Table of Contents

1. Introduction

The information retrieval techniques should provide users with easy access to the information in which they are interested. Unfortunately, characterization of the user information need is not a simple problem. Furthermore, retrieving useful information from a specific database is an important issue in many areas. For example, current search engines use the keyword-based queries to help users find information and they are still playing a very important role in Internet services. However, the simplicity of the approach prevents the formulation of more elaborated querying tasks [1]. In practice, using keywords to query normally assumes that users know exactly what they want [2]. For many kinds of data, however, it is difficult to specify query objects with a complete set of keywords. Moreover, even all the data are characterized and annotated, difficulties may still arise because users are likely to express the concept description from different angles and at different levels of perceptions. Especially, this problem is often confronted while retrieving the information from an ecological database. For example, users may have different interests while staring at the same bird. One could pay attention to its body size and another could be attracted by its beautiful color. In addition, most of them would have a problem to describe all of the bird’s features clearly and completely. It thus poses new challenges for data indexing, querying and retrieving. Due to the inexact and uncertain perceptions of users, data modeling should be intentionally designed to support fuzzy data content, structure, and presentation. It must also allow users to conduct a query by the fuzzy perceptions rather than by exact keywords. For those queries without explicitly giving the keywords or with the semantic contents that cannot be translated into crisp or clear forms, we call such queries the fuzzy semantic queries.

Try to imagine the following possible scenarios. When you are climbing a mountain, walking in a park, or simply idling along the avenue, a bird is flying over or stops in front of you suddenly. You are deeply attracted by its colorful body, harmonic sound, or even the way it flies. At this curious moment, is there any handy tool that can help us to find out the bird’s name and its relative information? PDAs or Tablet PCs with a bird searching system seem to be a reasonable and straightforward solution. While retrieving the information, however, can you tell the bird’s features at a glance of it? The answer should be “no” for most of us because people tend to remember things for a short period of time. Even when you are watching a bird in a zoo, can you describe its outward appearance without any observation error or judgment deviation? Can you describe all of the bird’s features clearly and completely? Most importantly, how to translate the ambiguous conditions in mind into exact keywords becomes a critical issue while conducting the query. Same demands might happen in a bird-watching activity, an outdoor-ecological course, or a school extracurricular activity. However, since most users cannot explicitly describe all the neat features of birds, fishes, or plants, that will become an obstacle for learners to access any advanced contents under such circumstances.

Over decades, some research works have been dedicated to the fuzzy query domains [3] [4] [5] [6]. To develop a fuzzy semantic query system, in practice, there are two common problems to be overcome. First, a flexible query interface has to be built because users do not know how to specify the queries. The weighting vector makes good beginnings for users who want to express the importance of individual feature. It can either be created in some swoop at the beginning of the system or it can be built and changed adaptively as the system processes the query. Second, the system has to locate the most distinguishing features to make the weighting vector more reasonable. The user should be careful when he/she specifies the features that have significant amount of weights. Our previous work used the probability basis to construct the weighting vector [5]. In this project, we propose an attribute relevance analysis method to enhance the precision of the weighting vector. Probability basis is easier to handle, but relevance analysis does a better job of approximating the importance of attributes.

In addition, a number of statistical and machine learning-based techniques for relevance analysis have been proposed in the literature [7] [8]. Unlike the work in [9] which used relevance analysis to remove irrelevant data attributes and further compact the training data, we apply attribute relevance analysis to derive the attribute weighting vector. Retrieving a bird database, for example, anyone searching for the target bird might have no idea about the database in existence. Therefore, the searching algorithm usually doesn’t identify a good match because the query is inherently imprecise and the data in the database are totally unknown. If an attribute analysis is done at the beginning, the statistical results can be used to stress the strongly relevant attributes by increasing their weights. That will make weighting vector begin to come much closer to real situations. Note that the analysis would not be very effective if the attribute values are evenly distributed in the database. Though it is not perfect, it does evolve some results better than without it. It is usually done on a case-by-case basis and stored in the system, but it can also be done in advance and used again and again, as long as the database is not dramatically changed.

2. Fuzzy Semantic Query Model

As we mentioned, characterization of the user information need is not a simple problem. In ecological databases, for example, it is very hard to formulate a query for a new learner or an inexperienced user. To overcome such problems, we have proposed an information retrieval method that allows users to conduct a query by transforming human perceptions into numerical data based on their impressions or opinions on the target. Such a fuzzy semantic query is a quite common information retrieval task which often happens in our daily life. For example, while you are watching birds outdoors and need to look up some information for an interested bird. At this moment, most users cannot exactly describe all the features that are needed to distinguish a bird. In our previous work [6], we tried to process fuzzy semantic queries by integrating two different abstract levels: centrality and intensity. The centrality is a set of features that provide the quantification of class (category) similarity while the intensity is a set of features that dedicate to the quantification of characteristic (attribute) similarity. Therefore, an instance in the database is expressed by a feature vector S(C, I), integrating its category centrality vector C and its attribute intensity vector I. Before getting into more details of them, we define some basic components.

Definition 1. The fuzzy semantic query can be modeled as [**D**,**Q**,**F**(q, d_x)], where

(1) **D** represents a set of instances in the database.

(2) **Q** is a set of query vectors for the user information needs.

(3) **F**(q, d_x) is a ranking function which defines the dissimilarity between the query q and the instance d_x, where $q \in \mathbf{Q}$ and $d_x \in \mathbf{D}$.

In this model, we do not intend to eliminate the semantic ambiguity neither in database nor in user query. On the contrary, we explicitly represent and process ambiguity by introducing two content descriptors, which will be illustrated in the following.

Definition 2. The instance vector and query vector are both composed of a centrality vector and an intensity vector, where

(1) **centrality vector *C*** is a set of terms that provide the quantification of class or category similarity. Let $\boldsymbol{C} = (C_1, C_2, \ldots, C_m)$, where m is the number of category and C_i denotes the similarity between the instance and category i.

(2) **intensity vector *I*** is a set of terms that provide the quantification of characteristic or feature similarity. Let $\boldsymbol{I} = (I_1, I_2, \ldots, I_n)$, where n is the number of features that characterize an object. Each feature I_j will have p_j terms if there are p_j options (or attributes) for the feature j. Each term in I_j indicates its similarity degree for that corresponding option.

The centrality vector C is composed of the terms C_1, C_2, …, C_m, where C_i (i = 1, …, m) represents the similarity between the instance and category *i*. For example, a vector C = {0.3, 0, 0.7, 0, 0} means that the instance has 30% of similarity with the first category, 70% of similarity with the third category, and no similarity with the rest of categories. The intensity vector I is composed of the feature vectors I_1, I_2, …, I_n, where I_j (j = 1, …, n) will have p_j terms if there are p_j options (attributes) for the feature j. That is, each vector I_j is composed of its attribute intensity terms, and each of which indicates its similarity degree for that corresponding option (See Table 1).

Example 1. A centrality vector ***C*** = {0.2, 0, 0, 0.8, 0} means that the instance has 20% of similarity with the first category, 80% of similarity with the fourth category, and no similarity with the remaining categories.

Example 2. A vector I_1 = {0.8, 0.2} denotes the user has 80% of confidence that the observed bird is bigger than a sparrow. Similarly, a vector I_2 = {0, 0.7, 0.3, 0} means the user has 70% of confidence for the long beak and 30% of confidence for the hooked beak for a given instance.

Similarly, the query vector Q can be formed as Q(C′, I′), where C′ is the centrality vector of the query object and I′ is the intensity vector of the query object. With different level of assignments, the centrality and intensity degrees can help the system to capture what the user observed. The dissimilarity measure (total distance *D*) is denoted as *D*(S(C,I), Q(C′, I′)), which is composed of two parts: the centrality distance d_1 and the intensity distance d_2. It can be computed with the following equation:

$$D(S,Q) = w_1 * d_1(C,C') + w_2 * d_2(I,I'), \tag{2-1}$$

where the weights w_1 and w_2 indicate the different significant levels for centrality and intensity. Note that it allows an adjustable weighting in combing these two distances according to distinctive characteristics of the query objects. Moreover, in order to emphasize the significance of an attribute that is relatively more important or certain than others, we can give a different weighting for each attribute. Therefore, d_2 can be derived as follows:

$$d_2(I,I') = \sum_{i=1}^{n} r_i * d_2(I_i, I_i'), \tag{2-2}$$

where r_i is the weighting factor for the ith attribute. One reasonable choice of r_i is the inverse of the total number of options in attribute i, which is called probability basis here.

This prior experimental bird searching system has proved that this approach is generic and adaptable to many application fields. However, the attribute weighting vector used in the system is fairly subjective. If we can provide more information from the existing database, there is a room for more effectiveness. Based on the prior results, we intend to address the effectiveness issues by proposing a different approach for the attribute weighting vector. In the next section, we will have a dedicated look at the proposed method.

3. The Proposed Attribute Relevance Analysis Method

As stated above, our previous work on the weighting factor r_i is on such probability basis. This is applicable regardless of the data distribution in the database. However, with the existing database as the training set, we can derive more certain information from attribute relevance analysis, and thus get better weighting factors to distinguish an object from others. The general idea behind attribute relevance analysis is to compute some measure that is used to quantify the relevance of an attribute with respect to a given class (category). Such measures include information gain, the Gini index, uncertainty, and correlation coefficients [10]. In this project, we use the information gain analysis technique to compute the attribute relevance. How to calculate the information gain is illustrated as follows.

Let S be a finite set of data samples. Suppose these samples are classified into m distinct classes C_i (for $i = 1, \ldots, m$), where class C_i contains x_i samples. Let p_i denote the probability of class C_i. Therefore, $p_i = x_i / x$, where x is the total number of samples in set S. The entropy is commonly used to measure the impurity of a set of samples. The expected information needed to classify a given sample can be given by

$$\begin{aligned} I(x_1, x_2, \ldots, x_m) &= -\sum_{i=1}^{m} p_i \log_2 p_i \\ &= -\sum_{i=1}^{m} \frac{x_i}{x} \log_2 \frac{x_i}{x}, \end{aligned} \tag{3-1}$$

which is known as the entropy of S. If the set S can be partitioned into the subset $\{S_1, S_2, \ldots, S_v\}$ based on an attribute A with values $\{a_1, a_2, \ldots, a_v\}$. That is, S_j contains those samples in S that have value a_j of A. Let S_j contains x_{ij} samples of class C_i. The expected information based on the partitioning by A is given by

$$E(A) = \sum_{j=1}^{v} \frac{x_{1j} + \ldots + x_{mj}}{x} I(x_{1j}, \ldots, x_{mj}). \tag{3-2}$$

Thus, the information gain obtained by this partitioning on A is defined by

$$Gain(A) = I(x_1, x_2, \ldots, x_m) - E(A). \tag{3-3}$$

If there are n attributes used to determine the class of the training samples, we have to calculate the information gain for each attribute A_k, for k = 1, …, n, by using (3.2) and (3.3) recursively. After obtaining the information gains for all the attributes $A_1, A_2, \ldots, A_n$, we can derive the weighting factor for each attribute:

$$r_i = \frac{Gain(A_k)}{\sum_{k=1}^{n} Gain(A_k)}. \tag{3-4}$$

The attribute with the highest information gain is considered the most discriminating attribute of the given database. Note that the weighting factor can still be adjustable by users. For example, if the query object is special in its color attribute, the color can be used as the main attributes while other attributes as auxiliary attributes. There is, however, no mathematically rigorous definition of similarity that accounts for semantic recognition or perceptual judgment of human beings. We adopt Euclidean distance as similarity measurements between a query and targets in the database. The measuring cost is thus highly subject to the number of instances in the database. Since the size of the database is almost fixed, both the computational complexity and the memory requirements are under control.

4. A Set of Flexible Weighting Vectors

The fuzzy semantic query model usually doesn't identify a perfect match because the query is inherently imprecise. For this reason, the system will output the top five probable matches listed by the descendent order of their match degrees from left to right. It is also usually impossible to prove that the results obtained through these weighting vectors are the best obtainable because the user plays an important role in judging the query quality. In our case, though the given query example satisfies a user, it might have different results for another example. The most important issue here is that the system does allow users to distinguish the importance among features based on their opinions.

Two sets of weighting factors are used to improve the drawbacks due to the uncertainties of users. The first set, i.e., w_1 and w_2, is used to indicate the relative significance levels of the centrality and the intensity. The second one, i.e., R, is used to emphasize the implicit degrees of importance among features. It is potentially useful when the user cannot be reasonably sure about this feature. With the assistance of our user interface, he/she can easily decrease the weighting to its effect. On the other side, the user can also increase the weighting for the feature in which he/she is the most confident. Our system shows a very promising result because the input procedure of weighting vectors is rather simple and appears to be quite reasonable.

5. Works on the Handheld Devices

There are three key types of handheld devices: two-way email pagers, mobile telephone handsets, and PDAs [11]. A handheld device must meet the following characteristics: (1) It must operate without cables, except temporarily recharging or synchronizing with a desktop. (2) It must be easily used while in one's hands, not resting on a table. (3) It must support the addition of additional applications or support Internet connectivity. Though a diverse range of products can be categorized as handheld devices according to the above straightforward definitions, we only examine the PDA (Personal Digital Assistant), a remarkable, tiny, and fully functional computer that dominates the current market.

Though originally simply intended to store contact information, take notes, and keep track of daily appointments in the 1990s, PDAs now have evolved into small computers for running applications, playing games or music, and downloading information from the Internet. In terms of their size and mode of data entry, today's PDAs fall into two major categories: handheld computers and palm-sized computers [12]. Compared to palm-sized computers, handheld computers tend to be larger and heavier. For data entry, handheld computers typically use a miniature keyboard in combination with a touch screen while palm-sized computers use a stylus and touch screen exclusively in combination with a handwriting recognition program. PDAs typically have one of two types of operating systems, Palm OS (3Com) or Pocket PC (formerly called Windows CE, Microsoft). Because Palm OS is specifically tailored to the basic uses of a PDA, it takes up less memory and runs faster, and is easier to use. Therefore, Palm OS dominates the market at the beginning. However, Pocket PC can easily support color displays, graphics, miniaturized Windows packages (e.g., Word and Excel), and other devices (such as built-in MP3 players or MPEG movie players). Because of its tight integration with Microsoft office applications, Pocket PC is challenging Palm OS, and thousands of third-party softwares are available for both operating systems.

As we know, PDAs are designed to work with your desktop or laptop computer. Therefore, they need to work with the same information in both places. To synchronize your data to or from your PDA, you have to install a data synchronization application (HotSync for Palm OS, ActiveSync for Pocket PC) on your computer to connect the PDA to your PC through a cable, modem accessories, infra-red (IR) light, or wireless methods. You will also have to have versions of your handheld's applications installed on your PC so that you can exchange information between your PDA and PC.

According to a report in the IThome magazine, more than 13 millions of PDAs were sold in 2002, and 18 millions of PDAs are estimated to be sold this year. The popularity of PDAs makes both technology and market change very quickly. Despite faster speed and smaller size, more and more exciting features are added on the PDAs. For example, PDAs and GPS (Global Positioning System) receivers will be combined into one handheld device. Some PDAs will be able to capture and store images, as a digital camera does. Manufactures are interested in using the Bluetooth short-range radio system to connect PDAs to other devices. Hewlett-Packard is developing software that could wirelessly connect a PDA to a printer. New PDAs will enhance the Internet wireless connectivity. The only one question left for you is “How much can I afford to spend on a PDA? ”, not “Can the PDA answer my requirements? ”.

The Tablet PC is designed for mobile computer users who have been relying on a combination of laptops and handheld devices. Aside from taking the best from a standard laptop, Tablet PC adds an impressive feature of multiple input methods: keyboard, pen, or voice. It allows people to revise, edit, or search their handwritten notes after they have written them on the computer screen using the stylus. With the ability of handwriting recognition in Tablet PC, you can take notes by handwriting and transform them into digital texts. Because Chinese and Japanese are pictorial languages with thousands of individual characters, it is a Herculean task to input these characters into an electronic document [13]. That is a really great news for Asian consumers though the online handwriting processing is still seeking improvements. Another exciting feature is its Journal application, which uses the pressure sensitive feature to make the ink of your handwriting looked much more like real ink, just as if you were handwriting on a piece of paper.

Tablet PCs have a lot of great features. Since it is a fully functional PC, it can run a wider range of softwares and services than other handheld devices. However, a disadvantage of Tablet PCs is their higher price and unclear future in the market. We already have one working version of our bird searching system on Tablet PC and will revise to another version running on PDAs without costly re-programming.

As handheld devices are rapidly becoming popular, many schools are finding that a mixture of desktops, laptops, and handheld devices can be used to meet a variety of educational needs. Recent advances in mobile technologies also open a new space of possibilities for education. As we will see, the implementation of handheld technologies such as PDAs presents challenges both to the school as an instruction tool and to the classroom culture [14]. Several studies have revealed that the handheld devices have the ability to enhance the student engagement in the classroom. For example, some researchers since 1996 have developed several projects to improve the educational level of the children about five or six years old of Chilean at-risk social class [15]. These projects have shown an amazing improvement by using educational games to enhance the children skills in mathematics and language areas. Penn State Abington has integrated the student use of PDA technology to foster active and collaborative learning experiences in the classroom and laboratory [16]. Using wireless networking would extend the range of classroom scenarios and processes to be served between different locations (e.g., school, the “field”, and home). PDAs appear to be a straightforward solution [17].

Many educators have added more and more functionality into handheld devices. Advances in wireless networks have contributed to this trend. For example, with the wireless communication technologies in PDAs, the Bird-Watching Learning (BWL) system offers a data mining system to assist e-learner to easily search the bird knowledge from the wireless database interface [18]. The BWL system is trying to integrate the scaffolding-aid learning model within a WLAN environment while our new bird searching system is striving to overcome the fuzzy semantic problems while retrieving the bird’s information. By integrating the mobile technology, information technology, and GPS technology, a more complicated learning system can be constructed as a GPS/PDA guide system that supports the learning in the native information [19]. Moreover, handheld devices offer an alternative teaching methodology for a computer science and engineering (CSE) program [20]. That was our former work and will be illustrated more details in the next section.

6. Experimental Results

We develop a bird searching system to demonstrate the effectiveness of our work. Fig. 1 is a snapshot of our system. The categories and features used in the system are shown in Table 2 and Table 3, respectively. The weighting vectors are shown in Table 4 and Table 5, which are derived with the probability basis and relevance analysis, respectively. Fig. 2 is a query bird observed by a user. Users can use a friendly user interface to input all the observed characteristics, and then the system will formulate and transform them into a query vector represented by centrality degrees, intensity degrees, and weighting vectors. The matching policy is based on combining the centrality distance and intensity distance between the query example and instances in the database. However, the searching algorithm usually doesn't identify a perfect match because the query is inherently imprecise. For this reason, the system will output the top five probable matches. The output targets are ordered from left to right according to their match degrees.

To examine the effectiveness of our approach, before conducting a query, users can choose the weighting vector at the lower left corner of the main screen, either on the probability basis or on the attribute relevance analysis. Fig. 3 is the search result on the probability basis while Fig. 4 is the search result on the attribute relevance analysis. The target bird is moved forward from rank 4 to rank 2. It is observed that the use of attribute relevance analysis leads to a better result.

In a fuzzy semantic query system, it is difficult to compare the accuracy of query results because the user plays an important role in judging the query quality. As we have seen, the weighting vector improves the query result in this example. Though it might have different results for another example, it does remind users to specify the most distinguishing features with more care. In our cases, the attributes foot, peck and tail have the highest information gains. Users have to pay more attention in describing them.

In this project, while performing the attribute relevance analysis, we only consider that an instance is uniquely classifiable, i.e., that each training sample can belong to only one class. In the experimental system, however, a sample may belong to more than one class with different levels of similarity as described in Section 2. To simplify the problem, we only take into account the majority class of an instance. Note that an instance will have the highest value of similarity for the majority class in its centrality vector.

The next experiment is used to examine the effect of flexible weighting vectors. Table 6 shows the default weighting vector and the modified one according to a user's opinions. Figure 6 is a query example and Fig. 7 is the search result on the probability basis while Fig. 8 is the search result on the user’s opinions. It is observed that the target bird is moved forward from rank 4 to rank 2.

The system that runs on PCs is the first version of our system. For educational purposes, the second version is running on a PDA with Pocket PC operating system. Figure 9 shows some screen shots of the system. On the other hand, since the Tablet PC will allow you to mark up your digital document with a pen and keep the annotations in digital format, we have revised this system into a Tablet PC version (See Fig. 10).

Note that, the data entry screens are divided into seven sections, allowing the users to easily and quickly enter their query specifications based on how they feel about the query object. Figure 11 to Fig. 17 are the data entry screens in our system, each of which is related to one of the options in the selection area. Each screen (area) is designed for the data entry of a specific vector. Such well-structured data entry interface provides the users a convenient way to input data. Moreover, the system also offers a data input interface for the system manager to enter all bird information into the database. The procedure is just the same as how the users input their queries and thus allows the system manager to quickly increase the size of the database.

Let’s conduct another experiment. Assume a user has seen a bird like the one shown in Figure 18. The user may formulate its query as Table 7. The output targets are shown in Figure 19. We can see this acceptable result even if some of the features are marked as unknown. Note that the left most bird is the top one with the smallest distance with your query. But, the users can click any one of the outputs for further information such as its full image, sound, brief introduction, live video, and so on.

7. Conclusions

In this project, we propose a promising method to derive a weighting vector based on the attribute relevance analysis for fuzzy semantic query systems. We do not attempt to build up a complicated data model. Instead, we build a bird searching system to demonstrate its effectiveness. A complete data model for an ecological database can be a pretty difficult task. Due to the lack of data models, we also bypassed the use of query languages such as SQL or PSQL. In the experimental system, we allow users to specify the query object with two major content descriptors: centrality and intensity. Since several descriptors are used simultaneously, it is necessary to integrate similarity scores resulting from the matching processes in different feature spaces. Two sets of weighting factors are used to handle this issue and thus improve the drawbacks due to the uncertainties of users. The first set is used to indicate the relative significance levels of the centrality and the intensity. The second one is used to emphasize the implicit degrees of importance among attributes, which is shown to be more effective with the help of attribute relevance analysis. Especially, it is potentially useful when the user can not be reasonably sure about this feature. As a consequence, he/she should drastically decrease the weighting to its effect and pay more attention in describing the most distinguishing features. It should be pointed out that the weighting vector is useful in particular domain, but there is no universal vector that adapts well to all data. Ideally, the attribute relevance analysis should be done in advance and then store the weighting vector in the system. It can be used again and again, as long as the database is not dramatically changed.

Our future work involves further investigation of the balance between the centrality and intensity after introducing the attribute relevance analysis. Moreover, we will derive a more elaborated weighting vector which considers an instance belonging to multiple classes. In addition, the categories classified in biology in fact are meaningless to most general users. We will classify the data samples with their attributes in the next version of our system. But we have to limit the number of categories because too many categories will result in tedious input procedures in the system.

The experimental results have already proved that our approach launches a feasible framework for the fuzzy semantic retrieval task. Moreover, the concept of fuzzy semantic query is based on the observation that many users only vaguely define information needs rather than exactly describe it. Such that, our experimental system also provides pictorial attribute examples and a friendly user interface for users to specify the query object. From a practical viewpoint, our approach is not only suitable for the presented bird searching system but also good for other information retrieval systems, such as plants, butterflies, fishes, and national flags.

Table 1. The feature intensity vectors.

Intensity	I_1 (r options)				I_2 (t options)				…	I_s (k options)			
Attribute	i_{11}	i_{21}	…	i_{r1}	i_{12}	i_{22}	…	i_{t2}	…	i_{ks}	i_{ks}	…	i_{ks}

Table 2. The categories defined in system.

	Category 1	Category 2	Category 3	Category 4	Category 5
Category name	Scolopacidae	Ardeidae	Timaliidae	Phasianidae	Turdidae

Table 3. All the features defined in system.

Feature	Attributes (options) for each feature				
Body size (I_1)	bigger than a sparrow	Similar or smaller than a sparrow	--	--	--
Peck shape (I_2)	duck type	long type	hooked type	short type	--
Tail shape (I_3)	long and forked	long and unforked	short and forked	short and unforked	--
Flying way (I_4)	wave	straight	spiral	--	--
Walking way (I_5)	jump	walk	--	--	--
Feather color (I_6)	dark	light	red	blue	green
Foot length (I_7)	long	short	--	--	--

Table 4. The weighting vector with probability basis.

Attribute	Body size	Peck shape	Tail shape	Flying way	Walking way	Feather color	Foot length
Probability	0.500	0.250	0.250	0.330	0.500	0.200	0.500
Weighting	0.198	0.099	0.099	0.130	0.198	0.079	0.198

Table 5. The weighting vector with attribute relevance analysis.

Attribute	Body size	Peck shape	Tail shape	Flying way	Walking	Feather color	Foot length
Information gain	0.277	0.813	0.618	0.340	0.197	0.277	0.985
Weighting	0.079	0.232	0.176	0.097	0.056	0.079	0.281

Table 6. The weighting vector **R**.

	Weighting vector **R**
Default (probability)	(0.5, 0.25, 0.25, 0.33, 0.5, 0.2, 0.5)
User’s opinions	(1.0, 0.25, 0.25, 0.33, 0.5, 1.0, 1.0)

Table7. The specifications of the example input.

Category/Feature	Attributes (options) for each feature
Category	(0, 100, 0, 0, 0)
Body size (I_1)	(10, 90)
Peck shape (I_2)	(20, 80, 0, 0)
Tail shape (I_3)	(50, 50, 0, 0)
Flying way (I_4)	(33, 33, 33)
Walking way (I_5)	(50, 50)
Feather color (I_6)	(0, 100, 0, 0, 0)
Foot length (I_7)	(90, 10)

Fig. 1. The main screen of the experimental system.

Fig. 2. Query example 1.

Fig. 3. The search result on probability basis.

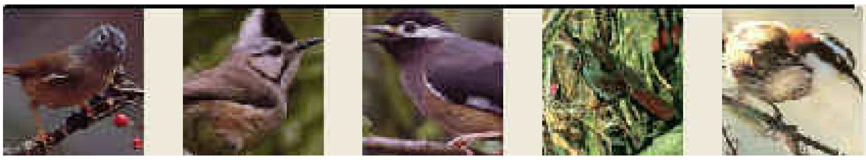

Fig. 4. The search result using relevance analysis.

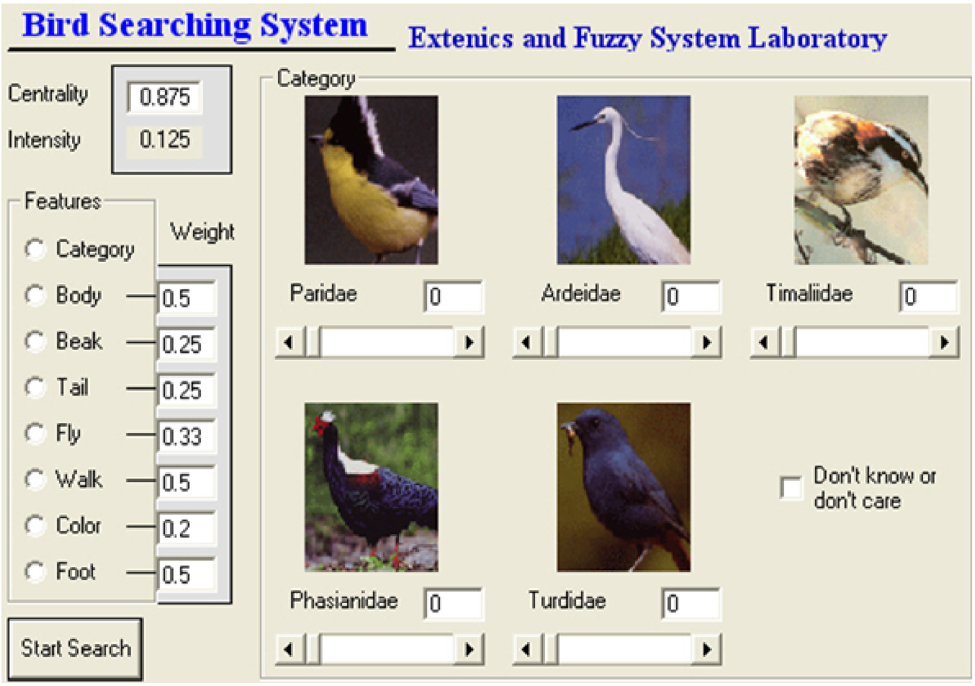

Fig. 5. A user interface that provides flexible weighting vectors.

Fig. 6. Query example 2.

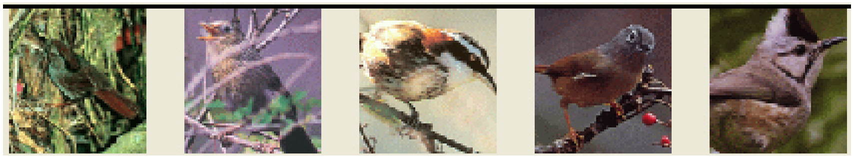

Fig. 7. The result based on the default weighting vector.

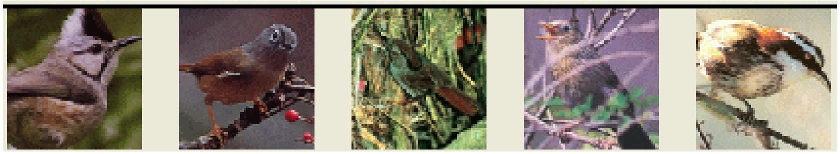

Fig. 8. The result based on the user's weighting vector.

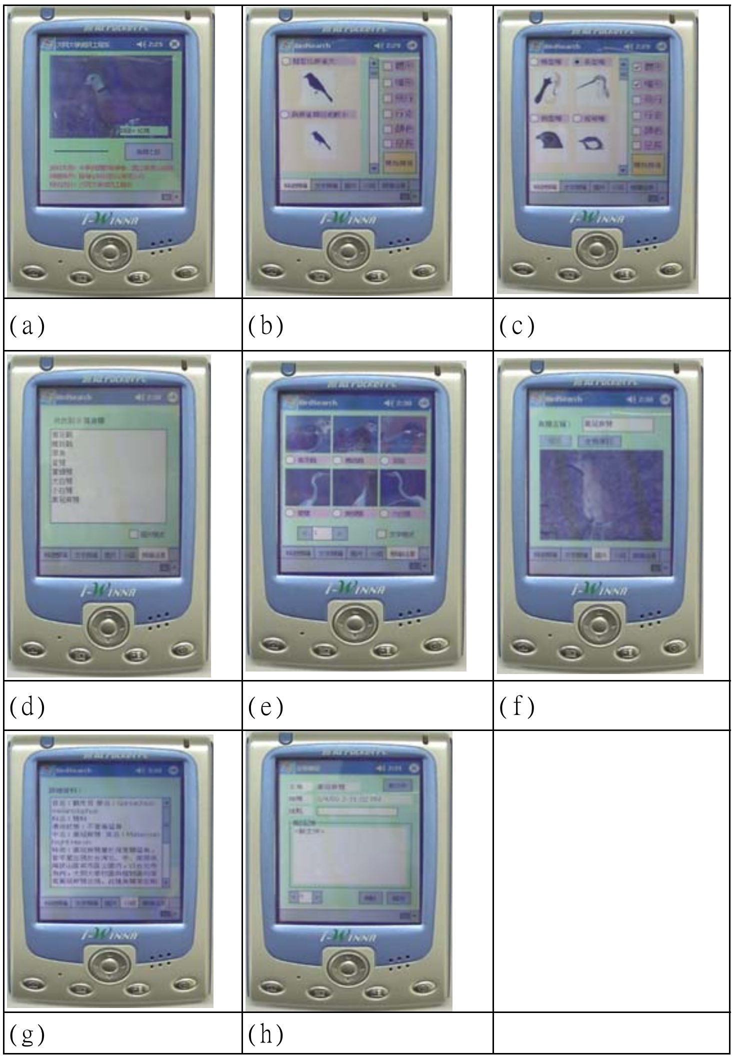

Fig. 9. The screen shots of the system running in PDA.

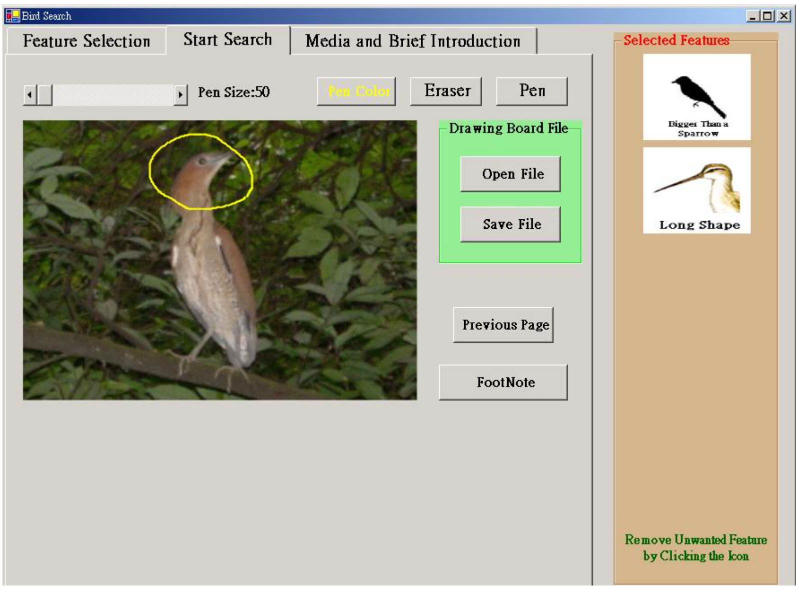

Fig. 10. An example screen of the system running in Tablet PC.

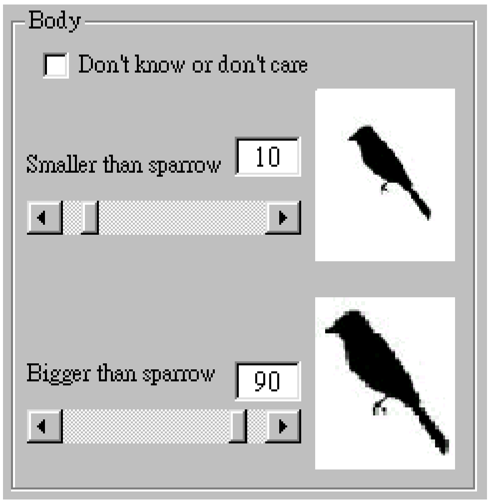

Fig. 11. The data entry screen for the “Body” feature.

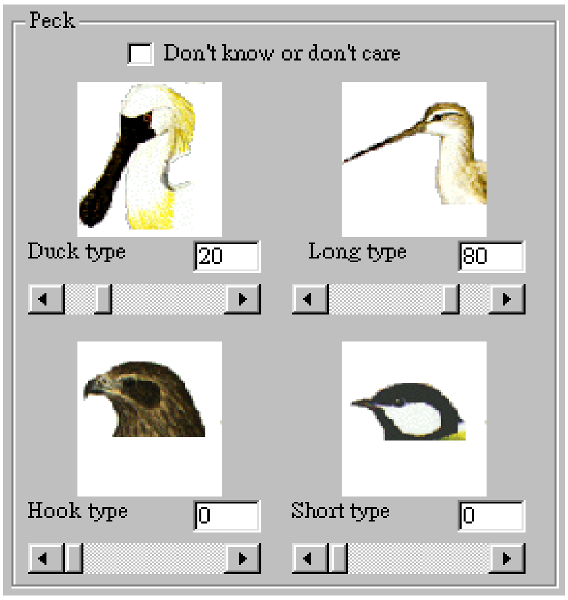

Fig. 12. The data entry screen for the "Peck" feature.

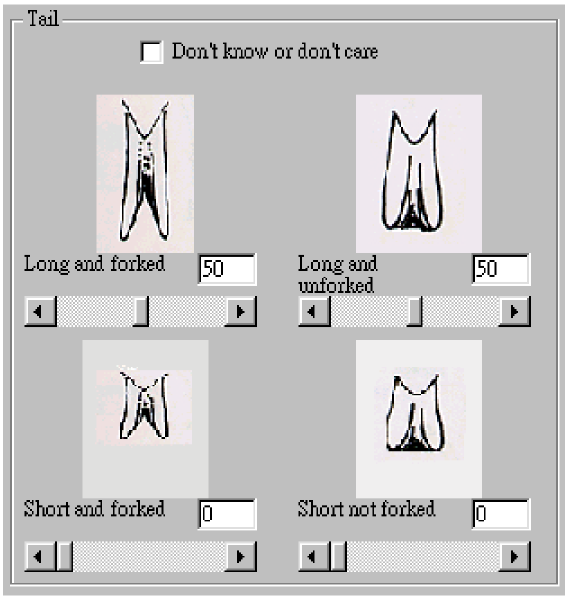

Fig. 13. The data entry screen for the "Tail" feature.

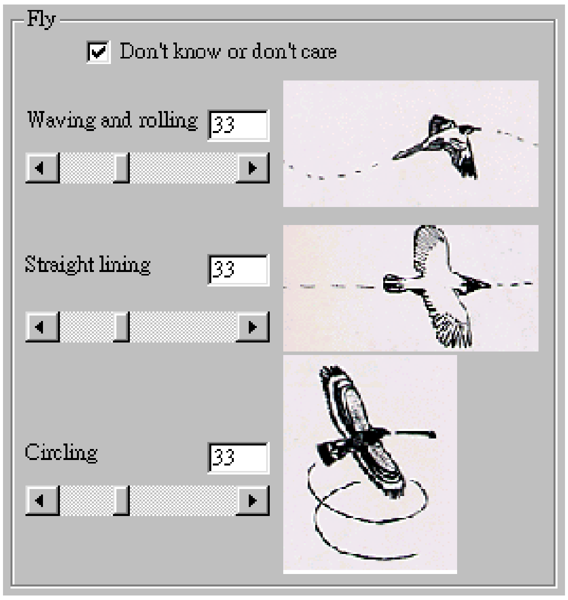

Fig. 14. The data entry screen for the “Fly” feature.

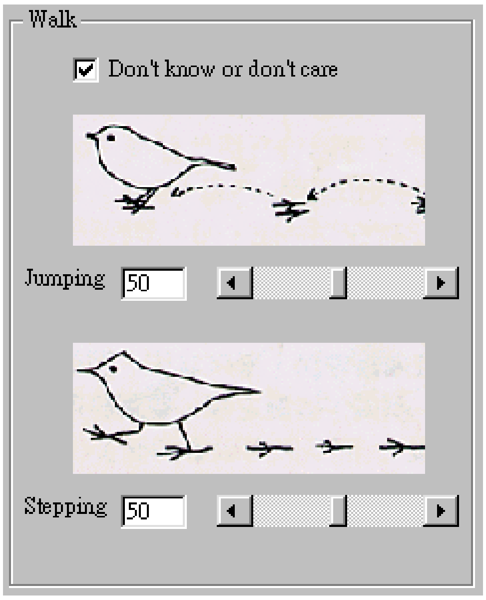

Fig. 15. The data entry screen for the “Walk” feature.

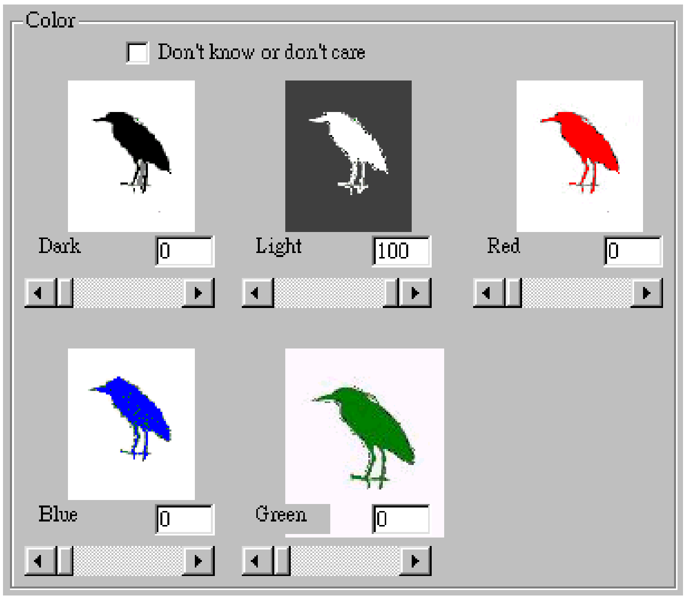

Fig. 16. The data entry screen for the “Color” feature.

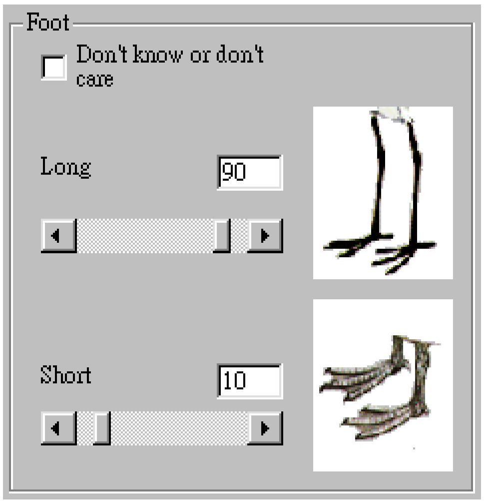

Fig. 17. The data entry screen for the “Foot” feature.

Fig. 18. An example bird seen by a user

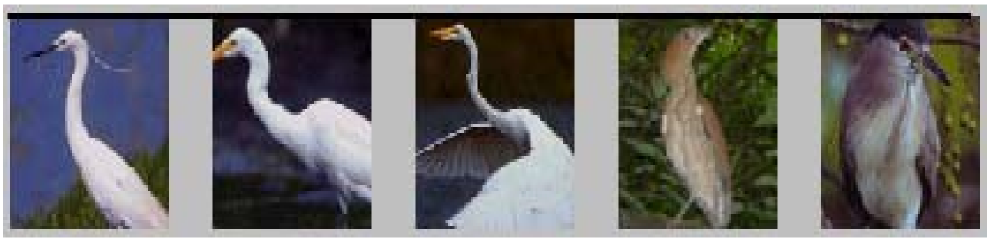

Fig. 19. The search results based on the example query.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison Wesley, Boston, MA, 1999.

[2] E. Megalou and T. Hadzilacos, “Semantic Abstractions in the Multimedia Domain,” IEEE Transactions on Knowledge and Data Engineering, vol. 15, no. 1, pp. 136-160, Jan/Feb. 2003

[3] S.-M. Chen and W.-T. Jong, “Fuzzy Query Translation for Relational Database Systems,” *IEEE Trans. On System, Man, and Cybernetics – Part B: Cybernetics*, vol. 27, no. 4, pp.714-721, Aug. 1997.

[4] S.-M. Chen, Y.-J. Horng, and C.-H. Lee, “Document Retrieval Using Fuzzy-Valued Concept Networks,” *IEEE Trans. On System, Man, and Cybernetics – Part B: Cybernetics*, vol. 31, no. 1, pp.111-118, Aug. 2001.

[5] Y.-P. Huang, L.-J. Kao, T. Tsai, and D. Liu, “Using fuzzy centrality and intensity concepts to construct an information retrieval model,” *IEEE International Conference on Systems, Man & Cybernetics*, Washington, DC, pp.3257-3262, Oct. 5-8, 2003.

[6] Y. Takahashi, “Fuzzy Database Query Languages and Their Relational Completeness Theorem,” *IEEE Trans. On Knowledge and Data Engineering*, vol. 5, no. 1, pp.122-125, Feb. 1993.

[7] H. Almuallim and T. G. Dietterich, “Learning with many irrelevant features,” *in Proc. 9th Nat. Conf. on Artificial Intelligence*, vol. 2, pp.547-552, Menlo Park, CA, July 1991.

[8] G. H. John, “Irrelevant Features and the Subset Selection Problem,” *in Proc. 11th Intl. Conf. on Machine Learning*, pp.121-129, San Fransisco, CA, July 1994.

[9] M. Kamber, L. Winstone, G. Wan, C. Shan and J. Han, “Generalization and Decision Tree Induction: Efficient Classification in Data Mining,” *in Proc. 7th Intl. Workshop on Research Issues in Data Engineering*, pp.111-120, April 1997.

[10] J. Han and M. Kamber, *Data Mining: Concepts and Techniques* , Morgan Kaufmann, San Francisco, CA, 2001.

[11] S. Weiss, *Handheld usability*, John Wiley & Sons, New Jersey, 2002.

[12] HowStuffWorks, Inc., online: http://electronics.howstuffworks.com/pda.htm

[13] S. Ross, “Inventing the Tablet PC,” online: http://research.microsoft.com/features/tablet.asp.

[14] L. Mifsud, “Alternative learning arenas–pedagogical challenges to mobile learning technology in education,” *Proc. of the IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE’02)*, pp.112-116, 2002.

[15] S. Contreras, “Designing a low cost educational handheld device for children,” *Fourth IEEE Int. Caracas Conf. on Devices, Circuits and Systems*, Aruba, pp.I027-1-I027-5, April 2002.

[16] R. Avanzato, “Student use of personal digital assistants in a computer engineering course,” *31th ASEE/IEEE Frontiers in Education Conf.*, vol. 2, pp.F18-F19, Oct. 2001.

[17] N. Pinkwart, C. Schafer, and H.U. Hoppe, “Lightweight extensions of collaborative modeling systems for synchronous use on PDAs,” *Proc. of the IEEE International Workshop on Wireless and Mobile Technologies in Education*, pp.125-129, 2002.

[18] Y.-S. Chen, T.-C. Kao, J.-P. Sheu, and C.-Y. Chiang, “A mobile scaffolding-aid-based bird-watching learning system,” *Proc. of the IEEE Int. Workshop on Wireless and Mobile Technologies in Education*, pp.15-22, 2002.

[19] Y.-W. Chen, Z.-J. Yan, J.-C. Huang, I-H. Peng, and J.-W. Zhan, “Implementation of a PDA/GPS based development platform and its applications in native education,” *IEEE Int. Conf. on Communications, Circuits and Systems and West Sino Expositions*, vol. 2, pp.1556-1560, 2002.

[20] Y.-P. Huang and T.-W. Chang, “Using handheld devices as alternatives to inspire students innovation in designing new information technology,” *Proc. Int. Conf. on Educational Engineering*, Valencia, Spain, July 2003.