The Dynamic Faceted Navigation in Decision Making
Using the Semantic Web Technology

Hak-Jin Kim\textsuperscript{a}, Yongjun Zhu\textsuperscript{b}, Wooju Kim\textsuperscript{c,*}, Taimao Sun\textsuperscript{c}

\textsuperscript{a}School of Business, Yonsei University, 50 Yonsei-ro, Seodaemun-gu, Seoul 120-749, Korea
\textsuperscript{b}College of Computing and Informatics, Drexel University, 3141 Chestnut Street, PA 19104, U.S.A.
\textsuperscript{c}Dept. of Information and Industrial Engineering, Yonsei University, 50 Yonsei-ro, Seodaemun-gu, Seoul 120-749, Korea

Abstract

Categorization in the decision making classifies decision makers’ experiences about the world and provides a guide to reach a goal. This implies that dynamically providing categories reflecting the given decision context gives a great enhancement in decision quality. This study discusses the dynamic category selection under the Semantic Web environment, focusing on an implementation of a decision support system, the dynamic facet navigation system working with an ontology. Predefined fixed categories are provided to refine search results to evade use of complex queries and tedious review of search results, but they often output insensible information because of never reflecting the difference in search results. This paper proposes a dynamic category selection mechanism by using the total gain ratio under a given ontology, and an reordering scheme for resulted categories. It proves the validity of the proposed approach with a statistical analysis lastly.

Keywords: facet navigation, semantic search, information gain, decision making
1. Introduction

Consumers face decision making in their everyday lives to find their products of interest such as when shopping in the supermarket and choosing their magazines from a variety of periodicals. Nowadays their decision making has gotten more difficult because of the product specificity and the consumer preference variation from mass customization in modern production. This burden of comparing the vast number of different products to find the proper demands to decision makers good strategies that eases the difficulty in their decision process. Since decision making commands gathering data and developing alternatives together with conceptual grouping, categorization is one of fundamental and important approaches in decision process—as in cognitive processes such as language acquisition, learning, prediction, production, and inductive reasoning [1]. Amos Tversky’s decision making process, elimination by aspects [2], says that decision making compares all alternatives by aspects; it chooses an aspect; any alternatives without that aspect are eliminated; it repeats this process with as many aspects as needed until there remains only one alternative. With the interpretation of Tversky’s decision process, decision makers group alternatives as categories—also called facets—and scope down to the alternatives in the relevant categories in their decision process. Categories, as the abstraction of decision maker’s experience in the world, hence have two roles: a category integrates alternatives into a general description toward a decision-making goal, and provides an instructional manual for inferences to guide for the goal [3]. The conceptual system of decision making therefore depends on construction of categories and their selection.

This paper focuses on a dynamic categorization—called faceted navigation in literature. The topic of the paper is, in particular, facet selec-
tion/ordering using the total gain ratio, applied to a movie search engine working with an ontology. Based on our initial work [4], this study focuses on implementing a dynamic faceted navigation system, as a decision support system, which groups results based on search context using the Semantic Web technology. The current state of search systems requires users to review long lists of items from search results laboriously. A conventional solution to its alleviation is the provision of refining fixed categories. Search engines provide categories that classify items in a search result into several groups, and users may refine their searches by selecting a category value more relevant to what they want to find. A fundamental problem residing in this approach is that refining categories are predefined and fixed regardless of the difference in search results. Considering human decision making, faceted navigation should dynamically evolve and provide relevant information by contextual grouping, but due to lack of contextual knowledge, fixed category systems do not fit on the condition. Fixed categories may provide insensible information. For example, IMDB1, an online database on films, has fixed categories in the fixed order such as “Refine By Type”, “Refine By Provider”, “Refine By Top Titles”, “Refine By Top Names”, “Refine By Genre” and “Refine By Payment Model” (Figure 1). Querying “CBS videos” to its search engine produces all videos provided by “CBS”, but in this case the category “Refine By Providers” becomes superfluous and insensible. The combination of free queries and facets selection asks for dynamic generation of facets.

The dynamic category selection is conceivable because the system is based on an ontology which makes meanings of instances and their properties

---

understood by the system. Using the information from the ontology, the selection is attained by the measurement of the relevancy of the properties. Items resulted in a query search hence are structured with their associated properties. For example, if a search result has a book “The Old Man and the Sea”, it has properties in a book ontology such as “Author”, “Publication Date” and “Publisher” (Figure 2). The properties for a given book item have their own values that describe the characteristics of the item. Importantly, properties may be used for classifying book items as well as for describing them. In this research, categories are properties whose values classify items

Figure 1: Fixed categories in IMDB

Figure 2: Properties of a book
in search results; items with the same property value go into the same group. The dynamic selection of categories hence can be rephrased as selecting properties that classify items depending on search results; different search results induce different selections. It needs to define a criterion used in the selection process—that is proposed in this research. The ordering in categories is another issue because a category beyond the selection range of order cannot be displayed. The ordering in categories may partially affect on the quality of classification. This research therefore suggests a dynamic category system by developing a measure for a criterion in the category selection and an algorithm for the category ordering.

The presentation is organized after this section as follows. Section 2 provides some background knowledge and information in related works about the dynamic faceted navigation. Section 3 presents the overview of the dynamic category system, the dynamic category selection measure, and the selection procedure. Section 4 proposes a reordering scheme for selected categories an its algorithm. Section 5 demonstrates an example on the calculation of the proposed measure and the reordering scheme. Section 6 evaluates the performance of the system in comparison with the fixed category system. The section of conclusion comes last.

2. Related Works

This research assumes that a search system is based on ontology. While the term ontology is used in many different disciplines such as philosophy, metaphysics, and information science, this study uses the meaning of ontology defined in the Semantic Web technology formed with the Resource Description Framework (RDF), RDF schema, the Ontology Web Language (OWL) [5, 6, 7, 8]. Ontology is a specification of a representational vo-
cabulary for a shared domain of discourse (definitions of classes, relations, and functions with other objects) [9]. Its expressiveness of resources and their relationships facilitates many information systems adopting it as their structural framework for organizing information and data, mainly due to its availability of information search.

In literature, categories is also called facets and the search using categories faceted search. The faceted search is known to be a good alternative in search for complex search queries, because the mean query length is about 2.4 words and most users do not use advanced search functionality according to [10]. Literature about faceted search or navigation mainly focuses on facet construction, user interface, and facet selection. For facet construction, how to map documents to facet hierarchy is a key issue. Many authors used varied ways: algorithms based on lexical subsumption in [11, 12], synsets and hypernym relations in [13], the RawSugar social tagging system in [14], and personalized PageRank values in [15]. For user interface, authors are interested in how to enable intuitive discovery-oriented navigation with easy presentation of a complex information space. A design recommendation in user interface is provided in [16], a visualization scheme in [17], and two-dimensional tables with axes of hierarchical categories and correlating pairs of facets in [18, 19], respectively. Facet selection is important in the drill down operation. [20] selects facets in the perspective of finding the “surprising” aspects of the data, [17] uses the combination of each facet’s a-priori, query independent usefulness and a dynamic, entropy-based measure in selection. [21] maximizes the utility of the selected facets to each individual searcher, and [22] sets up and solve the facet selection problem with approximation algorithms. [23, 24] applied the faceted search to product search on the Web.
In generating categories dynamically, a measure needs to be used in selecting one property against another. A fundamental concept that is necessary in order to construct such a measure is Shannon’s entropy. In information theory, entropy is a measure of the average information amount of a given content when it is considered as the value of a random variable [25, 26]. Consider the situation of using compression by variable-length codes, where different words are encoded in bit strings of different length. Frequent words are encoded with shorter codes and rare words with longer codes. Entropy is a precise lower bound of the expected number of bits required to encode an instance, denoted by a random variable $X$ and sampled with a probability distribution $p$ [27]. It is also considered as a measure of unpredictability; high entropy states that the outcome is unpredictable and zero entropy that the outcome value is constant. Shannon [25] defined the entropy $H$ of a discrete random variable $X$ with the finite number of values $x$ as

$$H(X) = E\left[\log \frac{1}{p(X)}\right] = -\sum_x p(x) \log p(x)$$

(1)

where $p(x)$ is the probability mass function of an outcome $x$. When two random variables involved, the joint entropy is defined as

$$H(X, Y) = E\left[\log \frac{1}{p(X, Y)}\right] = -\sum_{x,y} p(x, y) \log p(x, y)$$

(2)

To define the measure for faceted navigation, another conceptual measure information gain is needed. In Shannon’s theory, a receiver receives a corrupted message with noise. The situation may be represented with two random variables $X$ and $Y$, where $X$ denotes the original source of the message and $Y$ noise [28]. The amount of information of a random variable $X$ contained in another $Y$ is called information gain $IG(X; Y)$ or mutual information $I(X; Y)$, which is usually defined with conditional entropy. Conditional entropy of two random variables $X$ and $Y$ is the amount
of information that is newly obtained by $X$ beyond the information about $X$ contained in $Y$; that is, the additional cost of encoding $X$ given the encoding of $Y$. It is defined by the conditional probability

$$H(X|Y) = H(X, Y) - H(Y) = -\sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(y)}$$

(3)

where $p(x, y)$ is the probability that $X = x$ and $Y = y$. Using the conditional entropy, the information gain is defined as

$$IG(X; Y) = H(X) - H(X|Y)$$

(4)

According to [29], it is an asymmetric measure of the difference between two probability distributions, and also the change in information entropy from a prior state to a state that takes some additional information. This means that the entropy of $X$ is obtained by adding some extra information described with the conditional entropy $H(X|Y)$ to the information $IG(X; Y)$ in $X$ presented by $Y$.

The generation of dynamic categories has not been studied well, and it is hard to find an article on it in the literature. Park et al. [30], one of the rare studies, proposed an e-mail classification agent using a category generation method. Their agent generates categories dynamically, based on the information of titles and contents in e-mail messages. Since titles and contents in e-mail messages contain unstructured information that is hard to classify, categories cannot be prebuilt before search and selected from the prebuilt. Thus, they are generated on the spot, and because of that, users cannot anticipate what categories will be generated. This implies that it is very hard to avoid meeting odd and unintuitive categories that are not easily understandable. Users tend to dislike such categories because classifying e-mail messages according to such might be very awkward and make later references not easy. Search results considered in this research, however, are structured
because search is based on structured ontology databases; ontologies have predefined properties which can be used as categories that classify search results. That structuredness also enables precise analysis in search results and may generate more intuitive and understandable categories. Khoshnevisan [31] proposed a method that identifies a set of search categories based on category preference obtained from search results. Items in a search result are ranked according to the relevancy to query and each has category preference information. The preference of a category is determined by the order of items found from that category. On the other hand, all items of search results in this paper become correct answers to the given query because ontology search is used; items in a search result can not be ranked because they have no concept of priority and have the same relevancy to query. All items in an ontology search result are evaluated together, rather than dealt separately, in generating categories. Also, the generated categories may provide more appropriate classification than the method proposed by Khoshnevisan because this research uses an ontology-based complete search while the Khoshnevisan’s uses the conventional text-based search.

3. The Dynamic Category Selection

3.1. The overview of the dynamic category system

The goal of this research is to propose a dynamic category system where categories on search results dynamically change. Figure 3 shows the overview of the system. All instances are stored in the ontology. When a user asks a search query to the ontology, it returns its result as resources. The system calculates values of measures for categories to be proposed next using the knowledge stored in the ontology. The given number of categories are selected based on the values according to a certain criterion. The selected
categories are displayed to the user with the resources. Then the user checks if the displayed resources has an item she or he wants. If it is found, the system stops. Otherwise the user select a category, which makes the system find the corresponding property and identify the property’s values from the ontology. The system expands the values to the user, and the user choose a value. Using the chosen category and value, the system updates the collection of resources to the reduced version of resources that collects all items whose property is the chosen category and whose property value is the chosen value. Based on the updated resources, category measures are recalculated, and the given number of categories are selected and displayed again. This gives the user another chance to check her or his item. If one is not found, it goes to the step of the selection. This process continues until the user finds an item.
3.2. A measure for the category selection

An ontology and resources as a search result under the ontology become inputs to the process. The process analyzes the resources with the knowledge of the structure of the ontology, and outputs a classification of categories, also called properties, according to the resources. The dynamic category generation in ontology search compels selections of appropriate properties predefined in a given ontology. A natural arising question is then in what criterion to select properties. A notion is introduced to answer that question. Property A explains Property B well if the category made by Property A contains items homogeneous in the value of Property B. Table 1 lists ten movie titles with three relevant properties: “Director”, “Genre” and “Country”. If the “Genre” of a movie is known to be “Action”, it is easily deduced from Table 1 that the movie was released in “Korea” and directed by “Gyeongtaek Kwak”. If a movie is known to be directed by “Changdong Lee”, its “Genre” is “Drama” and it is a Korean movie. This deduction is not always available because the knowledge of “Korea” gives no hints on properties of “Genre” and “Director”. The main reason for this unavailability is that movies with the value “Korea” of the property “Country” have many different values in the other properties.

When movies in Table 1 are classified in the property “Country”, eight movies belong to one category of value “Korea”, but their similarity within the category is very small because they have different values in properties “Director” and “Genre”. This is the same on the items in the category of value “USA”. They have almost no similarity except that they are released in the same countries. When the movies are classified in the property “Genre”, the extent of similarity increases. Movies in “Action” and “Thriller” categories are all directed by “Gyeongtaek Kwak” and “Junho
Bong”, respectively. Even though the “Drama” category still has different values in the property “Director”, the others have the same value. The classification with the property “Director” also increases similarity in comparison with that with the property “Country”. When several classifiers are available in faceted navigation of objects, a question arises: which classifier is the best for faceted navigation? It is conventionally stated that a good classifier classifies objects similar in values of other properties or features into the same category, and different objects into different categories [32]. This question naturally leads to the requirement of a measure of similarity within each category, made by a given classifier. If one property explains the others better than any other does in such a measurement, the property should become the best classifier. In the example of Table 1, the measures of the properties, “Director”, “Genre” and “Country”, represent their abilities to explain “Genre” and “Country”, “Director” and “Country”, and “Director” and “Genre”, respectively.

The total gain ratio (TGR) is used in this paper for the measurement of the extent to which a property explains the others. Every property has a TGR value; if the value is bigger, it explains the other properties better.

<table>
<thead>
<tr>
<th>Movie</th>
<th>Director</th>
<th>Genre</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friend</td>
<td>Gyeongtaek Kwak</td>
<td>Action</td>
<td>Korea</td>
</tr>
<tr>
<td>Typhoon</td>
<td>Gyeongtaek Kwak</td>
<td>Action</td>
<td>Korea</td>
</tr>
<tr>
<td>The Host</td>
<td>Junho Bong</td>
<td>Thriller</td>
<td>Korea</td>
</tr>
<tr>
<td>Mother</td>
<td>Junho Bong</td>
<td>Thriller</td>
<td>Korea</td>
</tr>
<tr>
<td>Tokyo!</td>
<td>Junho Bong</td>
<td>Drama</td>
<td>Korea</td>
</tr>
<tr>
<td>SILENCED</td>
<td>Donghyeok Hwang</td>
<td>Drama</td>
<td>Korea</td>
</tr>
<tr>
<td>Secret Sunshine</td>
<td>Changdong Lee</td>
<td>Drama</td>
<td>Korea</td>
</tr>
<tr>
<td>Poetry</td>
<td>Changdong Lee</td>
<td>Drama</td>
<td>Korea</td>
</tr>
<tr>
<td>The Beaver</td>
<td>Jodie Foster</td>
<td>Comedy</td>
<td>USA</td>
</tr>
<tr>
<td>Battleship</td>
<td>Peter Berg</td>
<td>War</td>
<td>USA</td>
</tr>
</tbody>
</table>
TGR for a property \( p \) in a set of properties \( \mathcal{P} \) is formally defined by the sum of values of a subordinate measure, the gain ratio (GR).

\[
TGR(p) = \sum_{\hat{p} \in \mathcal{P} \setminus \{p\}} GR(\hat{p}; p)
\]  

(5)

For a property \( p \), gain ratio values of any other properties \( \hat{p} \) about \( p \) are calculated and summed to obtain its TGR value. After calculation of TGR values for all properties in \( \mathcal{P} \), those properties are ordered and considered as categories.

The calculation of the total gain ratio is sequentially built on top of the average information amounts, entropies, of the properties in \( \mathcal{P} \). Before describing the details of the calculation, we define some notations. First let \( \mathcal{O} \) be the ontology that defines the domain of the search, and \( \mathcal{R} \) be the set of resources obtained as a search result. Each element in \( \mathcal{R} \) is an instance of the type of \( \mathcal{R} \), a subject \( s \) of a triple \((s, p, o)\) for some property \( p \) and object \( o \) in the triple store \( \mathcal{T}(\mathcal{O}) \) representing \( \mathcal{O} \). Let \( \mathcal{P} \) be the set of properties whose domains are of the type of \( \mathcal{R} \); that is, the set of properties such that triples containing them have the subjects in \( \mathcal{R} \). For example, if a search result \( \mathcal{R} \) contains books such as “The Old Man and the Sea”, “Pride and Prejudice”, and “Oliver Twist”, the type of \( \mathcal{R} \) may be “Book”, and its properties “title” and “author” belong to \( \mathcal{P} \). Now let \( \mathcal{T} \) be the set of triples such that their subjects are search result items in \( \mathcal{R} \) and their properties are in \( \mathcal{P} \) with some objects; that is,

\[
\mathcal{T} = \{(s, p, o) \in \mathcal{T}(\mathcal{O}) \mid s \in \mathcal{R}, p \in \mathcal{P}\}
\]

If “The Old Man and the Sea” has the value of property “Author”, “Hemingway”, then a triple

\[
(\text{“The Old Man and the Sea”, “Author”, “Hemingway”}) \in \mathcal{T}
\]

exists. Let \((\cdot, p, \cdot)\) be a set of all triples with the specific property \( p \) for some
subject and object,

\((\cdot, p, \cdot) = \{(s, p, o) \mid \exists s, o : (s, p, o) \in T\}\)

If either the subject or the object of triples is specified, we have more restricted notations such as \((\cdot, p, b)\); that is,

\((\cdot, p, b) = \{(s, p, b) \mid \exists s : (s, p, b) \in T\}\)

Also given the specification of a property \(p_j\) and its object \(b\), the triple set with property \(p_i\) \((p_i \neq p_j)\) and the same subject as the specification is defined by,

\((\ast, p_i, \cdot | \ast, p_j, b) = \{(s, p_i, o) \mid \exists s, o : (s, p_i, o), (s, p_j, b) \in T\}\)

With more specification about the value of a property, the following is defined similarly

\((\ast, p_i, a | \ast, p_j, b) = \{(s, p_i, a) \mid \exists s : (s, p_i, a), (s, p_j, b) \in T\}\)

\(N(S)\) is used to denote the size of a set \(S\). For example, \(N(\cdot, p, \cdot)\) denotes the size of \((\cdot, p, \cdot)\), the number of triples in \(T\) with a property \(p\). \(N(\cdot, p, b)\), \(N(\ast, p_i, \cdot | \ast, p_j, b)\) and \(N(\ast, p_i, a | \ast, p_j, b)\) are similarly defined. \(V_R(p_i)\) denotes the set of all values that are of range of \(p_i\) in \(T\); i.e. \(V_R(p_i) = \{o \mid \exists s, o : (s, p_i, o) \in T\}\). In the similar fashion, given the specification of a property \(p_j\) and its object \(b\), the following set is defined

\(V_R(p_i | \cdot, p_j, b) = \{o \mid \exists s, o : (s, p_i, o), (s, p_j, b) \in T\}\)

Under the notations defined above, all subordinate measures to evaluate TGR can be constructed. The entropy \(H(p)\) for each property \(p \in P\) is calculated by:

\[
H(p) = - \sum_{a \in V_R(p)} \frac{N(\cdot, p, a)}{N(\cdot, p, \cdot)} \log_2 \frac{N(\cdot, p, a)}{N(\cdot, p, \cdot)} \quad (6)
\]

The conditional entropy of \(p_i\) given a specific value \(b\) of a property \(p_j\) is next
evaluated, for every pair of properties \( p_i, p_j \in P \) with \( p_i \neq p_j \),

\[
H(p_i \mid \cdot, p_j, b) = -\sum_{a \in V_R(p_i | p_j, b)} \frac{N(*, p_i, a | *, p_j, b)}{N(*, p_i, \cdot | *, p_j, b)} \log_2 \frac{N(*, p_i, a | *, p_j, b)}{N(*, p_i, \cdot | *, p_j, b)}
\]

(7)

Then the conditional entropy of \( p_i \) given \( p_j \) is defined as the average of those conditional entropy values of \( p_i \) for the values of \( p_j \),

\[
H(p_i \mid p_j) = \sum_{b \in V_R(p_j)} \frac{N(*, p_i, \cdot | *, p_j, b)}{\sum_{c \in V_R(p_j)} N(*, p_i, \cdot | *, p_j, c)} H(p_i \mid *, p_j, b)
\]

(8)

The information gain of a property \( p_i \) about a property \( p_j \) with \( p_i \neq p_j \) is obtained from the difference of the entropy of \( p_i \) and the conditional entropy of \( p_i \) given \( p_j \):

\[
IG(p_i; p_j) = H(p_i) - H(p_i \mid p_j)
\]

(9)

As mentioned earlier, the information gain provides the amount of information on \( p_i \) explained by the information on \( p_j \). In other words, the value of the information gain gives us how much the property \( p_i \) is explained by the property \( p_j \). Since the value of the information gain depends on the number of values of \( p_j \) and the total number of triples, it needs to be normalized, and the classification information is used for that purpose. The classification information of \( p_i \) and \( p_j \) with \( p_i \neq p_j \) is calculated as follows:

\[
CI(p_i; p_j) = \frac{|V_R(p_j)|}{\sum_{c \in V_R(p_j)} N(*, p_i, \cdot | *, p_j, c)}
\]

(10)

The denominator of the classification information is the total number of triples about \( p_i \) in all categories over \( p_j \). The score then represents how many categories of \( p_j \) each triple on \( p_i \) belongs to on average. The gain ratio of \( p_i \) about \( p_j \) is obtained by dividing the information gain with the classification information,

\[
GR(p_i; p_j) = \frac{IG(p_i; p_j)}{CI(p_i; p_j)}
\]

(11)

Thus, the gain ratio is the normalized version of the extent of the explanation.
of \( p_i \) by \( p_j \). When \( p_j \) has a lot of values and the category of each value has fewer triples, the classification information is big, and consequently the gain ratio gets smaller, which is not an ideal case. The total gain ratio of a property \( p \) measures the extent of explanation of all the other properties in \( \mathcal{P} \) by selecting \( p \) in (5).

When the ontology \( \mathcal{O} \) and the resources \( \mathcal{R} \) are given, the following summarizes the procedure of the calculation of the total gain ratio briefly:

1) Identify the type of \( \mathcal{R} \) to find the set \( \mathcal{P} \) of properties whose domains are of the type of \( \mathcal{R} \) by referring to \( \mathcal{O} \).
2) Construct a set \( \mathcal{T} \) of all triples that describe properties in \( \mathcal{P} \) and their values for items in \( \mathcal{R} \).
3) Calculate an entropy \( H(p) \) for each property \( p \) in \( \mathcal{P} \).
4) For each pair of properties \( p_i, p_j \in \mathcal{P} \) with \( p_i \neq p_j \), calculate a conditional entropy \( H(p_i \mid \cdot, p_j, b) \) for each \( b \in V_R(p_j) \), and \( H(p_i \mid p_j) \) by averaging \( H(p_i \mid \cdot, p_j, b) \).
5) For each pair of \( p_i \) and \( p_j \) with \( p_i \neq p_j \), calculate the information gain \( IG(p_i; p_j) \), the classification information \( CI(p_i; p_j) \) and the gain ratio \( GR(p_i; p_j) \) of \( p_i \) about \( p_j \).
6) Calculate the total gain ratio \( TGR(p) \) for each property \( p \) in \( \mathcal{P} \).
7) Order the properties in \( \mathcal{P} \) according to the decreasing order of their TGR scores. A property with a higher score is a better classifier.

3.3. A replacement procedure in the category selection

From Section 3.2, properties whose domain consists of resources are arranged as categories by the TGR score values. This may be improved by looking down in the tree spanned out of the type of resources on the ontology further. Consider the situation when a candidate property in \( \mathcal{P} \) has
its range type which has also subsequent properties. In Figure 4, the class “Film” has many properties that link many classes as their ranges. The class “Actor” is one of them that is linked by a property, say \( p_f \), of “Film”, and has also other properties (e.g. \( p_g \)) as their domain to link other classes such as “Education”, Country”, “Height” and so on. If a property \( p_g \) of “Actor” gives more amount of information than \( p_f \) in terms of the TGR score values, then it is better that \( p_g \) replaces \( p_f \) for categories. This replacement process may go down through the tree on the ontology to find a better category. This procedure is given as follows:

1) According to the decreasing order of scores of the TGR, order properties in \( \mathcal{P} \) for resources \( \mathcal{R} \).

2) If the range of a property \( p \in \mathcal{P} \) has no properties that have it as their domain, \( p \) is ranked to its original order.

3) If the range of a property \( p_i \in \mathcal{P} \) has properties that have it as their domain—denote the set of the properties by \( \mathcal{P}(p_i) \), then replace \( p_i \) for \( p_j \in \mathcal{P}(p_i) \) and calculate the TGR score for \( p_j \). For the replacement, consider the sequence of triples of \((s, p_i, o_i), (s_j, p_j, o_j)\) where \( o_i = s_j \), as one triple \((s, \tilde{p}, o_j)\) by introducing \( \tilde{p} = p_i p_j \). Then replace \((s, p_i, o_i)\) for \((s, \tilde{p}, o_j)\) and calculate the TGR for \( p_j \).
4) Choose \( p_j \in P(p_i) \) that has the highest TGR score and compare it with that of \( p_i \).

5) If the TGR score of \( p_i \) is the greater than that of \( p_j \), \( p_i \) remains in its original position of the property ordering of \( P \).

6) If the TGR score of \( p_j \) is the greater than that of \( p_i \), \( p_i \) replaces \( p_j \) in the property ordering of \( P \).

By considering all the properties branched out of resources, it is possible to generate the best properties that explain the resources well.

4. Reordering by Similarity

\[
\begin{align*}
\text{Algorithm 1: Property Reordering} \\
1 & \quad L = \emptyset; U = P; \\
2 & \quad \text{pick } p \in U; \\
3 & \quad L \leftarrow L \cup \{p\}; U \leftarrow U - \{p\}; \\
4 & \quad \text{while } U \neq \emptyset \text{ do} \\
5 & \quad \quad \text{forall the } p_u \in U \text{ do} \\
6 & \quad \quad \quad \text{sim}_L(p_u) = \sum_{p_s \in L} \text{sim}(p_u, p_s); \\
7 & \quad \quad \text{end} \\
8 & \quad \quad p^* \leftarrow \arg \min_{p_u \in U} \text{sim}_L(p_u); \\
9 & \quad \quad L \leftarrow L \cup \{p^*\}; U \leftarrow U - \{p^*\}; \\
10 & \quad \text{end}
\end{align*}
\]

This section discusses a method to reorder categories resulted from the last section by using a similarity measurement. In the category selection, the TGR score reflects the amount of knowledge about the other properties and as a property contains the most amount of knowledge about the others, it is selected first. Hence the selection chooses properties in the order of the amount of knowledge about the others. When the similarity of a pair of properties is defined as the extent to which they share their domain instances in triples, choosing a property that has the lowest similarity from the already
chosen may give a better classifier because it induces more different items. Using this idea, the categories may be reordered and may provide a better choice to users. Algorithm 1 describes this rearrangement procedure.

$L$ and $U$ are a pair of ordered and unordered sets of properties, respectively, with $L \cup U = \mathcal{P}$ and $L \cap U = \emptyset$. The algorithm starts from the empty $L$ with $\mathcal{P}$ for $U$. Initially it selects an arbitrary $p$ for $L$ from $U$, and for each step selects a property $p^*$ from $U$ that has the minimum similarity with $L$. The similarity of a property to the set $L$ is the sum of the similarity values to properties in $L$.

<table>
<thead>
<tr>
<th>Algorithm 2: Similarity Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 sim($p$, $q$)</td>
</tr>
<tr>
<td>2 R ← $V_{\mathcal{R}}(p)$;</td>
</tr>
<tr>
<td>3 C ← $V_{\mathcal{R}}(q)$;</td>
</tr>
<tr>
<td>4 total ← 0;</td>
</tr>
<tr>
<td>5 foreach $v \in R$ do</td>
</tr>
<tr>
<td>6 $\text{val}<em>v \leftarrow \min</em>{w \in C} \bar{x}_p(v) \cdot \bar{x}_q(w)$;</td>
</tr>
<tr>
<td>7 $w^* \leftarrow \arg \min_{w \in C} \bar{x}_p(v) \cdot \bar{x}_q(w)$;</td>
</tr>
<tr>
<td>8 R ← $R - {v}$;</td>
</tr>
<tr>
<td>9 C ← $C - {w^*}$;</td>
</tr>
<tr>
<td>10 total ← val$_v$ + total;</td>
</tr>
<tr>
<td>11 end</td>
</tr>
<tr>
<td>12 return total;</td>
</tr>
<tr>
<td>13 end</td>
</tr>
</tbody>
</table>

Algorithm 2 shows the calculation of the similarity between two properties, which is used in Algorithm 1. Let $p$ and $q$ be two properties in $\mathcal{P}$. For each property $p$, the resource set $\mathcal{R}$ are partitioned into several groups according to its range values; equivalently, $\mathcal{T} = \bigcup_{v \in V_{\mathcal{R}}(p)} (\cdot, p, v)$ for $p \in \mathcal{P}$ then each partite $(\cdot, p, v)$ is represented with the 0-1 vector $\bar{x}_p(v) = (a_r)_{r \in \mathcal{R}}$ such that $a_r$ is 1 if $(r, p, v) \in \mathcal{T}$; otherwise, 0. For a pair of properties $p$ and $q$, a matrix $M(p, q)$ is defined by the product of two representative vectors for partites of $p$ and $q$, i.e. $(\bar{x}_p(v) \cdot \bar{x}_q(w))_{v,w}$ where
Table 2: An example for the dynamic category selection

<table>
<thead>
<tr>
<th>Movie</th>
<th>Genre</th>
<th>Country</th>
<th>Director</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highway Star</td>
<td>Comedy</td>
<td>Korea</td>
<td>Sangchan Kim, Hyeonsu Kim</td>
</tr>
<tr>
<td>The Killer Elite</td>
<td>Action, Thriller</td>
<td>USA, Australia</td>
<td>Gary McKendry</td>
</tr>
<tr>
<td>War of the Arrows</td>
<td>Action, Adventure</td>
<td>Korea</td>
<td>Hanmin Kim</td>
</tr>
<tr>
<td>Live Free or Die Hard</td>
<td>Action</td>
<td>USA</td>
<td>Len Wiseman</td>
</tr>
<tr>
<td>SILENCED</td>
<td>Drama, Thriller</td>
<td>Korea</td>
<td>Donghyeok Hwang</td>
</tr>
</tbody>
</table>

$v \in V_R(p), w \in V_R(q)$. This implies that each entry of the matrix is the number of common subjects $s$ such that $(s, p, v)$ and $(s, q, w)$ belongs to $T$.

The similarity value of $p$ and $q$ is calculated on $M(p, q)$ recursively. For $v \in V_R(p)$, select $w^* = \arg\min_w [M(p, q)]_{v,w}$ with its minimum value $val_v$.

Then the submatrix is obtained by removing the row and column corresponding to $v$ and $w^*$. The final similarity value is obtained by adding $val_v$ and the similarity value from the submatrix.

5. An example

This section uses the example of Table 2 to explain better the process of the dynamic category selection described in the last sections. In Table 2, the search result consists of five movies whose type is “Movie” with three properties: “Genre”, “Country” and “Director”. To arrange the properties in order of the TGR, their entropy values first should be calculated. Since “Genre” has eight triples with five different values, its entropy is 2.16 from (6). In the same fashion, “Country” and “Director” has 1.46 and 2.58, respectively. The highest entropy value for “Director” implies that the property value is the most unpredictable. To calculate the gain ratio, the resources of the result is classified according to the values of “Genre”. It is shown in Ta-
Table 3: Movies classified by “Genre”

<table>
<thead>
<tr>
<th>Comedy</th>
<th>Target: Country</th>
<th>Target: Director</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highway Star</td>
<td>Korea</td>
<td>Highway Star: Sangchun</td>
</tr>
</tbody>
</table>
|               | Highway Star: Hyeonsu
| Action        | The Killer Elite: USA | The Killer Elite: Gary |
|               | The Killer Elite: Australia | War of the Arrows: Hanmin |
|               | War of the Arrows: Korea   | Live Free or Die Hard: Len |
| Thriller      | The Killer Elite: USA | The Killer Elite: Gary |
|               | The Killer Elite: Australia | SILENCED: Donghyeok |
|               | SILENCED: Korea     | SILENCED: Donghyeok    |
| Adventure     | War of the Arrows: Korea | War of the Arrows: Hanmin |
| Drama         | SILENCED: Korea | SILENCED: Donghyeok    |

Table 3 when targets are set to “Country” and “Director”. The conditional entropy of “Country” given the values of “Genre” are $0 (= -1 \log_2 1)$ for “Comedy”, $1.5 (= -2/4 \log_2 2/4 - 1/4 \log_2 1/4 - 1/4 \log_2 1/4)$ for “Action”, $1.58 (= -1/3(\log_2 1/3 + \log_2 1/3 + \log_2 1/3))$ for “Thriller”, $0 (= -1 \log_2 1)$ for “Adventure”, and $0 (= -1 \log_2 1)$ for “Drama”.

The conditional entropy of “Country” given “Genre” can be obtained from averaging these values. Its value is $1.07 (= 4/10 \times 1.5 + 3/10 \times 1.58)$. The corresponding information gain has the value $0.39 (= 1.46 - 1.07)$ from the difference between the entropy and the conditional entropy. The classification information is $0.5 (= 5/10)$ since the number of range values of “Country” is five and the number of triples whose range type is “Country” ten. The gain ratio is calculated to be $0.78 (= 0.39/0.5)$. In the same way, the gain ratio of “Director” about “Genre” is $2.88$ by setting “Director” as the target. Then the total gain ratio of “Genre” is $3.66 (= 0.78 + 2.88)$ by adding the gain ratio values of “Country” and “Director” about “Genre”. The total gain ratio scores of all properties are given in Table 4. The properties are arranged in the order of “Country”, “Genre” and “Director” ac-
Table 4: The TGR scores calculated from IG, CI, and GR

<table>
<thead>
<tr>
<th>Target Property</th>
<th>Genre IG</th>
<th>CI</th>
<th>GR</th>
<th>Country IG</th>
<th>CI</th>
<th>GR</th>
<th>Director IG</th>
<th>CI</th>
<th>GR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genre</td>
<td>0.52</td>
<td>0.3</td>
<td>1.73</td>
<td>1.49</td>
<td>0.07</td>
<td>2.22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Country</td>
<td>0.39</td>
<td>0.5</td>
<td>0.78</td>
<td>1.17</td>
<td>0.86</td>
<td>1.36</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Director</td>
<td>1.61</td>
<td>0.56</td>
<td>2.88</td>
<td>1.15</td>
<td>0.43</td>
<td>2.67</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TGR</td>
<td>3.66</td>
<td></td>
<td></td>
<td>4.4</td>
<td></td>
<td></td>
<td>3.58</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

According to the TGR scores.

Alternatively, another arrangement may be obtained by using the measure of similarity. For binary representative vectors, the vector indexes correspond to the movies shown in Table 2. Since five resources in the table, vectors are five dimensional. Consider the calculation of the similarity between “Country” and “Genre”. Since

\[ V_R(“Country”) = \{ “Korea”, “USA”, “Australia” \} \]
\[ V_R(“Genre”) = \{ “Comedy”, “Action”, “Thriller”, “Adventure”, “Drama” \} \]
the subjects \( s \) satisfying \( (s, “Country”, “Korea”) \in \mathcal{T} \) are “Highway Star”, “War of the Arrows”, and “SILENCED”, and its corresponding representative vector is \((1, 0, 1, 0, 1)\). In the same way, the vector for the range values “USA” and “Australia” are \((0, 1, 0, 1, 0)\) and \((0, 1, 0, 0, 0)\), respectively. The property “Genre” has representative vectors \((1, 0, 0, 0, 0)\) for “Comedy”, \((0, 1, 1, 1, 0)\) for “Action”, \((0, 1, 0, 0, 1)\) for “Thriller”, \((0, 0, 1, 0, 0)\) for “Adventure”, and \((0, 0, 0, 0, 1)\) for “Drama”. Also, The property “Director” has \((1, 0, 0, 0, 0)\) for “Sangchan”, \((1, 0, 0, 0, 0)\) for “Hyeonsu”, \((0, 1, 0, 0, 0)\) for “Gary”, \((0, 0, 1, 0, 0)\) for “Hanmin”, \((0, 0, 0, 1, 0)\) for “Len”, and \((0, 0, 0, 0, 1)\) for “Donghyuck”. The products of the vectors for “Country” and “Genre” and for “Country” and “Director” make two matrices in Table 5. \( u_1, \ldots, u_3 \) stand for the values of \( V_R(“Country”) \), \( v_1, \ldots, v_5 \) for those of \( V_R(“Genre”) \), and \( w_1, \ldots, w_6 \) for those of \( V_R(“Director”) \). In the matrices, selections of
entries are denoted by bold-faced numbers. Each entry is the product of two representative vectors. If the value is higher, the two corresponding groups share more items and have higher similarity. For each value of one property, the smallest value of the other is selected in the calculation of the similarity. In the first matrix, \( v_1 \) is selected for \( u_1 \); after removing those row and column, \( v_4 \) for \( u_2 \); after removing \( u_2 \) and \( v_4 \), \( v_5 \) for \( u_3 \). The similarity value of “Country” and “Genre” is the sum of the chosen values, 1+0+0=1. In the same fashion, the similarity of “Country” and “Director” is obtained with its value 0. Since the similarity value between “Country” and “Director” is smaller than that between “Country” and “Genre”, we can switch “Genre” and “Director” in their positions. We have a new arrangement: “Country”, “Director” and “Genre”.

6. Evaluation

This section proves the performance of the proposed scheme by an experimentation. Since this study assumes the Semantic Web environment, the usual unstructured Web data whose search method is text-based is not good for the purpose; it requires that data be organized under an ontology and available to ontology-based search systems. A film ontology was constructed after that of FreeBase.\(^2\) In Figure 5, Class Film has fifteen properties; the classes of type Person, such as Director, Actor, Producer and so forth, have

Figure 5: Structure of Film ontology

Table 6: Search Methods

<table>
<thead>
<tr>
<th>No.</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fixed faceted navigation (F)</td>
</tr>
<tr>
<td>2</td>
<td>TGR faceted navigation without update (T)</td>
</tr>
<tr>
<td>3</td>
<td>TGR faceted navigation with update (T-update)</td>
</tr>
<tr>
<td>4</td>
<td>TGR-similarity faceted navigation without update (T-S)</td>
</tr>
<tr>
<td>5</td>
<td>TGR-similarity faceted navigation with update (T-S-update)</td>
</tr>
</tbody>
</table>
five properties; Class Country has two properties. All individual data were collected from the Web site of DAUM Movies\(^3\) by using a parser. The class instance data were excerpted as many as possible, and property information, about such as directors, actors and so on, was also collected from the site. The ontology store thereby contains more than one thousand movie instances and nine thousand person instances.

This experimentation assumes the situation of browsing—to see what is available for certain combinations of facets. It compared the performances for the conventional fixed faceted navigation and four variant methods of the proposed scheme with TGR and the similarity measure in Table 6. For each method, a navigation system was implemented to help users search. Twenty-four users were asked to use these systems representing the given methods to search their target items with help of their hinting category values about the targets. Table 7 displays all search items that users searched. Each user was assumed not to know exactly the target item he or she wants to find but to know three hinting categories and their values for the unknown target item. A user started with a search result and the categories generated from the initial query by each system. Then he or she chose a category and selected the corresponding value according to the hinting category-value pairs of the target item; it is said to be a step in a navigation process. A system generates the reduced search result with the selection of a category and its value as its input and moves to the next step of the process. Since a selection of category and its value strengthens the search condition, the result narrows down to the smaller number of search items.

The fixed faceted navigation (F) always provides a fixed set of categories

\(^3\)http://movie.daum.net (2012)
<table>
<thead>
<tr>
<th>TARGET ITEM</th>
<th>HINTING CATEGORIES/VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Scent of Love</td>
<td>Producer Taewon Jung, Art Director Sangho Ha, Story Writer Hain Kim</td>
</tr>
<tr>
<td>The Scam</td>
<td>Writer Hojae Lee, Music Director Youngjin Mok, Distribution Co. Showbox</td>
</tr>
<tr>
<td>Skills</td>
<td>Cinematographer Ola Magnestam, Editor Johannes Runeborg, Costume Designer Karolina Nilsson</td>
</tr>
<tr>
<td>To Catch a Virgin Ghost</td>
<td>Writer Jungwon Shin, Cinematographer Hyunjae Oh, Art Director Sangman Oh</td>
</tr>
<tr>
<td>The Sorcerer’s Apprentice</td>
<td>Story Writer J. W. von Goethe, Production Co. Jerry Bruckheimer, Music Director Trevor Rabin</td>
</tr>
<tr>
<td>Barking Dogs Never Bite</td>
<td>Editor Ensoo Lee, Writer T. Sohn, J. Bong, Music Director Sungwoo Jo</td>
</tr>
<tr>
<td>Source Code</td>
<td>Distribution Co. Synergy Pierre Perrault, Art Director Renee April, Costume Designer</td>
</tr>
<tr>
<td>Aegis</td>
<td>Music Director Trevor Jones, Story Writer Harutoshi Fukui, Editor William M. Anderson</td>
</tr>
<tr>
<td>Antarctic Journal</td>
<td>Initial Release Date 2005-05-09, Music Director Kenji Kawai, Writer J. Bong, P. Yim</td>
</tr>
<tr>
<td>Dororo</td>
<td>Story Writer Osamu Tezuka, Writer Masa Nakamura, Initial Release Date 2007-10-25</td>
</tr>
<tr>
<td>A Million</td>
<td>Initial Release Date 2009-08-06, Story Writer Sungkyu Jo, Cinematographer Jaehoon Lyu</td>
</tr>
<tr>
<td>Jealousy Is My Middle Name</td>
<td>Writer Chanok Park, Editor Chanok Park, Producer Jokwangsoo Kim</td>
</tr>
<tr>
<td>The City of Violence</td>
<td>Music Director Joonsuk Bang, Art Director Hwasung Jo, Writer Wonjae Lee</td>
</tr>
<tr>
<td>My Love</td>
<td>Director Han Lee, Costume Designer Heejoon Ahn, Production Co. Ozone Film</td>
</tr>
<tr>
<td>Over the Rainbow</td>
<td>Director Jinwoo Ahn, Music Director Hojoon Park, Initial Release date 2002-05-17</td>
</tr>
<tr>
<td>The International</td>
<td>Cinematographer Frank Griebe, Producer Lloyd Phillips, Initial Release Date 2009-02-26</td>
</tr>
<tr>
<td>The Ring Virus</td>
<td>Writer Dongbin Kim, Music Director Il Won, Art Director Bongoh Kim</td>
</tr>
<tr>
<td>The Last Legion</td>
<td>Writer Valerio Manfredi, Music Director Patrick Doyle, Roberto Caruso</td>
</tr>
<tr>
<td>The Girl Is Bad Ass</td>
<td>Initial Release Date 2010-03-25, Producer Prachya Pinkaew, Distribution Co. DigitalKIN</td>
</tr>
<tr>
<td>Stormbreaker</td>
<td>Story Writer Anthony Horowitz, Writer Anthony Horowitz, Andrew MacRitchie</td>
</tr>
<tr>
<td>Soo</td>
<td>Distribution Co. Cinema Service Youngwoo Shin, Music Director Byungwoo Lee</td>
</tr>
<tr>
<td>HAHAHA</td>
<td>Music Director Youngjin Jung, Cinematographer Hongil Park, Production Co. Jeonwonsa</td>
</tr>
<tr>
<td>Running Turtle</td>
<td>Art Director Sungjoo Nam, Initial Release Date 2009-06-11, Writer Yeonwoo Lee</td>
</tr>
<tr>
<td>Countdown</td>
<td>Music Director Youngkyu Jang, Cinematographer Taekyung Kim, Art Director Hongsam Yang</td>
</tr>
</tbody>
</table>
displayed for whatever the search query and the chosen value are. This is the conventional faceted navigation. The second (T) calculates TGR scores when the system generates the search result from the initial query statement. All categories are ordered according to the TGR scores and the fixed number of them are displayed to users. In this search process, TGR scores are never updated. If one category is selected and removed in the next step, the category not displayed with the highest TGR score is added to the display to show the same number of categories in display. The third (T-update) is the same as the method T but updating TGR scores according to the search result at every step. The fourth (T-S) uses the similarity measure in reordering categories without update. As in method T, it orders all categories by TGR scores with respect to the initial search result and reorders them by the similarity. It never updates the order in the following steps. The last (T-S-update) updates TGR and similarity measures at every step.

Each navigation process proceeds step by step according to the selections of categories and values until the number of items in results becomes at most fifteen. When the final result contains the target item, the process is considered as a success. The process should never go on infinitely because a user never keeps selecting categories in reality. This experimentation gave five steps as the maximum. The number of steps, corresponding to the number of clicks to users, is counted until the search finds its target item successfully. If the process passes the fifth step with neither the reduction of the result up to at most fifteen items nor the containment of the target item, the process is considered to go forever and marked as a failure; this case is scored 10. All methods display five categories to users.

Table 8 shows the mean number of steps each system took to find tar-
Table 8: Mean Values for the Number of Steps

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>T</th>
<th>T-S-update</th>
<th>T-S</th>
<th>T-update</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5.8333</td>
<td>5.8333</td>
<td>5.6667</td>
<td>4.8750</td>
<td>3.6250</td>
</tr>
</tbody>
</table>

get items with the given samples. Method T is almost the same as the fixed faceted navigation in performance, which is reasonable because it is fundamentally the same as Method F. Method T is different only in the initial display while both keep the fixed set of categories. For the different scores, significance tests were performed by the non-parametric studentized bootstrapping method [33, 34] to see whether the differences are statistically significant. An open source statistical language, Language R, with a package “bootstrap” was used for implementation. The methods were pairwise compared with mean comparison hypotheses. $R = 999$ bootstrap samples were replicated out of values for the items in Table 7. Each bootstrap sample is then used to evaluate student-$t$ statistics $t^*_r$. The resulted $p$ value is calculated as follows:

$$p = \frac{1 + \#\{t^*_r \geq t\}}{R + 1}$$

where $t$ is the student-$t$ statistic value for the original sample and $\#\{A\}$ is the number of elements satisfying a statement $A$. The results of this significance test are given in Table 9. In the table, the value of each cell is the $p$-value for the alternative hypothesis that the mean for the corresponding row method is greater than that for the corresponding column method. Under the 5% significance level, method T-update significantly improves the performance that any other methods but method T-S. Since the method accompanies update of TGR value at every step, it is a complete dynamic faceted navigation scheme, and is clearly better than the fixed schemes of methods F and T. While the mean step number for T-update is smaller than that of T-S, their difference is insignificant. From the comparison of
methods F and T, it seems that how to determine the fixed set of categories does not give an impact in performance. Unfortunately, the result of the table shows that the use of the similarity measure in reordering has unclear results on the performance improvement. Using the similarity at every step (T-S-update) resulted in aggravated score in Table 8 in comparison with the mean step value of the method T-S, even though they are not significantly different. If TGR and the similarity measure is used only at the initial step without update (T-S), there is a little improvement but it is not significantly different. Since the reordering with the similarity is based on how much a category does not share values with the top TGR category, it seems not to measure its natural information containment but the relative extent far off from the top category. The similarity does not seem to be a good measure in faceted navigation. It is supposed to give just a disruptive effect on the faceted navigation using TGR.

In this experimentation, a simple query statement “film” is used to obtain the initial search result. Using other or complex queries has no problem in the current experimentation scheme while complex query statements require the collection of the vast amount of data. Considering the requirement of the construction of structured data under the Semantic Web technology, it is very hard to construct a system for experimentation that is very close to the real situation. The implication obtained in this section, however, can be easily extended to the cases of complex queries as well as other simple

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>T</th>
<th>T-S-update</th>
<th>T-S</th>
<th>T-update</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>-</td>
<td>0.481</td>
<td>0.511</td>
<td>0.439</td>
<td>0.025*</td>
</tr>
<tr>
<td>T</td>
<td>-</td>
<td>-</td>
<td>0.521</td>
<td>0.820</td>
<td>0.023*</td>
</tr>
<tr>
<td>T-S-update</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.796</td>
<td>0.023*</td>
</tr>
<tr>
<td>T-S</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.120</td>
</tr>
</tbody>
</table>
queries. The reason is that the search process depends only on the information contained in the initial search result—the extent to which the result is explained by other related categories, but not on the form of queries. In particular, asking a complex query about film may put out resulted items of narrower scope because of the stronger condition, but they are still related to the same kinds of properties as in the simple query “film”; this suggests that the search step for the complex query meets the same state as in the simple query, and the search process proceeds in the same fashion. The improvement of search by the dynamic faceted navigation is therefore useful even when using complex queries.

7. Conclusion

This paper proposes a dynamic faceted navigation under the Semantic Web environment through a dynamic category system in the context of the ontology-based search. Selecting categories dynamically has advantages in that it provides different categories for users according to their search contexts. If users query with different key words, they might assume different contexts and naturally categories provided with the results should be adapted to their situations. The proposed approach evaluates search results with a measure and shows different categories according to the results. This is significant because this faceted navigation leads to different paths of search experience that suit for users’ search contexts, and users eventually reach the items with ease that they want to search for. It can save much time for users and improve the efficiency in search. While this study focuses on the ontology-based search in the Internet, the proposed dynamic faceted navigation can be extended to any category systems based on structured data. Together with the proposed measure, which analyzes
information of search results to select categories, this paper also suggests a replacement procedure to improve the categories obtained from the measure, and an reordering algorithm based on a measure of similarity. The experimentation for performance evaluation shows that the outcome from the dynamic faceted navigation is more efficient than the fixed counterpart while using the similarity for reordering gives no effect. One possible future research direction is to re-validate the effect of the similarity for reordering because this paper consider only search steps as a unique performance measure. Since using similarity requires more computational work at each step, a computational study might be needed to measure the true effect of this approach.

References


