and the number of knowledge units in a text document has an upper limit of \( n_{\text{max}} \). It can be calculated that \( |A_{\text{cand}}| \leq n_{\text{max}} \sum_{t_i \in \mathcal{T}} n_i \), where \( \sum_{t_i \in \mathcal{T}} n_i \) indicates the total number of knowledge units in the whole document set. If we do not make use of the locality of preorder relation to filter out the impossible KU-pairs, the number of candidate KU-pairs will reach \( \sum_{t_i \in \mathcal{T}} n_i \cdot (\sum_{t_i \in \mathcal{T}} n_i - 1) \). Thus, our method reduced the time complexity of the preorder relation mining complexity from quadratic to linear in the number of knowledge units.

5.3 Feature Selection for Preorder Relation Recognition

The remaining issue now is to recognize the preorder relations of given candidate pairs. To this end, we have analyzed features of preorder relations and introduced a binary classification to recognize the potential preorder relations. According to experimental observation and analysis, there are three useful features namely term, distance and type. For the convenience of discussion, the candidate pair of preorder relation is denoted as \((u_f, u_b)\) where \( u_f \) and \( u_b \) are two knowledge units. To a better understanding these features, an example is given as follows:

\( u_f \) (Definition of IP Address): an Internet Protocol (IP) address is a numerical identification and logical address that is assigned to devices participating in a computer network utilizing the Internet Protocol for communication between its nodes.

\( u_b \) (Definition of Subnet Mask): Subnet mask is the sequence of leading bits of an IP address that precede the portion of the address used as host identifier in IPv4 networks.

According to section 4.2, the core term of \( u_f \) is Internet Protocol or IP and the core term of \( u_b \) is subnet mask.

1) Term feature

Term feature \( F_{fb} \) attempts to highlight the asymmetry of term distribution, the feature can be computed from Equation (1).

\[
F_{fb} = \frac{F_f}{F_f + F_b}
\]

(1)

Where \( F_f \) denotes the frequency of core term of \( u_f \) appearing in content of \( u_b \), and similarly \( F_b \) denotes the frequency of core term of \( u_b \) appearing in content of \( u_f \). In the example \( F_{fb} = 1 \) and \( F_{bh} = 0 \). Given a candidate pair \((u_f, u_b)\), obviously \( F_{fb} \in [0, 1] \). And the greater the value is, the more likely that \((u_f, u_b)\) will have a preorder relation.

2) Distance feature

Distance feature \( D_{fb} \) reflects the distance between \( u_f \) and \( u_b \). According to the locality of preorder relation, the feature can be formulated as:

\[
D_{fb} = e^{-\beta d_{fb}}
\]

(2)

where \( d_{fb} \) is the distance between \( u_f \) and \( u_b \). \( D_{fb} \) indicates the possibility of preorder relation existing in \((u_f, u_b)\) decays exponentially while \( d_{fb} \) grows.

3) Type feature

Knowledge units have their semantic types which are classified into eight types, namely definition, property, instance, case, method, classification, distinction and evolution [11]. The semantic type of a knowledge unit usually indicates what kind of information it intends to deliver. According to our statistical analysis on the annotated preorder relations corpus, we found that:

1) \( KP_{\text{ex}} \) is a set containing the five most frequently used patterns of preorder relation. Those five patterns occupy 15.9%, 12.5%, 12.5%, 10.5% and 9.1% of the total corpus respectively.

2) \( KP_{\text{ex}} \) is the set containing the five least frequently used patterns of preorder relation. Those five patterns occupy 2.1%, 1.3%, 1.0%, 0.3% and 0.1% of the total corpus respectively.

The evidences reveal the phenomenon that preorder relation tends to exist in the pair who has the certain combination of types. Ternary digital \( KP_{fb} \) can be used to measure this characteristic:
\[ K_{P_{tb}} = \begin{cases} 1 & (u_f, u_h) \in K_{P_{max}} \\ -1 & (u_f, u_h) \in K_{P_{min}} \\ 0 & \text{otherwise} \end{cases} \] (3)

In the previous example, since the semantic types are both definition, \( K_{P_{tb}} = 0 \).

Based on the above features, we experiment with various kinds of binary classifiers to recognize the potential preorder relations.

### 6. EXPERIMENTAL EVALUATION

In order to validate the efficiency of the method, we conducted experiments on a set of documents about three computer science courses: “Computer Network”, “Computer Organization and Architecture” and “Database System and Application”. To the best of our knowledge, there is no public data set for mining knowledge unit relation.

We established experimental data sets by following two steps. First, we have generated knowledge units in the given corpus using the method of [11], and then we manually annotated preorder relations between the extracted knowledge units. The number of knowledge units and their relations in the data sets are shown in Table 2.

**Table 2. Experimental Data Sets**

<table>
<thead>
<tr>
<th>ID</th>
<th>Course Name</th>
<th>#KUs</th>
<th>#preorder relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Computer Network</td>
<td>1,039</td>
<td>437</td>
</tr>
<tr>
<td>2</td>
<td>Computer Organization and Architecture</td>
<td>878</td>
<td>319</td>
</tr>
<tr>
<td>3</td>
<td>Database System and Application</td>
<td>1,669</td>
<td>685</td>
</tr>
</tbody>
</table>

Given the number of knowledge units \( n \), the number of all possible pairs \( \# \text{possible pairs} \) that should be checked for preorder relations is \( n \times (n - 1) \). However, in practical classifiers, we only check candidate pairs generated by the algorithm described in Section 5.2.

**Table 3. Number of Candidate Pairs and Test instances**

<table>
<thead>
<tr>
<th>ID</th>
<th>#possible pairs</th>
<th>#candidate pairs</th>
<th>#training samples</th>
<th>#test instances</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>1,078,482</td>
<td>1,039</td>
<td>292</td>
<td>610</td>
</tr>
<tr>
<td>2</td>
<td>770,006</td>
<td>878</td>
<td>213</td>
<td>486</td>
</tr>
<tr>
<td>3</td>
<td>2,783,892</td>
<td>1,669</td>
<td>493</td>
<td>967</td>
</tr>
</tbody>
</table>

As shown in Table 3, the number of candidate pair \( \# \text{candidate pairs} \) is far less than the number of all possible pairs, and is proportional to the number of knowledge units. In experiments, we randomly selected 70% of knowledge units to build the training set and the remainder 30% to build the test set. The \#training samples denotes the number of preorder candidate pairs in training set, in which ‘+’ denotes the number of positive instances, i.e. pairs who have preorder relation, and the same goes for ‘-’. Usually there are fewer positive instances than negative instances in the corpus, so we duplicated some of positive instances and restricted the number of negative instances to balance the training set. In addition, the \#test instances denotes the number of KU-pairs in test set.

Four types of binary classifiers, including NB (Naïve Bayes), DT (Decision Tree), SVM and MLP (Multilayer Perceptron), are employed in the experiments. In the experiments, if the recognized relation label can match the annotated relation label, then we view it as a correct case. We used precision, recall, and \( F_1 \)-score in evaluation of preorder relation recognition results. The experimental results are presented in Table 4.

**Table 4. Recognition Results**

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Criteria</th>
<th>ID = 1</th>
<th>ID = 2</th>
<th>ID = 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>Precision</td>
<td>95.0</td>
<td>98.1</td>
<td>97.4</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>86.8</td>
<td>76.3</td>
<td>85.0</td>
</tr>
<tr>
<td></td>
<td>( F_1 )-score</td>
<td>90.7</td>
<td>85.9</td>
<td>92.8</td>
</tr>
<tr>
<td>DT (C4.5)</td>
<td>Precision</td>
<td>97.3</td>
<td>96.2</td>
<td>96.3</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>87.4</td>
<td>71.6</td>
<td>82.1</td>
</tr>
<tr>
<td></td>
<td>( F_1 )-score</td>
<td>92.1</td>
<td>82.1</td>
<td>88.6</td>
</tr>
<tr>
<td>NB</td>
<td>Precision</td>
<td>93.4</td>
<td>77.8</td>
<td>87.9</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>82.3</td>
<td>36.3</td>
<td>42.7</td>
</tr>
<tr>
<td></td>
<td>( F_1 )-score</td>
<td>87.5</td>
<td>49.5</td>
<td>57.5</td>
</tr>
<tr>
<td>MLP</td>
<td>Precision</td>
<td>99.0</td>
<td>99.0</td>
<td>96.2</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>87.0</td>
<td>72.6</td>
<td>82.9</td>
</tr>
<tr>
<td></td>
<td>( F_1 )-score</td>
<td>92.6</td>
<td>83.8</td>
<td>89.1</td>
</tr>
</tbody>
</table>

It turns out the performance of DT and MLP is almost the same, but NB performs the worst. This seems to be caused by the fact that the independence assumption of features is not satisfied and some information is lost during the feature normalization. As shown in the Tab.4, SVM excels the others especially on small-scale sample sets. Its average precision ranges from 95% to 98%, recall from 76% to 86% and \( F_1 \)-score from 85% to 92%. Besides, its execution time, although slightly more than DT and NB, is far less than that of MLP. So, we recommend SVM binary classifier in the issue.

The experimental results show the proposed method can effectively extract knowledge unit preorder relations from the text. It reduces the quadratic complexity of linking mining to linear complexity and therefore improves the performance of algorithm.

The results of extracted preorder relations were represented in the language of XTM (XML Topic Maps) and visualized using the viewer of our ETMToolkit (Figure 5).
7. CONCLUSION AND FUTURE WORK
Knowledge Unit Relation Mining as a novel and challenging issue is the foundation for knowledge organization and navigation learning. Due to the particulars of knowledge units and their relations, Ontology learning, KAT and RDC can hardly be applied to the issue. To solve the problem, this paper proposes two characteristics of preorder relation, i.e., the locality of preorder relation and the distribution asymmetry of domain terms, and then presents a method to extract the preorder relation from text. The method overcomes the quadratic complexity problem of link mining and therefore improves the performance of the mining algorithm. On the other hand, the method discusses features for KU-pairs and introduces a suitable preorder relation mining algorithm based on classification. Experimental results show the method exhibits satisfactory precisions and recalls.

There are two specific directions for future work. First, in the paper, we only examine the method on the corpus related to computer science. However, investigation on corpus in different domains and languages will be also significant. In addition, we plan to extend the method to other types of relation between knowledge units such as reference and demonstration.

8. ACKNOWLEDGMENTS
The research was supported by the National High-Tech R&D Program of China under Grant No.2008AA01Z131, the National Science Foundation of China under Grant Nos.60825202, 60803079, 60921003, the National Key Technologies R&D Program of China under Grant Nos. 2006BAK11B02, 2006BAJ07B06, the Program for New Century Excellent Talents in University of China under Grant No.NECT-08-0433.

9. REFERENCES