Data Annotation based on Scenario in Chance Discovery Process

Y. Iwase[1] and Y. Takama[1]

[1]Tokyo Metropolitan University

Abstract

The mapping method between the graph generated by KeyGraph and the scenario drawn up by a user is proposed for supporting chance discovery process. Although KeyGraph is widely known as one of the effective tools that support the process of chance discovery, further improvement seems to be required, concerning the ambiguity involved in user's interpretation of the graph. The mapping found by the proposed algorithm is used for extracting the data referred to in the scenario and for annotating those in the original data file. The annotated data files are expected to be used for further data analysis as well as for supporting group discussion. The preliminary experimental result shows how the algorithm works.

Keywords: Data annotation, chance discovery, KeyGraph, information visualization.

1. Introduction

The mapping method between the graph generated by KeyGraph [3] and the scenario, which a user draws up from the graph, is proposed for supporting chance discovery process. The KeyGraph, which visualizes the hidden structure within the data set, is widely known as one of the effective tools for supporting the process of chance discovery. However, further improvement seems to be required for resolving the ambiguity involved in user's interpretation of the graph.

The proposed method is considered as a kind of "smart data" approach [1] in the sense that the original data set used as the input to KeyGraph is annotated based on the mapping between the scenario and the graph. The proposed algorithm extracts the set of bridges and islands that are referred to in the scenario. The extracted bridges and islands are translated into logical expressions, which are used for finding the corresponding records in the original data set. Section 2 discusses the smart data approach for chance discovery process, and the algorithm is proposed in Section 3. The preliminary experimental result shows how the algorithm works in Section 4.

2. Smart Data Approach for Chance Discovery

2.1. Roles of Computer Systems in Chance Discovery Process

Chance discovery process generally involves the interaction between humans and computer systems. Computer systems help humans understand the data collected from a domain of interest, and make decisions for business, research, etc. Typical tasks that are performed by the computer systems are data mining and information visualization. In general, data mining is a task that extracts general rules or abstract information from huge data set in terms of predefined criterion. In other words, typical data mining techniques find the information supported by a large number of data, which is useful for prediction in a stable environment. However, using only data mining techniques is not enough for chance discovery, because its aim is different from prediction based on general rules. That is, the word 'chance' means information about an event or a situation that is significant for making decisions [2], and such event or situation is usually a rare one.

As the saying "a chance favors a prepared mind" indicates, chance discovery is possible if the context information brought by humans meets the information extracted from data sets. That is why information visualization is another important task for computer systems in chance discovery process. Information visualization systems display the collected data with the structure, so that a user can easily grasp the important characteristics of the data set, such as the relationship among objects and trends in the domain of interest. Compared with scientific visualization systems, which handles huge but well-organized data set, their target data set is usually ill-organized and has a variety of potential structure. Therefore, not only visualization itself but also giving the data appropriate structure is important for information visualization. A chance is not common for everyone, but depends on the context in which a person tries to make decisions.

That means information visualization systems will be effective if they can display the data set with the structure that matches the person's context.

2.2. KeyGraph

Concerning the important role of information visualization, a variety of information visualization systems have been designed for chance discovery process [3,4], among which one of the most famous systems is KeyGraph [3]. This section briefly describes the graph generated by KeyGraph.

Fig. 1 shows the example graph generated by KeyGraph, which consists of the following objects.

- **Black nodes** indicate the items frequently occurred in a data set.
- **White nodes** indicate the items not occurred so frequently in a data set.
- **Double-circled nodes** indicate the items that can be considered as keywords.
- **Links** indicate that the connected item pair co-occurs frequently in a data set. A solid line is used for forming an *island*, while a dotted line is used for connecting islands.

There are also composite objects that are very important for grasping what a graph shows.

- **An island** is defined as the connected component of the black nodes with solid lines.
- **A bridge** is defined as the dotted line that connects between islands or nodes.

The islands can be viewed as the underlying common contexts, because they are formed by the set of items co-occurred frequently in the data set. For example, the island in the left part in Fig. 1 refer to daily food. On the other hand, bridges are important in the sense that they connect two common contexts with new context, which is brought by the items that are not frequently occurred. While the common context represented by islands are widely known, the context represented by bridges are not so popular at this moment, which will lead to a chance.

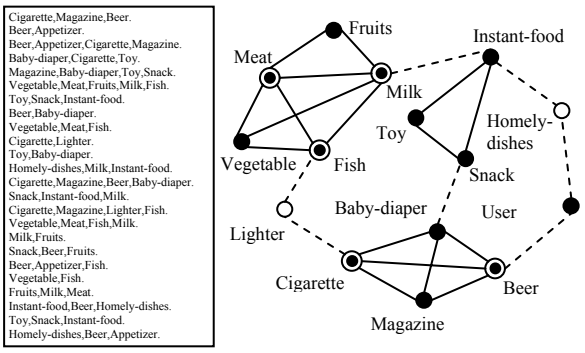

Fig. 1: Example data set and KeyGraph.

2.3. Smart Data: Scenario to Data Mapping

The typical usage of KeyGraph in chance discovery process is that users draw up the scenario from what they can read from the graph. Although KeyGraph is one of the successful information visualization systems designed for chance discovery process, it should be noted that users have possibilities to misread the graph with the following problems.

1. A user tends to estimate the relationships between nodes based on their distance on the graph.
2. Various graphs can be made from a single data set.

As for the former problem, KeyGraph displays the relationship between nodes with links, and the distance between nodes are determined just in terms of arrangement. However, the distance among nodes is often used for showing their similarity in information visualization systems [4], and there is a possibility that a user find a "phantom" island. Although a user who gets familiar with using KeyGraph can hardly make such mistakes, this problem becomes serious when a large number of nodes exist on a graph.

The latter problem is brought by the interactive facility the KeyGraph has. That is, users can generate preferred graphs by adjusting several parameters, such as the number of black nodes, solid links, and bridges. This facility is very useful for the data set to be visualized in the suitable manners for the user's purpose, but different scenarios might be generated from each graph, even if these graphs are generated from the same data set. Although it is preferable in the context of chance discovery that each person generates different scenario from the same data, too much variety of scenarios might cause a problem.

To solve these problems, this paper proposes to find the data referred to in the scenario and annotate those in the original data file. In other words, this approach finds the mapping from scenario to data. Finding the data that are referred to in the scenario lets users validate their scenarios. That is, users can examine whether the island they found is a phantom or not, as noted in the former problem, by looking into the corresponding data. Comparison between the data annotated with a scenario and those annotated with another scenario in the same data file is also helpful for solving the latter problem. Furthermore, the following benefits are also expected.

- Scenario comparison between different users can be possible. A scenario is written in natural language, which makes it difficult to be compared with another one. On the other hand, the comparison on the basis of annotated data can be

performed objectively, which could make the group discussion with KeyGraph more fruitful.

- The annotated data can be extracted and used for generating a new graph, which focuses on the specific part of the data. As the chance discovery process is generally regarded as the double-helical model [2], which involves the repeated interaction between humans and computer systems, the graph generated from specific part of data will help users go into the further level of analysis.

It is interesting that computer processing has been moved from procedure-oriented approach, through object-oriented one, to data-oriented (i.e. application-independent) one just like the Semantic Web [1]. The data-oriented approach is often referred to as the "smart data" approach, which makes data smart. The proposed approach also goes along with the trend.

3. Scenario to Data Mapping Algorithm

3.1. Outline of Mapping Algorithm

This section proposes the algorithm for mapping from a scenario to an original data file. Fig. 2 shows the flow of the algorithm along with chance discovery process. The algorithm consists of 3 steps: The "(1) Graph analysis" step extracts a set of keywords (K_{all}), bridges (B_{all}), and islands (SL_M) from the graph data file, which is the output of KeyGraph. The "(2) Scenario analysis" step analyzes the text in the scenario and obtains a set of bridges (B_I) and islands (SL_I), which are of interest for the user. Finally, in the "(3) Data annotation" step, the obtained bridges and islands are translated into logical expressions, which are used to retrieve the corresponding data and annotate those.

3.2. Graph Analysis

This step extracts the set of keywords, bridges, and islands from the graph with the following procedures.

1. Extract a set of keywords that are appeared on the graph. The keyword k_i has two attributes, a word w_i and the node type t_i.
$$K_{all} = \{k_i(w_i, t_i)\}, t_i \in \{black, white\}. \quad (1)$$
2. Extract a set of links that are appeared on the graph. The link b_i is represented as a set of two keywords k_f and k_t, which are its endpoints.
$$B_{all} = \{b_i\}, b_i = \{k_f, k_t\}, k_f, k_t \in K_{all}. \quad (2)$$

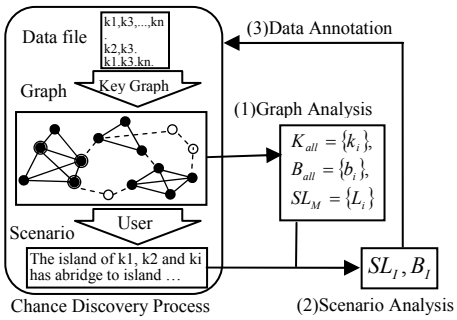

Fig. 2: Outline of mapping algorithm.

3. Obtain a set of islands found on the graph. An island is mathematically defined as a connected component with solid lines, and represented by a set of keywords within the island.
$$SL_M = \{L_i\}, L_i = \{k_j \in K_{all}\}. \quad (3)$$

3.3. Scenario Analysis

This step finally obtains the set of bridges and islands, which are referred to by the user in his / her scenario.

1. Extract the set of keywords appeared in the scenario, $K_S \subseteq K_{all}$.
2. Obtain the set of optional islands (SL_U), which are defined by the user in the scenario. Since one node can belong to only one island, exclude the island which overlap with SL_M .
for $^{\forall} L_i \in SL_U$,
if $^{\exists} L_j \in SL_M, L_i \cap L_j \neq \phi$, then $SL_M \rightarrow SL_M - \{L_j\}$. (4)
3. Obtain the set of the islands which is referred to in the scenario, $SL_S \subseteq SL_M$. It is assumed that an island L_j is referred to, if one or more keywords in the island co-occur with the term denoting island, such as "island", "cluster" and "group", in at least one sentence in the scenario. The set of islands SL_I , which are of interest for the user, can be obtained by extending the referred island set with user-defined island set.
$$SL_I = SL_S \cup SL_U \quad (5)$$
4. Obtain the set of the nodes (K_A) which are valid in the scenario. K_A can be obtained by extending K_S with the nodes which belongs to SL_I .
$$K_A = K_s \cup \{k_i |^{\exists} L_j \in SL_I, s.t. k_i \in L_j\}. \quad (6)$$
5. Obtain the set of the bridges (B_A), which are valid in the scenario. B_A can be obtained by excluding the bridges inside SL_I from ones which exist among K_A .
$$B_A = \{b_n \mid b_n \subset K_A\} - \{b_m |^{\exists} L_j \in SL_I, b_m \subset L_i\}. \quad (7)$$
6. Obtain the set of the bridges (B_I), which are of interest for the user, by applying the following steps to each sentence in the scenario.

(i) If the sentence refers to only one node (k_I) or one island (L_k), which co-occurs with the term denoting bridge, such as "bridge", "link" and "connection", then add the set of bridges which meet the following requirements to B_I .

$$\forall b_n \in B_A, k_i \in b_n \rightarrow b_n \in B_I. \quad \text{(node)}$$
$$\forall b_n \in B_A, b_n \cap L_k \neq \phi \rightarrow b_n \in B_I. \quad \text{(island)}$$

(ii) Otherwise, add the set of bridges which meet the following requirements to B_I . K_I indicates the set of keywords which occur in the *i*th sentence (i.e. the sentence in process).

$$\forall k_j \in K_i, \{b_n \mid b_n \in B_A, k_j \in b_n, b_n \subseteq K_i\} \subseteq B_I.$$

7. When a sentence refers to the pair of islands (L_m, L_n) , which has no bridges extracted in step 6-(ii), add all the bridges existing between those islands to B_I .

$$\forall b_i \in B_A, (b_i \cap L_m \neq \phi) \wedge (b_i \cap L_n \neq \phi) \rightarrow b_i \in B_I. \quad (8)$$

8. As a white node often plays an important role as a relay of bridges, extend the set of bridges B_I in the following way.

for $\forall b_i \in B_I$, if $\exists k_j \in b_i$, s.t. $t_j = white$,
then $\forall b_j \in B_A, k_j \in b_j \rightarrow b_j \in B_I$.

3.4. Data Annotation

Finally, the obtained bridges (B_I) and islands (SL_I) are translated into the logical expressions, by which the records to be annotated are extracted from the original data set. For $L_i \in SL_I$, the record containing 2 or more keywords that belong to L_i is extracted and annotated. For $b_i \in B_I$, the record containing both keywords (endpoints) of b_i is extracted and annotated. Whether a record corresponds to a bridge or an island is identified with attribute value of the tag.

4. Experiments

This section shows how the proposed algorithm works on the example data set given in Fig. 1. By graph analysis step (Sec. 3.2), the following sets are obtained.

- $K_{all} = \{k_0$(cigarette,*b*),k_1(magazine,*b*),k_2(beer,*b*), k_3(appetizer,*b*), k_4(baby-diaper,*b*),k_5(toy,*b*),k_6(snack,*b*),k_7(vegetable,*b*),k_8(meat,*b*), k_9(fruits,*b*),k_{10}(milk,*b*),k_{11}(fish,*b*),k_{12}(instant-food,*b*),k_{13}(lighter,*w*), k_{14}(homely-dishes,*w*)$\}$,
- $B_{all} = \{b_0, b_1, \ldots, b_{23}\}$. $b_0 = \{k_0, k_1\}, \ldots$,
- $SL_M = \{L_0 = \{k_0, k_1, k_2, k_4\}, L_1 = \{k_5, k_6, k_{12}\}, L_2 = \{k_7, k_8, k_9, k_{10}, k_{11}\}\}$.

Let us consider the following scenario that is given for the graph.

"Beer, cigarette, and magazine, which a man frequently purchases, form an *island* together with baby-diaper that he might be asked to buy because of its weight. When he purchases a baby-diaper, he will remind of his baby, and purchase toys and snacks as well. Alternatively, he might purchase instant-food or snacks, together with appetizer and homely-dishes, with which he enjoys beer."

From this scenario, the following sets are extracted as a result of scenario analysis step (Sec. 3.3).

- $K_S = \{k_0, k_1, k_2, k_3, k_4, k_5, k_6, k_{12}, k_{14}\}$, $SL_U = \phi$, $SL_I = \{L_0\}$.
- $K_A = \{k_2, k_4, k_5, k_6, k_{12}, k_{14}\}$. $B_A = B_I = \{\{k_2, k_3\}, \{k_3, k_4\}, \{k_4, k_5\}, \{k_5, k_6\}, \{k_3, k_{14}\}, \{k_{12}, k_{14}\}, \{k_6, k_{12}\}\}$.

Finally, the data set is annotated as shown in Fig. 3.

```
<s01 author="A" type="island">Cigarette,Magazine,Beer. </s01>
<s01 author="A" type="bridge">Beer,Appetizer.</s01>
<s01 author="A" type="island, bridge">Beer,Appetizer,Cigarette,Magazine. </s01>
<s01 author="A" type="island, bridge">Baby-diaper,Cigarette,Toy. </s01>
<s01 author="A" type="island, bridge">Magazine,Baby-diaper,Toy,Snack. </s01>
Vegetable,Meat,Fruits,Milk,Fish.
<s01 author="A" type="bridge">Toy,Snack,Instant-food. </s01>
<s01 author="A" type="island">Beer,Baby-diaper. </s01>
Vegetable,Meat,Fish.
Cigarette,Lighter.
<s01 author="A" type="bridge">Toy,Baby-diaper. </s01>
<s01 author="A" type="bridge">Homely-dishes,Milk,Instant-food. </s01>
<s01 author="A" type="island">Cigarette,Magazine,Beer,Baby-diaper. </s01>
<s01 author="A" type="bridge">Snack,Instant-food,Milk. </s01>
<s01 author="A" type="island">Cigarette,Magazine,Lighter,Fish. </s01>
Vegetable,Meat,Fish,Milk.
Milk,Fruits.
Snack,Beer,Fruits.
<s01 author="A" type="bridge">Beer,Appetizer,Fish. </s01>
Vegetable,Fish.
Fruits,Milk,Meat.
<s01 author="A" type="bridge">Instant-food,Beer,Homely-dishes. </s01>
<s01 author="A" type="bridge">Toy,Snack,Instant-food. </s01>
<s01 author="A" type="bridge">Homely-dishes,Beer,Appetizer. </s01>
```

Fig. 3: Annotation results for data file in Fig. 1.

5. Conclusions

The mapping method between a graph generated by KeyGraph and a scenario is proposed for supporting chance discovery process. The mapping found by the algorithm is translated into logical expressions, which are used for extracting the data referred to in the scenario and for annotating those in the original data file. The preliminary experimental result shows how the algorithm works. Future works include its evaluation with test subjects.

6. References

[1] Daconta M. C, Obrst L. J., and Smith K. T. (2003). "1. What is the Semantic Web?," *The Semantic Web*, John Wiley & Sons.

[2] Ohsawa Y. (2003). "1. Modeling the Process of Chance Discovery," in Ohsawa Y. and McBurney P. Eds., *Chance Discovery*, pp. 2-15.

[3] Ohsawa Y. (2003). "18. KeyGraph: Visualized Structure Among Event Clusters," in Ohsawa Y. and McBurney P. Eds., *Chance Discovery*, pp. 262-275.

[4] Takama Y. and Kajinami T. (Aug. 2004). "Keyword Pair Extraction for Relevance Feedback based on Interactive Keyword Map," *1st European Workshop on Chance Discovery in ECAI2004,* pp. 41-50.