International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:5, 2011

Analyzing Methods of the Relation between Concepts based on a Concept Hierarchy

Ke Lu and Tetsuya Furukawa

Abstract**—**Data objects are usually organized hierarchically, and the relations between them are analyzed based on a corresponding concept hierarchy. The relation between data objects, for example how similar they are, are usually analyzed based on the conceptual distance in the hierarchy. If a node is an ancestor of another node, it is enough to analyze how close they are by calculating the distance vertically. However, if there is not such relation between two nodes, the vertical distance cannot express their relation explicitly. This paper tries to fill this gap by improving the analysis method for data objects based on hierarchy. The contributions of this paper include: (1) proposing an improved method to evaluate the vertical distance between concepts; (2) defining the concept horizontal distance and a method to calculate the horizontal distance; and (3) discussing the methods to confine a range by the horizontal distance and the vertical distance, and evaluating the relation between concepts.

Keywords**—**Concept Hierarchy, Horizontal Distance, Relation Analysis, Vertical Distance

I. Introduction

RELATION analysis between data objects are used in various fields, such as classifying [1], clustering [8], [13] and text retrieval [10]. Many relation analysis methods are based on distance metric, e.g. Euclidean distance, cosine distance, in flat partition. Compared with treating data objects flatly, it is a common tendency to analyze the relations between data objects hierarchically [1], [11].

In a hierarchy, nodes are connected by edges, and both the intermediated nodes and leaf nodes have a label used to indicate a category for itself. Data objects are usually assumed to be classified into one category and annotated with the label of that category [14]. Assuming data objects can be assigned into the intermediated levels, the concept distance is used to analyze the relations between those data objects.

If data objects are classified into the same category, they are related to each other closely. Even if data objects are classified into different categories, the relations between them, for example how close they are, can also be measured by calculating the distance between them based on hierarchy. Counting the number of edges in the shortest way between any two nodes is a simple way to calculate the distance [8], however some detail information, for example which node has a higher level than the other one, would be ignored.

For two nodes, if a node is an ancestor of the other node, it is said that there is ancestor-descendent (*AD* for short) relation between them. Some methods are proposed to evaluate the distance between data objects which have *AD* relation between them, i.e. consider the distance between data objects vertically [8]. To some extent, vertical distance is enough to evaluate those relations if there is *AD* relation between the labels of those data objects. Document is usually treated as a concrete example of data object. If a document covers a topic or a certain part of a topic, the document should be classified into that topic class or a subclass of that topic. Classification becomes complicated when a document covers many topics, furthermore these topics are on different granularity levels of the hierarchy and there may not be *AD* relations between some of those topics. As another example, there are three data objects, which are labeled with *China*, *Japan* and *France*, respectively. If a query asks for a data object which is close to the data object *Japan*, which one is a better answer for it? These three data objects are all on the country level based on region concept hierarchy, and there are not *AD* relations between them. Although the vertical distances between each two of them are the same, this paper proposes the notion of horizontal distance to argue that *China* is actually a better answer for this query.

An improved method, which analyzes relations through both horizontal and vertical direction in a hierarchy, is proposed in this paper. Horizontal distance is used to evaluate the relation between data objects which do not have *AD* relation. For simplicity, this paper only discusses the situation that both the query and the data are described by a single label in the same hierarchy, and assumes that the concept hierarchy is given in advance.

This paper is organized as follows. Related works and background knowledge are summarized in Section 2. Section 3 discusses the relation between the labels which have *AD* relation, and proposes an improved vertical distance calculating method. In Section 4, the relation between the labels which have no *AD* relation is discussed, and an improved horizontal distance calculation method is also proposed. More details of the newly proposed methods and some examples to use these methods are discussed in Section 5. Section 6 concludes this paper.

II. Related Work And Discussion

Two kinds of literatures give much inspiration for this paper: (1) the way how to organize and describe data objects, and (2) the way how to evaluate the relations between data objects. Concepts are the characteristics generated for objects by

K. Lu is with the Department of Economic Engineering, Kyushu University, Hakozaki 6-19-1, Higashi-ku, Fukuoka 812-8581 Japan (e-mail: looker@en.kyushu-u.ac.jp).

T. Furukawa is with the Department of Economic Engineering Department, Kyushu University, Hakozaki 6-19-1, Higashi-ku, Fukuoka 812-8581 Japan (e-mail:furukawa@en.kyushu-u.ac.jp).

various means such as manual, statistical, rule-based methodologies. Labels, sometimes in form of keywords, are usually used to capture the concepts of data objects. Although keywords and concepts are usually distinguished [12], this paper treats key words (or labels) as the concrete expression of concepts. There are some relations between concepts, such as hyponym and hypernym relation (called *AD* relation in this paper), so that concepts can be organized in some certain structures, such as concept hierarchy.

Objects are usually classified into one category [3]. Based on a hierarchy, data can be classified into one category or some categories. Some literatures focusing on classifying data hierarchically assume that data is classified into one terminal class [2]. Unlike the former methods, [9] proposed a top-down level-based classification method that can classify documents to both leaf and internal categories. In other words, data can be classified into various levels based on a hierarchy. Hierarchical classification has been used to many fields, such as aid to database and hit-list browsing [6], web content [7]. With the rise of web 2.0, more attentions are paid to the data labeled with multiple-labels. Some special orders were proposed in [11] for multi-labeled data expressed by a set of labels and [1] analyze multi-labeled data based on the roll of a concept against a semantic range with these orders. The literatures of this direction are usually related to some data mining algorithms such as clustering and classification, [8] as an example.

This paper derives much inspiration from those researches, but the works that give us the most useful inspirations are [8] and [10]. A new context-based semantic distance measure method is proposed in [10] for textual data. They both tried to classified data objects based on the distance in a hierarchy.

Most of the former literatures calculate the distance between two labels in a hierarchy by simply calculating edges. The distance between two labels is the total number of the edges of the shortest path between them. That is an easy way to calculate the distance between these two labels, but there is an obvious disadvantage. Given a label *D*, finding a label whose distance from *D* is three, it would get a descendant which is 3 levels lower than *D* or an ancestor which is 3 levels higher than *D*, and even a label two levels higher than *D* without the *AD* relation. It cannot be confirmed that whether the level of the resulted label is higher or lower than a given label cannot be confirmed.

The former literatures pay no attention to two details. The one is that the distance is only considered as scalar. If the distance is simply considered as a scalar without direction, the subordination between the concepts would be ignored when the distance is calculated. The other one is that only the vertical distance is considered without considering the distance horizontally. This paper differs from the former literatures in that the vertical distance is considered as a vector, and both the vertical distance and the horizontal distance are considered.

III. Vertical Distance Between Labels

Data objects are usually organized based on a hierarchy. A hierarchy *H* of a node set is a tree whose root is a special node, which does not have parent nodes, and if a node has child nodes then each node form a partition of their parent. Every node has a label indicating a category. If a data object is classified into a category, it would be expressed with the label of corresponding node. The level of a label *L* is denoted by *level(L)*, which is also used to indicate the level of a data object conceptually. *level(L)* is usually treated as a positive integer and the integer of higher level is smaller than that of the lower level.

For labels L_1 and L_2 in a concept hierarchy, if the level of L_1 is higher than the level of L_2, their relation can be denoted by $level(L_1)<level(L_2)$. If there is an upward-only or a downward-only path between L_1 to L_2, it is said that there is ancestor-descendant (*AD* for short) relation between L_1 and L_2. The set of objects whose labels have *AD* relation with a label *L* is denoted by $\overline{L}^{AD}$. If $level(L_1)<level(L_2)$ and there is *AD* relation between them, it is said that L_1 is an ancestor of L_2 and L_2 is a descendent of L_1, denoted by $L_2 \prec L_1$. In this section, only the labels with *AD* relation are discussed.

Let a query *Q* be expressed by a label *L*, and $\tilde{o}$ be the label of an object *o* in the concept hierarchy. Finding answers for queries then can be regarded as matching some objects under various constraints. For $\overline{L}^{AD}$, there are basically three subtypes of objects for a query as following.

1) The simplest situation is that the objects labeled with *L* are the answer for query *Q*, which is called exact-objects, denoted by $\overline{L}^{E}$. For example, in a sports hierarchy, if *court game* is given as a query, all of the objects labeled with *court game* should be the answer for this query.
2) Suppose the label *L* is the root of a subtree in a hierarchy. The answer is the objects labeled with any labels in the sub tree $\{o|\ \tilde{o} \prec L\}$. This kind of objects is called descendant-objects, denoted by $\overline{L}^{D}$. For example, in a hierarchy of sports, if *court game* is given as a query, all of the objects labeled with *court game*, even labeled with *basketball* or *football* (*basketball* ≺ *court game* and *football* ≺ *court game*) are included into the answer for this query.
3) Another type of answer is that objects labeled with the higher labels, such as *athletic game* over *court game*, denoted by $\{o|\ L \prec \tilde{o}\}$, which are called ancestor-objects of *L*, denoted by $\overline{L}^{A}$. $\overline{L}^{A}$ is also a useful answer. For example, getting the economic position of the city Fukuoka in the range of Japan needs the economic information of the country level. The answer of this query can be expressed as $\{o|\ Fukuoka \prec \tilde{o}\}$ which may at least conclude the economic information of Japan.

For the labels with *AD* relation, these three types of objects can satisfy most of queries. Besides that, there are still some advanced queries. There are some cases about the objects related to some certain levels labels. For example, the query, which needs the objects of the *court game* level in the subtree rooted by *athletic game*, is an advanced query for those three types of objects. When the interest is extended from the level of *L* to different levels upward or downward, it is spontaneous to consider the distance between the given label *L* and the labels of objects. For solving this problem, it is necessary to find a method to calculate the difference of levels, and fix a range vertically to find the answer for query.

The level of label is a useful characteristic for calculating distance. Different from the former researches, in this paper, the vertical distance is defined as a vector as follows.

Definition 1 For labels L_1 and L_2, the vertical distance from L_1 to L_2 is $level(L_2) - level(L_1)$, denoted by $d_V(L_1, L_2)$. □

$d_V(L_1, L_2)$ is used to evaluate how far L_1 is from L_2. The larger the absolute value of vertical distance is, the farther these two labels are. $d_V(L_1, L_2)$ also shows the direction of the vertical distance. If $d_V(L_1, L_2)>0$, the level of L_1 is higher than L_2. If $d_V(L_1, L_2)<0$, the level of L_1 is lower than L_2. If $d_V(L_1, L_2)<0$ and there is *AD* relation between L_1 and L_2, L_2 is a descendent of L_1.

Given a query expressed by a label L, the answer for the query may be on certain levels in a hierarchy. Based on the vertical distance, a level range to find the objects related to the given label L can be confined.

Definition 2 For a label L, and integers v_H and v_L ($v_L \leq v_H$), the objects confined by v_H and v_L vertically is $\{o |\ v_L \leq\ d_V(\tilde{o}, \mathrm{L}) \leq\ v_H\}$, denoted by $V(L, v_H, v_L)$. □

In this section, the function $V(L, v_H, v_L)$ is only used to constrain the vertical distance between the objects which have *AD* relation. The parameters v_H and v_L are used to limit the highest and the lowest level boundaries, respectively. The value of v_H is the level of L minus the highest level, and the value of v_L is the level of L minus the lowest level. When $0 \leq\ v_L\ \leq\ v_H$, the labels of objects are all higher than L; when $v_L \leq v_H \leq 0$, the labels of objects are all lower than L; and when $v_L \leq 0 \leq v_H$, the labels of objects are either higher or lower than L.

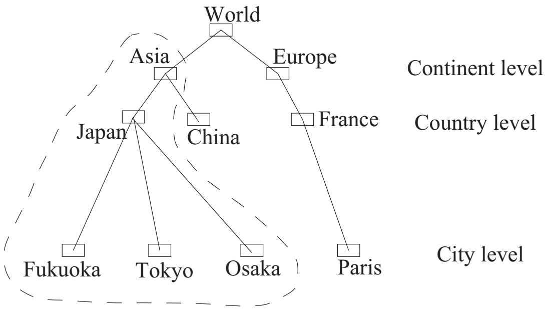

Fig. 1 A simplified region hierarchy

Example 1 In a region hierarchy shown as Fig.1, it is supposed that L=*Japan*, which is on the country level, is given as the query label. When $0\ \leq\ v_L\ \leq\ v_H$, $V(L, v_H, v_L)$ is the set of the objects whose levels are higher than the country level. The objects labeled with *Asia*, which is on the continent level, are included in the answer for *Japan*. When $v_L\ \leq\ v_H\ \leq\ 0$, $V(L, v_H, v_L)$ is the set of objects whose levels are lower than the country level, such as *Fukuoka* on the city level, may be the label of objects in the answer. And when $v_L \leq 0 \leq\ v_H$, such as $v_L = -1$ and v_H =1, $V(L, v_H, v_L)$ is the set of objects shown as the dashed line range in Fig. 1. □

Actually, these three types of objects $\overline{L}^A$, $\overline{L}^D$ and $\overline{L}^E$ can be expressed by combining $\overline{L}^{AD}$ and $V(L, v_H, v_L)$.

$\overline{L}^E \cap V(L, v_H, v_L) = \overline{L}^E \cap V(L, 0, 0)$, which is the same as $\overline{L}^{AD} \cap V(L, 0, 0)$. In the same way, $\overline{L}^D \cap V(L, v_H, v_L)$ is equal to $\overline{L}^{AD} \cap V(L, v_H, 0)$, and $\overline{L}^A \cap V(L, v_H, v_L)$ is equal to $\overline{L}^{AD} \cap V(L, v_H, 0)$. In a word, the answers for queries about $\overline{L}^{AD}$ can be described by $\overline{L}^{AD} \cap V(L, v_H, v_L)$.

IV. Horizontal Distance Between Labels

For two labels L_1 and L_2 which have *AD* relation, the vertical distance can be used to evaluate how close they are to each other. However, for the labels which have no *AD* relation, such as the sibling labels, the vertical distance has less sense because the vertical distance between them is 0. In this section, the notion horizontal distance is proposed to measure the relation between labels which have no *AD* relation, and a method used to evaluate the horizontal distance is proposed.

The vertical distance between the concepts, which are on the same level, is always 0 as the definition of vertical distance. Since *Fukuoka*, *Tokyo* and *Paris* shown as Fig. 1 are all on the city level, the vertical distance between each two of them is 0. However, the relations between them are different. *Tokyo* is the sibling concept of *Fukuoka* and their parent concept is *Japan*, while *Fukuoka* and *Paris* do not have the common parent concept. It seems that *Tokyo* is more proximate than *Paris* to *Fukuoka*. Under this condition, the vertical distance has no use to evaluate the relations between them. Sometimes the objects whose labels have no *AD* relation with the query label are also needed. For example, there is a case to compare the economic situation between Fukuoka and other conceptually near cities, such as cities of the other countries in Asia. The labels of such cities do not have *AD* relation with *Fukuoka*.

Only under constraints of vertical distance, it is impossible to locate such objects, because there is not *AD* relation between *Fukuoka* and *China*. In addition, vertical distance cannot compare the proximity relation between labels. For example, it is unable to judge the objects labeled with *France* and *China*, respectively, which is more proximate to the query *Japan* because *France*, *China* and *Japan* are all at the country level and each two of them have no *AD* relation.

The horizontal distance is introduced to evaluate how proximate these labels are. The horizontal distance calculation is based on Lowest Common Ancestor (*LCA* for short). There is a *LCA* label for the any labels which have no *AD* relation. To some extent, the distance between the labels and their *LCA* reflects the relation between labels.

Definition 3 For labels L_1 and L_2, the label L is the Lowest Common Ancestor of L_1 and L_2, denoted by LCA (L_1, L_2), if

$L_1 \preccurlyeq L$, $L_2 \preccurlyeq L$ and $\nexists$ L' *s.t.* $L_1 \preccurlyeq L'$, $L_2 \preccurlyeq L'$ and $L' \prec L$.

□

Any two nodes have only one *LCA* node, and in an extreme situation, the root is the *LCA* for two nodes. If L_1 and L_2 have *AD* relation, the *LCA* of them is either L_1 or L_2. Inspired by the former researches, the horizontal distance proposed in this paper is determined by *LCA*.

Definition 4 For labels L_1 and L_2, the horizontal distance from L_1 to L_2 is denoted by $d_H(L_1, L_2)$. If there is no *AD* relation between L_1 and L_2, $d_H(L_1, L_2)=d_V(LCA(L_1, L_2), L_1)$, and 0 otherwise. □

Here, the horizontal distance is one-sided distance, so that $d_H(L_1, L_2)$ is not necessarily equal to $d_H(L_1, L_2)$. In the definition of d_H, there are two parameters. The first parameter is treated as the standard to evaluate the horizontal distance. In Fig. 1, $d_H(Japan, Paris)$ is 3 while $d_H(Paris , Japan)$ is 2.

For two labels, the smaller the horizontal distance is, the more proximate from the basic label to other one is. Even on the same level in a hierarchy, the proximity between them is different as shown in the following example.

Example 2 In the region hierarchy shown as Fig. 1, suppose a label *L*, such as *Japan*, is on the level of country. Label *France* and label *China* are at the same level as *Japan*, the country level. Objects labeled with *China* is more related to Japan because the horizontal distance $d_H(Japan, China) < d_H(Japan, France)$. □

Let a positive integer *h* constrain the horizontal distance from label *L* to other labels. *h* fixes the *h* levels higher ancestor label than *L*, which is the farthest ancestor of *L* within *h*. There is a subtree whose root is the farthest ancestor. The label $\tilde{o}$ in the subtree may have different *LCA* with *L*, while $d_H(L, \tilde{o}) \leq h$ for an object *o* whose label is in the subtree. Intuitional explain is that Japan and China are both in Asia, while France is in Europe.

Definition 5 For a label *L* and an integer $h \geq 0$, the objects labeled with ancestor-shared labels of *L* confined by *h* is $\{o|\ d_H(L, \tilde{o}) \leq h,\}$, denoted by *H(L,h)*. □

When *h=0*, *H(L,0)* is actually $\overline{L}^{AD}$. *H(L,h)* confines a certain range for the labels without *AD* relation. The range is determined by the proximity between *L* and other labels.

Example 3 In a region hierarchy shown as Fig. 1, suppose a label *L*, such as *Japan*, is on the level of country. When *h*=1, a fixed ancestor can be found on the continent level *Asia*. From *L* to the labels in the subtree whose root is *Asia* the horizontal distance is 1. □

V. The Objects Related To A Given Label

Based on vertical distance and horizontal distance introduced in Sections 3 and 4, the hierarchy can be treated as an abstract two-dimension coordinate. These two constraints can be combined in a function defined as follows.

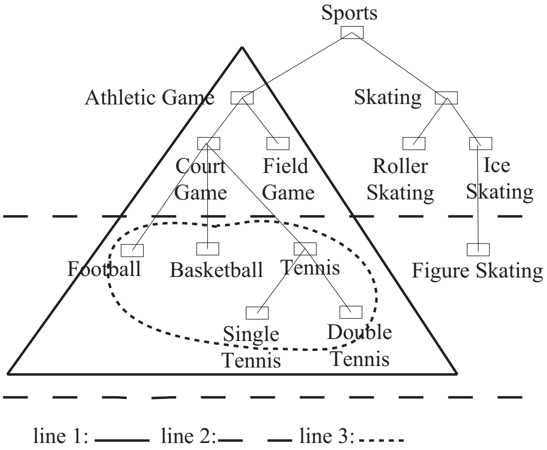

Fig. 2 The sports hierarchy where $h \geq v_H \geq v_L$

Definition 6 For an label *L* and integers v_H, v_L and *h*, where $h \geq 0$ and $v_L \leq v_H$, the labels which are restricted by vertical and horizontal distance is

$R(L, v_H, v_L, h) = \{o|\ v_L \leq d_V(L, \tilde{o}) \leq v_H,\ d_H(L, \tilde{o}) \leq h\}$ □

Actually, the range confined by $R(L, v_H, v_L, h)$ is the intersection of the range confined by vertical distance and that confined by horizontal distance. There are three kinds of situations about the labels related to *L* according to the relation between v_H, v_L and *h*.

Fig. 2 is a simplified concept hierarchy based on wordnet 2.1 database. Based on this hierarchy, given a label *Court Game* to express the query, these three kinds of situations can be explained as follow:

1) $h \geq v_H \geq v_L$

The most common answer for the query is the intersection of the subtree, whose root is the *h* levels higher ancestor of *L*, and the range confined by both v_H and v_L. Suppose *h*=1, $v_H = -1$ and $v_L = -2$, shown as Fig. 2, the range confined by the horizontal distance is marked by line 1, the range confined by the vertical distance is marked by line 2, so that the final result confined by $R(L, v_H, v_L, h)$ is marked by line 3. The objects labeled with *Football*, *Basketball*, *Tennis* and their descendant labels should be retrieved for query *L*. These retrieved labels are all in a subset of the subtree whose root is 1 level higher than *Court Game*, here is *Athletic Game*.

2) $v_H \geq h \geq v_L$

Sometimes, the parameter v_H is given higher than *h*. Under this situation, the highest level boundary is higher than the level of the subtree whose root is fixed by *h*, the parameter

v_H has no use to confine the range for the final retrieved result, $R(L, v_H, v_L, h) = R(L, h, v_L, h)$. Suppose h=1, v_H =2 and $v_L = -1$, the retrieved labels are in the range marked by dotted line shown as Fig. 3.

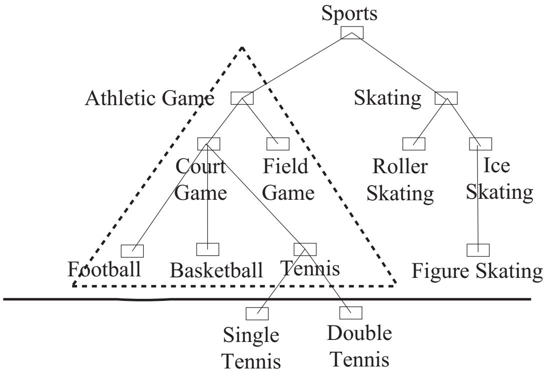

Fig. 3 The sports hierarchy where $v_H \ge h \ge v_L$

3) $v_H \ge v_L \ge h$

Sometimes, the parameters are given in a wrong way, the level of the root to confine the range horizontal distance is lower than the level decided by v_L. There is no intersection between the vertical confine and horizontal confine. Under this condition, there are not labels to be retrieved.

Actually when v_H, v_L and h are all equal to *0*, $R(L, v_H, v_L, h) = \overline{L}^E$.

Definition 6 confines the range by combing vertical distance and horizontal distance in a simple way, which just uses them separately without considering the internal relation between them. Furthermore, the vertical distance and horizontal distance are given the equal weight. Definition 6 cannot satisfy such query that prefer flat range, which need to consider the objects on the ancestor and descendant levels meanwhile prefer the labels on the same level.

If the vertical distance and the horizontal distance are combined as a whole, the sum of the vertical distance and horizontal distance is confined within a range. In addition, if the distances are treated as parameters in linear functions, the weights of vertical distance and horizontal distance can also be considered. Definition 7 proposes an advanced way to combine these two distances.

Definition 7 For a label L, a positive integer m, and linear functions f_1 of $d_V(L, \tilde{o})$ and f_2 of $d_H(L, \tilde{o})$, the objects whose sum of vertical distance and horizontal distance to L are restricted by m is $\{o \mid f_1(d_V(L, \tilde{o})) + f_2(d_H(L, \tilde{o})) \le m\}$, denoted by $F(L, m)$. □

Different linear functions make rang different. When the linear functions are fixed, the label L and positive integer m would also affect the final retrieved labels.

Example 3 In the sports hierarchy, suppose the label *Cout Game* is given as query. If the horizontal distance is preferentially considered, f_1 is given as $2 \times d_V(L, \tilde{o})$ and f_2 is transformed to $d_H(L, \tilde{o})$, $F(L, m) = \{o \mid 2 \times d_V(L, \tilde{o}) + d_H(L, \tilde{o}) \le m\}$. When m=3, the labels retrieved for query L are included in the range marked by the dashed line shown as Fig. 4. □

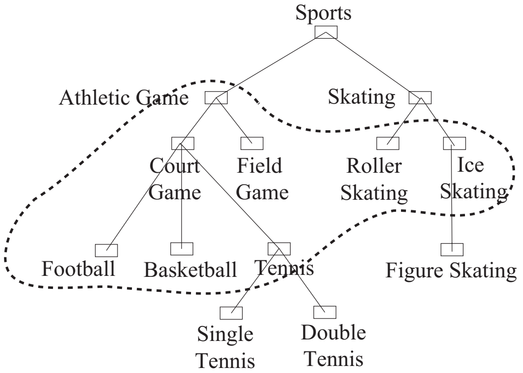

Fig. 4 A *F(L, m)* range in the sports hierarchy

VI. Conclusion

Based on the concept hierarchy, the relations between the objects labeled with the concept labels have been discussed. If the query is treated as label, finding some related data objects is equal to find some related labels for the query label under some constraints. The notion of distance is used to analyze the relations between various labels. Different from the former researches, this paper uses not only the vertical distance but also the horizontal distance to evaluate how close between different labels. Furthermore, this paper treats the vertical distance as a vector to distinguish the ancestor and descendant.

For simplicity, this paper only discusses the situation that both the query and the single-labeled data objects in the same hierarchy, and assumes that the concept hierarchy is given in advance. The future work includes the similar analysis under the multiple-labels condition.

References

[1] M. Kuzunishi, T. Furukawa, and K. Lu, “Analyzing Multi-Labeled Data Based on the Roll of a Concept against a Semantic Range,” in *Proc. of the Int'l Conf. on World Academy of Sciences, Engineering and Technology*, Singapore, 2010, pp. 498–504.

[2] D. Koller, and M. Sahami, “Hierarchically Classifying Documents Using Very Few Words,” in *Proc. of the Fourteenth Int'l Conf. on Machine Learning*, 1997, pp.170–178.

[3] S. Amit, “Modern information retrieval: a brief overview,” *IEEE Data Eng. Bull.*, vol. 24, Dec. 2001, pp. 35–43.

[4] T. Li, S. Zhu, and M. Ogihara, “Topic hierarchy generation via linear discriminant projection,” in *Proc. of the 26th annual international ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2003, pp. 421–422.

[5] Y. Wang, and Z. Gong, “Hierarchical Classification of Web Pages Using Support Vector Machine,” in *Proc. of the 11th Int'l Conf. on Asian Digital Libraries*, 2008, pp. 12–32.

[6] J. R. Rose, and J. Gasteiger, “Hierarchical classification as an aid to database and hit-list browsing,” in *Proc. of the third Int'l Conf. on Information and Knowledge Management*, 1994, pp. 408–414.

[7] S. Dumais, and H. Chen, “Hierarchical classification of Web content,” in *Proc. of the 23rd annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2000, pp. 256–263.

[8] C. C. Hsu, and Y. P. Huang, “Incremental clustering of mixed data based on distance hierarchy,” *Expert Syst. Appl.* Vol.35, 2008, pp. 1177–1185.

[9] A. X. Sun, and E. P. Lim, “Hierarchical Text Classification and Evaluation,” in *Proc. of the 2001 IEEE Int'l Conf. on Data Mining*, 2001, pp. 521–528.

[10] A. El Sayed, H. Hacid, and D. Zighed, “Using semantic distance in a content-based heterogeneous information retrieval system,” in *Proc. of the 3rd ECML/PKDD Int'l Conf. on Mining Complex Data*, 2008, pp. 224–237.

[11] M. Kuzunishi, and T. Furukawa, “Representation for multiple classified data,” in *Proc. of the 24th IASTED Int'l Conf. on Database and Applications*, 2006, pp.135–142.

[12] B. Catherine, and P. Wanda, “Better Rules, Few Features: A Semantic Approach to Selecting Features from Text,” in Proc. of the 2001 IEEE Int'l Conf. on Data Mining, 2001, pp. 59–66.

[13] K. Bade, and A. Nürnberger, “Creating a Cluster Hierarchy under Constraints of a Partially Known Hierarchy,” in *Proc. of the 2008 SIAM Int'l Conf. on Data Mining*, 2008, pp. 13–24.

[14] A. M. Funes, C. Ferri, J. Hernández-Orallo, and M. J. Ramírez-Quintana, “Hierarchical Distance-Based Conceptual Clustering,” in *Proc. of the Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2008, pp. 349–364.

[15] K. Toutanova, F. Chen, K. Popat, and T. Hofmann, “Text Classification in a Hierarchical Mixture Model for Small Training Sets,” in *Proc. of Int'l Conf. on Information and Knowledge Management*, 2001, pp. 105–112.