Query by Example for Geographic Entity Search with Implicit Negative Feedback

Makoto P. Kato
Department of Social Informatics
Graduate School of Informatics
Kyoto University, Kyoto, Japan
kato@dl.kuis.kyoto-u.ac.jp

Ohshima Hiroaki
Department of Social Informatics
Graduate School of Informatics
Kyoto University, Kyoto, Japan
ohshima@dl.kuis.kyoto-u.ac.jp

Satoshi Oyama
Division of Synergetic Information Science
Graduate School of Information Science and Technology
Hokkaido University, Hokkaido, Japan
oyama@ist.hokudai.ac.jp

Katsumi Tanaka
Department of Social Informatics
Graduate School of Informatics
Kyoto University, Kyoto, Japan
tanaka@dl.kuis.kyoto-u.ac.jp

ABSTRACT

We propose a method of searching for geographic entities in an unknown place with a query by example in a known place. Geographic entity searches are installed in many Web sites such as those for shopping and restaurants. Most of the sites hold classic attribute-based or keyword-based interfaces for entity retrieval; however, specifying each attribute users want is time-consuming, and keywords are not effective for representing users' complex intentions. The proposed query by example method in a map interface allows users to intuitively query by selecting entities in places they know well. The most similar entities to an input are returned based on the similarity varying with individuals. Our proposed method is robust for estimating the similarity using not only selected examples, but also *implicit negative feedback*, which is predicted by how the user selects examples as a query in the map interface. Experimental results proved the effectiveness of our method, and the performance exceeded that of a previously proposed method.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval*

General Terms

Algorithms, Experimentation

Keywords

Geographic entity search, similarity, implicit negative feedback

1. INTRODUCTION

ICUIMC'2010, January 14–15, 2010, SKKU, Suwon, Korea.

1.1 Background

A large amount of geographic entity data, such as shops, hotels and landmarks, are available on the Web. Geographic information in distant places is useful particularly for visitors that have never been there. Geographic entity search has recently received much attention from several Web services. *GourNavi* [4], which is a Japanese restaurant Web guide, stores over 500 thousand restaurants, and Booking.com [2] is an online hotel reservation site with more than 70 thousand hotels listed. Some commercial search engines [3, 5, 1] provide map interfaces for searching geographic entities written on private and commercial Web pages. More geographic entities are being uploaded on the Web, making it more difficult to retrieve desired information for users, even in a small area. Thus, retrieval systems for geographic entities have become more important.

Geographic entities in commercial sites are often searched for by a keyword query, however, keywords are not effective for representing users' complex intentions. For example, suppose that a person wants to find "a Japanese restaurant serving *sushi* for around $40 in the warm atmosphere, and near Tokyo station," and inputs a query "sushi AND $40 AND Tokyo station" to a search system. Although several results matching the query would be returned, the user may find only a few entities relevant to his/her intention in the search results. A keyword query search does not take into account the relevance of attributes such as budget. Also, a non-verbalized intention is naturally ignored. Users often fail to convey their ambiguous or subtle intentions in keyword search systems, e.g., a somewhat high-class Asian-style restaurant that is in the warm atmosphere.

On the other hand, attribute-based search, where a user can specify desired attributes, can take into account the relevance of attributes. However, as geographic entities have many attributes, such as price and equipment, it takes a long time to input several kinds of attributes and requires users to express concrete values for retrieval. Time-consuming searches would not be acceptable for users away from home, and it is rare that search intentions are clearly fixed and easily represented.

These two approaches can be regarded as *deductive*, where an input query is a prior condition, and relevant entities are judged based on that condition [7]. In a deductive search, users are required to translate their search intentions into concrete words or values. For a visitor in a place he/she does not know, however, it is difficult

Figure 1: Query by example interface for geographic entity retrieval. Users can click and select entities relevant to their search intentions on source map (left). When search button is clicked, entities on target map (right) are ranked and returned based on ones selected as query in source map. Search results are shown below maps.

for him/her to determine a condition that meets her/his needs for finding information. This is because it is assumed that a visitor has little knowledge about the place. For a restaurant search, a user may find it difficult to input a deductive query without knowledge as to what kinds of food are popular and the average cost.

1.2 Proposed Method

We use, an *inductive* approach for geographic entity search, a query by example paradigm. Many studies on query by example have been done in multimedia retrieval (often called *content-based retrieval*) [29, 24, 10]. Query by example is a search based on given examples as a query. Given multiple cases relevant to a search intention, information matching these examples is searched, which can be considered as *inductive*. The advantage of an inductive query is that users do not need to express their search intentions explicitly, but only choose examples they think to be relevant in a well-known domain. Selecting examples conveys plenty of information about users' needs, and multiple selections of examples is also an important clue in recognizing intentions that cannot be put into words.

For geographic entity search, the query by example paradigm is naturally suitable. Figure 1 shows the query by example interface we developed for geographic entity retrieval. The interface presents two maps: one is a known place to the user, e.g., a hometown, and the other is an unknown place containing entities to be retrieved. They are called *source map* and *target map*, respectively. Users can select which area is shown for both maps, and which entities in the source map are relevant to their search intentions. The method is used to rank entities in the target map based on entities selected as a query in the source map.

Query by example for geographic entity search enables users to use analogies, which have already been adopted for a search in our previous work [15]. Even if a user does not have any knowledge about a place where he/she wants to find information, he/she can make a query by imagining what entities would be relevant if in a familiar place. Examples cannot be chosen, such as relevance feedback in the target map, because of the lack of knowledge about the place. Our proposed method overcomes the problem by recommending users to use well-known examples in a familiar place as a query. We assume that users have enough knowledge on at least one place such as a hometown or a favorite area.

1.3 Problems

To rank entities based on an input query, the similarity between entities should be defined. There are mainly two problems in measuring similarity.

1. The notion of similarity can vary for different contexts.
2. Accurately measuring similarity between entities in different domains.

First, similarity judgment depends on users and even on their contexts as shown in a psychological study [26]. In other words, the rule of similarity measurement is variable, and changes for each user, query, and context. For example, similarity between two restaurants, one French for a $40 meal, and another Japanese for a $40 meal, can be either high or low by focusing on the prices or the styles. Attributes that are focused on vary for users. For example, people who are concerned about budget would emphasize price similarity, and ones that favor healthy food may judge the similarity only on styles. Context also affects the selection of important attributes, e.g., in a situation that a user feels very cold and wants to eat something hot, Korean and Chinese restaurants may be regarded as almost the same. Thus, this **variability** should be included in the query by example model to correctly measure the similarity between two entities.

The second problem is measuring similarity between entities in different domains, i.e., a known domain and an unknown domain. It is not adequate to calculate similarity between very different types of entities directly. For example, accurately calculating similarity between restaurants in Japan and China is difficult. There is a large gap between price range, popular food and styles, and scales of distance. The differences raise the following questions: how similar is 1,000 yen and 10 renminbi, and whether 5 km in Japan and 5 km in China are sensuously the same. Even in a little distant places, such as Tokyo and Kyoto, this problem makes it difficult to accurately measure similarity. **Inter-domain mapping** is required for bridging the gap between different domains, which relates features in one domain to ones in another domain. For example, we need to find what corresponds to 1,000 yen in China, and then measure the similarity between entities in Japan and China.

In this paper, we tackle the first problem, the variability of entity similarity in query by example search. To predict similarity, we followed the MindReader model proposed by Ishikawa et al. [14], adjusted the model to our method for measuring similarity between geographic entities, and improved it by using features of geographic entity selection. We evaluated the effectiveness of our proposed method using a test set created by volunteers, and clarified its effectiveness by comparing it with MindReader and another baseline method.

The remainder of this paper proceeds as follows: Section 2 summarizes related work on geographic and content-based search. Section 3 introduces a model of query by example for geographic entity search, and our method for predicting the user's notion of similarity is proposed in Section 4. Section 5 shows our implementation of a restaurant search using GourNavi data, presents our evaluation of our method, and discusses the results. Section 7 presents the conclusion.

2. RELATED WORK

2.1 Geographic Search

Markowetz et al. [19, 12] developed a geographic search engine, which constrains and orders Web search results, by focusing a query on a particular region. Lieberman et al. [18] developed STEWARD, which also retrieves Web documents referencing input regions, and visualizes locations the documents refer to. These geographic search systems crawl and index Web pages, as well as process them to be associated with one or multiple locations. The process is called *geo coding*, and extracting and disambiguating geographic references are essential techniques in geographic search [20, 6]. McCurley [20] described various geographic indicators such as zip codes, addresses, and telephone numbers. He showed an application of the geo coding, spatial Web browsing, that allows users to select documents allocated in a map for browsing. Hiramoto and Sumiya [13] proposed a method for automatically retrieving Web pages by users' operation in a digital map. They predict users' intentions from operation sequences containing zooming-in and panning, and create a query for document retrieval. Recently, Web document retrievals related to a specific region have received much attention in geographic research. Methods of efficient indexing and retrieving geographic entities by spatial queries have been studied in the information retrieval communities [17]. However, we believe that a large amount of geographic entities on the Web require a more intuitive query beyond spatial or keyword queries.

2.2 Content-based Retrieval

Query by example (or content-based retrieval) has been used in many information retrieval research areas, such as image, music, and geographic search [29, 24, 10, 11]. The Multimedia Analysis and Retrieval System (MARS) [22, 23] is an image search system developed by Rui et al. and has a relevance feedback system, which takes into account the variability of users' notion of similarity by calculating the variance of each feature dimension of selected positive examples. Ishikawa et al. proposed a theoretical method for predicting which attributes are important, as well as which correlations of attributes are important in the user's notion of similarity (MindReader [14].) Ashwin et al. [8] indicated that if only positive examples are used for similarity prediction, such as with MARS and MindReader, items close to non-relevant ones continue to be retrieved. They proposed a robust technique for incorporating non-relevant items to determine the feasible search region. A decision surface is determined so as to split the attribute space into relevant and non-relevant regions. The differences between our method and the two by Ishikawa et al. and Ashwin et al are discussed in Section 4.2.

Nakajima et al. proposed a relative query for image search [21], which is similar to the query by example method we propose. A relative query consists of selected examples in a domain users know well to search for information in an unknown domain. They calculate the similarity between input queries and data to be found using the *relativeness* in a well-known domain. The relative query uses a novel feature, the relativeness of selected examples in a set of examples; however, it does not focus on the similarity varying with individuals.

3. MODEL

A query by example model for geographic entity search is presented in this section. We first define data representation for geographic entities. Second, we discuss a search model for our proposed query by example following the definition of information retrieval.

3.1 Data Representation

Geographic entities have several attributes, including position, and each attribute value is represented as a point in k-dimensional space.

Entity schema $\boldsymbol{R}$ is defined by a set of attributes

$$\boldsymbol{R} = \{a_1, a_2, \ldots, a_M, \mathrm{pos}\}, \tag{1}$$

where a_i is an attribute, such as names and descriptions, and pos is a position represented by latitude and longitude and differentiates itself from ordinary entities. Entity schema $\boldsymbol{R}$ is pre-defined for each class of geographic entities, e.g., restaurants, hotels, and landmarks.

An entity $\mathbf{e}$ for an entity schema $\boldsymbol{R}$ is a point in N-dimensional space.

$$\mathbf{e} = (\mathbf{e}_{a_1}, \mathbf{e}_{a_2}, \ldots, \mathbf{e}_{a_M}, \mathbf{e}_{\mathrm{pos}}), \tag{2}$$

where $N = \sum_{i=1}^{M} n_i + n_{\mathrm{pos}}$, and n_i is the number of dimensions for $\mathbf{e}_i$, which is an attribute value for a_i. $\mathbf{e}_i$ can represent a scalar, such as budget for restaurants, or descriptions as a term frequency-inverse document frequency (tf-idf) vector. $\mathbf{e}_{\mathrm{pos}}$ is a position vector containing latitude and longitude, and the number of dimension is $n_{\mathrm{pos}} = 2$.

3.2 Search Model

Query by example for geographic entity search enables users to select examples in a known place, and retrieves and ranks entities in an unknown place based on their similarity. Users can also choose the known place and the unknown one, which is called *source domain* $\boldsymbol{E}_s$, and *target domain* $\boldsymbol{E}_t$.

The two domains are subsets of all the entities $\boldsymbol{E}$:

$$\boldsymbol{E}_s \subset \boldsymbol{E},\ \boldsymbol{E}_t \subset \boldsymbol{E}, \tag{3}$$

They are generally $\boldsymbol{E}_s \cap \boldsymbol{E}_t = \phi$, that is, the source and target domains are coprime because the concepts *known* and *unknown* are also coprime.

From the source domain $\boldsymbol{E}_s$, users can select a subset as a query to search for entities in the target domain $\boldsymbol{E}_t$. Consequently, a set of queries $\boldsymbol{Q}$, and data $\boldsymbol{D}$ to be retrieved in query by example for geographic entity search are defined as follows:

$$\boldsymbol{Q} = \mathfrak{P}(\boldsymbol{E}_s),\ \boldsymbol{D} = \boldsymbol{E}_t, \tag{4}$$

where $\mathfrak{P}(x)$ is a power set of x.

To follow the definition of information retrieval, queries $\boldsymbol{Q}$ and data $\boldsymbol{D}$, and also a function Rank are to be defined. Rank is a function from queries $\boldsymbol{Q}$ and data $\boldsymbol{D}$ to real number $\mathbb{R}$, i.e., $\mathrm{Rank} : \boldsymbol{Q} \times \boldsymbol{D} \to \mathbb{R}$.

The function determines results to be retrieved and their ranks for a query. Query by example for geographic entity search ranks entities based on the similarity between a query $\boldsymbol{Q}_i \subset \boldsymbol{Q}$ and data. Accordingly, we should consider the similarity as $\mathrm{Rank}(\boldsymbol{Q}_i, \mathbf{d}_j)$ for data $\mathbf{d}_j \in \boldsymbol{D}$. The similarity can be regarded as an inverted distance between them, and it is more convenient to compare our method with MindReader. Thus, the ranking function is defined as:

$$\mathrm{Rank}(\boldsymbol{Q}_i, \mathbf{d}_j) = \exp\left(-\mathrm{dist}(\boldsymbol{Q}_i, \mathbf{d}_j)\right). \tag{5}$$

The term $\mathrm{dist}(\boldsymbol{Q}_i, \mathbf{d}_j)$ is the distance between selected entities $\boldsymbol{Q}_i$ and data $\mathbf{d}_j$. The form $\exp(-x)$ just takes an inverse of the distance, and it normalizes the inverse distance into $[0, 1]$.

Table 1: Symbol usage.

Symbol	Definition
$\boldsymbol{E}$	all entities.
$\boldsymbol{E}_s$	source domain (a subset of $\boldsymbol{E}$.)
$\boldsymbol{E}_t$	target domain (a subset of $\boldsymbol{E}$.)
$\boldsymbol{Q}$	queries that are all the subsets of $\boldsymbol{E}_s$.
$\boldsymbol{Q}_i$	a query that is an element of $\boldsymbol{Q}$ (a set of entities.)
$\boldsymbol{D}$	entities to be retrieved ($= \boldsymbol{E}_t$).
$\mathbf{e}$	an entity in $\boldsymbol{E}$.
N	the number of dimension of an entity $\mathbf{e}$.
$\mathbf{q}_k$	an entity in $\boldsymbol{Q}_i$.
g_k	a goodness value for $\mathbf{q}_k$ (0/1 or multi-level.)
$\mathbf{d}_j$	an entity in $\boldsymbol{D}$.
$\mathrm{Rank}()$	a ranking function ($\boldsymbol{Q} \times \boldsymbol{D} \rightarrow \mathbb{R}$.)
$\mathbf{m}$	an ideal query point (a N-d vector.)
$\mathbf{W}$	a $N \times N$ matrix that gives a distance function.
$\mathrm{dist}()$	a distance function ($\mathfrak{P}(\boldsymbol{E}) \times \boldsymbol{E} \rightarrow \mathbb{R}$.)

4. PREDICTION OF USER'S NOTION OF SIMILARITY

We have obtained a search model of query by example for geographic entity search. Next, the *variable* distance function used in the search model should be discussed, which is a main contribution of this research. We present the variable distance function proposed in MindReader, reveal the limitation of the method, and propose an adaptation to geographic entity search. The symbols we use are listed in Table 1.

4.1 Formulation of MindReader

Generally, The Euclidean distance between two points is defined as $\mathrm{dist}(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y})}$, and equals $(\mathbf{x}-\mathbf{y})^T\mathbf{I}(\mathbf{x}-\mathbf{y})$ (root is omitted for convenience). By replacing the unit matrix with a diagonal matrix $\mathbf{D}$, it becomes a weighted Euclidean distance $(\mathbf{x}-\mathbf{y})^T\mathbf{D}(\mathbf{x}-\mathbf{y})$, i.e., where the differences in important dimensions are weighted more and ones of meaningless dimensions less. The weighted Euclidean distance can take account of the importance for each dimension or attribute. For example, price distance is important for a user, and style distance for another. In addition, using a symmetric matrix $\mathbf{W}$ instead of the diagonal matrix, the distance function becomes able to capture the correlation of importance for each dimension, e.g., both expensive and Japanese style restaurants are desirable. The symmetric matrix determines the distance function, and prediction of a user's notion of similarity leads to prediction of matrix $\mathbf{W}$.

In MindReader, the distance function between a query $\boldsymbol{Q}_i$ and data $\mathbf{d}_j$ is

$$\mathrm{dist}(\boldsymbol{Q}_i, \mathbf{d}_j) = (\mathbf{d}_j - \mathbf{m})^T\mathbf{W}(\mathbf{d}_j - \mathbf{m}), \tag{6}$$

where it is assumed that a user has an *ideal* query $\mathbf{m}$ and an expected distance corresponding to a symmetric matrix $\mathbf{W}$ in mind, and the problem is to predict $\mathbf{m}$ and $\mathbf{W}$ from the given set of examples $\boldsymbol{Q}_i$. Figure 2 illustrates this problem. The black circles are selected and given examples, and the problem is to estimate an ideal bound to retrieve relevant data, represented as white circles.

The basic idea of the prediction is to minimize the distance between selected examples $\boldsymbol{Q}_i$ and the ideal query $\mathbf{m}$. If a user had an ideal query and an ideal distance, examples selected by the user would be similar to the ideal query based on the distance. In Figure 2, the ideal query is represented as the star and is located at the center of the ideal bounds. According to the assumption, the prediction of the ideal query $\mathbf{m}$ and the matrix of the ideal distance $\mathbf{W}$ is formulated as a minimization problem

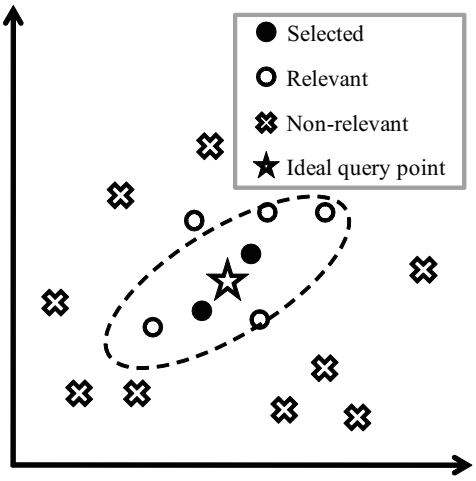

Figure 2: Prediction of user's notion of similarity by selected examples. Ideal bound is illustrated as dotted line, which is determined by matrix $\mathbf{W}$ in Equation 6.

$$\min_{\mathbf{m},\mathbf{W}} \sum_{\mathbf{q}_k \in \boldsymbol{Q}_i} g_k(\mathbf{q}_k - \mathbf{m})^T\mathbf{W}(\mathbf{q}_k - \mathbf{m}), \tag{7}$$

subject to the constrain

$$\det(\mathbf{W}) = 1, \tag{8}$$

where $\det(\mathbf{W})$ is the determinant of the matrix $\mathbf{W}$ (this constraint is not essential, but uniquely determines the matrix). The scalar g_k is a goodness value for selected examples, and the default value is 1 (it can be multi-level).

The problem was solved theoretically. The ideal vector $\mathbf{m}$ equals an average of selected examples (weighted by g_k), and the matrix of the ideal distance is proportional to an inverse covariance matrix (weighted by g_k.)

In addition, using MindReader proved that a method used for weighting dimensions in MARS [22] is a special case of Equation 6, where matrix $\mathbf{W}$ is a diagonal matrix and the goodness value $g_k = 1$. A matrix for weighting each dimension in MARS is computed as the inverse of the variance of each feature among all relevant examples, that is, $w_{ii} = \frac{1}{\sigma_i^2}$.

4.2 Our Approach

A theoretical solution to the prediction of users' notions of similarity was given by Ishikawa et al. However, it cannot be applied directly to our distance function because there are two fundamental problems.

1. MindReader does not work well if only one or a few examples are given.
2. For entities, some attributes have ordinary weights for a distance measure, unlike image features.

Table 2: Categorization of Known and Unknown, Positive and Negative examples. Known and Negative examples are *implicit negative examples***.**

	Known	**Unknown**
Positive	Selected	Not Selected
Negative	Not Selected	Not Selected

First, it is generally difficult to predict which attributes are important for a user inputting few examples. However, the number of input elements, such as keywords for a search task, is too small. For content-based search in particular, or query by example, it cannot be assumed that many examples are selected as a query because of the difficulty of inputting them. At least, a robust method for predicting users' notions of similarity is required, and should work even for only one example.

Second, there is a common, or standard similarity for entities such as people and restaurants. For example, restaurants for $100 and for $10 meals that share other attributes in common should not be similar, and restaurants with different names but sharing the same features should be similar. It can be easily accepted that we have a standard similarity for each entity, and a similarity far from the standard cannot be acceptable to users. However, the previously proposed method ignores the standard similarity, especially when there are not enough examples given.

Therefore, we propose an improvement and an adaptation of the previously proposed method for geographic search. The basic ideas for solving the problems mentioned above are

1. *Implicit negative examples* in a selection of geographic entities are used to support prediction of users' notions of similarity.
2. A similarity far from the pre-defined standard similarity is given a penalty.

In the following sections, the proposed two approaches are explained.

4.2.1 Implicit Negative Examples

The main problem of predicting users' notions of similarity is the small number of inputs by users. Therefore, we assume implicit negative examples, which are not explicitly selected as negative examples, but are assumed to be irrelevant to users' needs from their behavior. Additional inputs, not only selected positive examples, but implicit negative examples make the prediction more robust.

In query by example for geographic entity search, users select only positive examples to search for entities in a map. Non-selected examples are categorized into two types, examples unknown to the users or known negative ones, as summarized in Table 2. Naturally, users select known and positive examples, and the others are not. It is impossible to use unknown examples to predict users' notions of similarity because even users do not know whether they are relevant. Implicit negative examples are exactly the known and negative examples that are not selected because they are not relevant to users' intentions.

Particularly for geographic entity search, implicit negative examples are predictable. Examples are shown in Figure 3. The black circles are selected examples as a query, and they are obviously known and positive examples. The white circles and crosses are unknown examples, and they are ignored in this discussion. The goal is to find known and negative examples that are not selected.

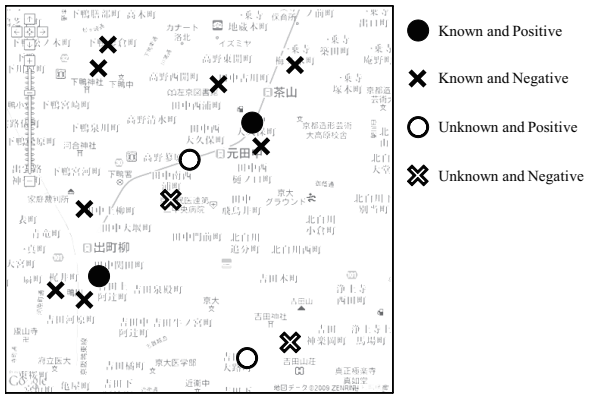

Figure 3: Implicit Negative Examples

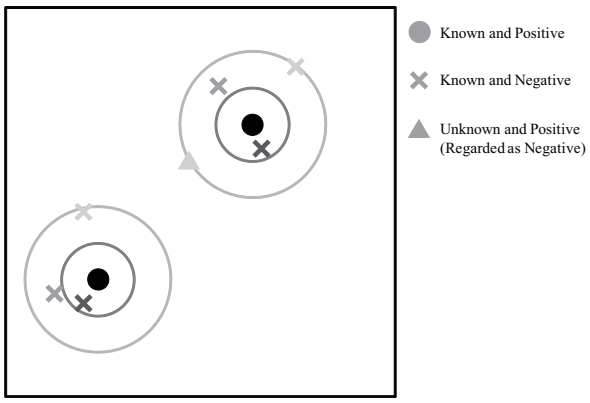

Figure 4: Prediction of Known Examples. Dark shade means high probability, and light means low.

Therefore, we propose a simple assumption on selection of positive examples in a query by example system.

ASSUMPTION 1. *Examples geographically close to selected positive examples are not selected intentionally despite the fact that they are known, that is, they are implicit negative examples.*

Near the selected positive examples in Figure 3, there are few positive examples because such known and positive examples must have been selected, and examples around known examples (in this case, the selected positive examples) may be known. This assumption highly depends on the input system, and is also strongly related to the area of a known place, which is a characteristic feature of a geographic entity. It is rare that only one geographic entity is known in a well known place, but some of the entities are known in that area. In addition, in selecting examples, positive examples close to selected examples may not be missed since positive ones are recognized at that time. (Note that this case is not general, but specific to our proposed method.) Although this assumption seems to be specific to geographic entities, the idea of implicit feedback has been widely used and evaluated in information retrieval research [16, 28].

Therefore, we have to calculate at what probability non-selected examples are implicit negative examples. Intuitively, examples closer to selected examples are known at higher probability. Thus, 2-D Gaussian mixture models are used to predict a range of known examples. Given a set of positive examples $\boldsymbol{Q}_i$, the probability

$P(\mathbf{e}|\boldsymbol{Q}_i)$ that an example $\mathbf{e}$ is known is defined simply as the following.

$$P(\mathbf{e}|\boldsymbol{Q}_i) = \sum_{\mathbf{q}_k \in \boldsymbol{Q}_i} \frac{g_k}{N_g} \mathcal{N}(\mathbf{e}_{\rm pos}|\mathbf{q}_{k,\rm pos}, \boldsymbol{\Sigma}), \quad (9)$$

where $N_g = \sum_{\mathbf{q}_k \in \boldsymbol{Q}_i} g_k$, a sum of g_k, which is a goodness value for a selected example $\mathbf{q}_k$. 2-D vectors $\mathbf{e}_{\rm pos}$ and $\mathbf{q}_{k,\rm pos}$ are positions containing latitude and longitude. The probability $P(\mathbf{e}|\boldsymbol{Q}_i)$ is a mixture of Gaussian distributions with a fixed variance whose means are positions of selected examples, and each Gaussian is weighted by the goodness value for a selected example. It is obviously proven that $P(\mathbf{e}|\boldsymbol{Q}_i)$ is a probability since g_k/N_g is also a probability. Figure 4 shows the probability of known examples. The dark shade means high probability. In this example, an unknown and positive example is also regarded as a negative one at a low probability. However, if the assumption was reasonable, the benefit of the gain of inputs would have cancel the missed prediction.

Next, Equation 7 must be adapted for the obtained negative examples. The easiest way to use negative examples is to replace the goodness value g_k with the probability $-P(\mathbf{e}|\boldsymbol{Q}_i)$ for non-selected examples. Unfortunately, a negative goodness value does not work well with MindReader, and negative examples were not discussed well in the research. For an example of data with one dimension, suppose that a user selects 0.9, 1, and 1.1 as a positive example, and 100 as a negative example for retrieving data close to 1. According to Equation 7, the ideal vector $\mathbf{m}$ (scalar in this case) would be much less than 1 since $\mathbf{m}$ is the average of 0.9, 1, 1.1, and 100 weighted by their goodness values. Using this ideal vector, much lower values would be retrieved at the top. In addition, a negative distance can be allowed by using a negative goodness value.

As mentioned in Section 2.2, Ashwin et al. [8] proposed a method for estimating a decision surface using negative examples. However, in our proposed method and also content-based retrieval, negative examples are not given explicitly. Thus, such a method for clearly splitting positive and negative examples is not acceptable for implicit negative examples.

Therefore, we modify Equation 7 for negative examples, and a new minimization problem is

$$\min_{\mathbf{W}} \sum_{\mathbf{e}_k \in \boldsymbol{E}_s} v_k (\mathbf{e}_k - \mathbf{m})^T \mathbf{W} (\mathbf{e}_k - \mathbf{m}) \quad (10)$$

subject to the constraint,

$$\|\mathbf{W}\| = 1,\ w_{ij} \geq 0, \quad (11)$$

where w_{ij} is an i-j element of matrix $\mathbf{W}$, v_k is a goodness value for selected examples, and a negative probability with a parameter α for non-selected examples,

$$v_k = \begin{cases} g_k & e_k \in \boldsymbol{Q}_i \\ -\alpha P(\mathbf{e}|\boldsymbol{Q}_i) & \text{else} \end{cases} \quad (12)$$

(13)

and $\mathbf{m}$ is an average of $\boldsymbol{Q}_i$ (weighted by g_k,)

$$\mathbf{m} = \frac{1}{N_g} \sum_{\mathbf{q}_k \in \boldsymbol{Q}_i} g_k \mathbf{q}_k. \quad (14)$$

The major modifications can be seen in the range of summation, the ideal vector, and the constraint for matrix $\mathbf{W}$. First, we take a sum for all the examples including non-selected examples with negative values $-\alpha P(\mathbf{e}|\boldsymbol{Q}_i)$, and parameter α depends on the number of examples. Second, the ideal vector is fixed to the average of selected examples, which is no longer a variable in the minimization, but the same as in MindReader. Finally, a little-changed constraint is given to prevent a negative distance function and excess values in matrix $\mathbf{W}$ (this constraint is not yet essential.)

4.2.2 *Penalty for Over-fitting Distance Function*

So far, attributes of entities are treated equally; however, in judging the distance between entities, some attributes are important, whereas others are meaningless, unlike image features. Thus, we pre-define a standard distance, and an extraordinary distance obtained with the prediction is given a penalty. This kind of penalty can be seen in machine learning to prevent an over-fitting of parameters.

Letting $\hat{\mathbf{W}}$ be a matrix for a standard distance, we finally obtain a formalization of the prediction of users' notions of similarity.

$$\min_{\mathbf{W}} \sum_{\mathbf{e}_k \in \boldsymbol{E}_s} v_k (\mathbf{e}_k - \mathbf{m})^T \mathbf{W} (\mathbf{e}_k - \mathbf{m}) + \frac{\rho}{2} \|\mathbf{W} - \hat{\mathbf{W}}\|^2 \quad (15)$$

subject to Equation 11. The term $\|\mathbf{W} - \hat{\mathbf{W}}\|^2$ is a penalty for distance matrix $\mathbf{W}$ far from the standard $\hat{\mathbf{W}}$, and the penalty makes the distance matrix close to the standard.

Finally, we have an optimal distance matrix $\mathbf{W}^*$, and the ranking function for a query $\boldsymbol{Q}_i$ and data $\mathbf{d}_j$ (Equation 5) reduces to

$$\begin{aligned} \text{Rank}(\boldsymbol{Q}_i, \mathbf{d}_j) &= \exp\left(-\text{dist}(\boldsymbol{Q}_i, \mathbf{d}_j)\right) \\ &= \exp\left(-(\mathbf{d}_j - \mathbf{m})^T \mathbf{W}^* (\mathbf{d}_j - \mathbf{m})\right) \end{aligned} \quad (16)$$

The decision to use certain parameters, i.e., $\boldsymbol{\Sigma}$ for a covariance matrix of 2-D Gaussian distribution, α for a weight of implicit negative examples, and ρ for a weight of a penalty is discussed in Section 5. We explain how to make the standard distance matrix $\hat{\mathbf{W}}$, and how to solve the optimization problem (Equation 15) under the constraint (Equation 11) in Section 5.2.

5. EXPERIMENT

This section first describes our implementation of our query by example method for geographic entity search and explains the solution of the distance matrix discussed in Section 4. We then discuss the set up of a test set for our experiment, present metrics we used and baseline methods, and show the experimental results.

5.1 Implementation

We developed an implementation of our proposed query by example method for geographic entity search. The geographic entities we used were obtained from GourNavi, which is a service for introducing Japanese restaurants. A map interface was implemented with Google Maps API [1], which allows users to select examples in a known area to search for restaurants in an unknown area.

Restaurant entities were obtained through the GourNavi Web service[2]. The number of retrieved entities was 46,945, and were stored and indexed by latitude and longitude. Table 3 lists examples of available attributes for restaurants in GourNavi.

There are various types of attributes, text, float, integer, and set; however, some attributes are insignificant for entity similarity. We

[1]http://www.google.com/apis/maps/

[2]http://api.gnavi.co.jp/

Table 3: Examples of attributes for restaurants in GourNavi

Attribute name	Type	Example of value
name	Text	Japanese bar: *Sushi*
latitude	Float	35.025764
longitude	Float	135.78125
category	Text	Japanese sushi bar
url	Text	http://sushi.jp/
address	Text	Yoshida, Sakyo, Kyoto, Japan
open time	Time	17:00
holiday	Day	Wed
introduction	Text	Various kinds of Japanese food.
category label	Set	{bar, sushi, Japanese food}
budget	Integer	$30
equipment	Set	{private room, car park}

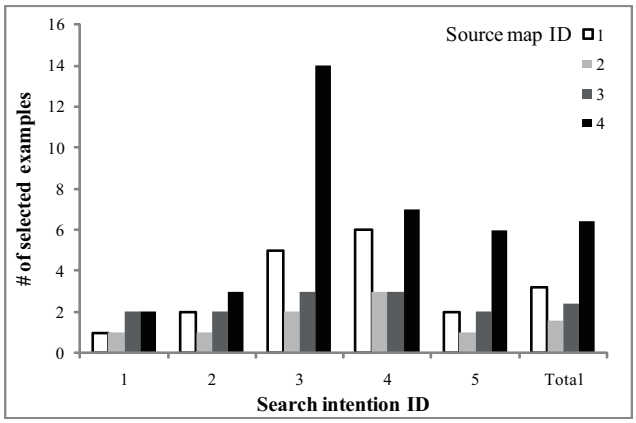

Figure 5: Number of selected examples as query.

only used five attributes for measuring similarity between entities, names, categories, category labels, introductions, and budgets.

The standard tf-idf method was used for a text attribute value. tf_{ij} is the number of the i_{th} term in j_{th} text, df_i is the number of texts including the i_{th} term, and $idf_i = \log(L/df_i)$. L is the number of the all the texts. Generally, the similarity between documents is defined as the cosine of two document vectors, that is, $\cos(\mathbf{v}, \mathbf{u}) = \langle \mathbf{v}, \mathbf{u} \rangle / \|\mathbf{v}\| \|\mathbf{u}\|$, where $\mathbf{v}$ and $\mathbf{u}$ are vectors consisting of the products of tf_{ij} and idf_i. The Euclidean distance was used in the proposed distance function instead of the cosine similarity, and the Euclidean distance does now work well for measuring the similarity in high dimensional space such as a document similarity [25]. However, by normalizing the vectors $\mathbf{v}$ and $\mathbf{u}$ into 1, the squared Euclidean distance equals $2 - 2\cos(\mathbf{v}, \mathbf{u})$, and is monotonically decreasing for the cosine. Therefore, the Euclidean distance of normalized vectors is almost equivalent to the cosine for our use.

For set attributes, the attribute values have a set of elements, as can be seen in the attribute *category label* in Table 3. The bag of elements can be treated as a bag of words for the representation of documents. In this implementation, set attributes were considered as text ones, and their distances were calculated.

These attribute values are very sparse and are represented as points in a high dimensional space. For example, 27,212 dimensions are allocated for text attribute values (the number of unique terms in the name, category, and introduction,) and 158 for category labels. Many attributes require many parameters, and it is difficult to suitably estimate a large number of parameters because of few input examples. Thus, we apply *latent semantic analysis* to compress their dimensions into 50 for text and 20 for category labels.

The budget attribute was normalized so that the maximum distance is 2, which is the same as the others. The schema is $\boldsymbol{R} = \{\text{text}, \text{category\_label}, \text{budget}, \text{pos}\}$ as defined in Section 3, and the number of dimensions for an entity is $50 + 20 + 1 + 2 = 73$ (the position is ignored for calculating similarity.)

5.2 Estimation of Distance Matrix

The optimization problem (Equation 15) under the constraint (Equation 11) cannot be solved analytically, but can be dissolved numerically in the context of semi-definite programming [9]. However, a large number of parameters in the symmetric matrix for the distance function requires many calculations, it would be difficult to return results on-demand. Therefore, the symmetric matrix is limited to a diagonal matrix to reduce the number of dimensions and calculations in this implementation. The limitation makes the optimization problem a convex quadratic programming problem, and the number of parameters to be estimated is the same as the dimensions of an entity.

To estimate the distance matrix, $\hat{\mathbf{W}}$ must be determined as a standard distance matrix for $\mathbf{W}$. We manually determined a 5-scaled distance (1 to 5) for 275 pairs of restaurants, and determined the standard distance matrix $\hat{\mathbf{W}}$ by using support vector machine (SVM) regression with additional constrained conditions, i.e., $\|\hat{\mathbf{W}}\| = 1$, and $\hat{w}_{ij} \geq 0$ $(i, j = 1, 2, \ldots, N)$ [27].

5.3 Test Set

Four volunteers manually created a test set for performance evaluation. The test set consisted of search intentions, queries, and data with relevance score. The search intentions, source maps (where users select examples as a query), and target maps (where examples are retrieved) are listed in Table 4. Search intention 1 is that the absolute price is important in calculating similarity, 2 and 3 is the style and food served, 4 is the relative price, and 5 is that the combination of food and budget is important. The source and target maps were selected from major cities in Japan.

The test set was created as follows. We first showed the volunteers four source maps; one to each volunteer, asked them to browse restaurants in the maps for 5 minutes, and to select examples for each intention as a query for 2 minutes. We obtained five queries for each map, that is, a total of 20 types of queries. These selection time limits were set for practical use. Without these time limits, all the restaurants might have been checked, and all the positive examples might have been selected as a query, which is not real user behavior. Figure 5 shows the statistics on the number of selected examples as a query for each target map and question. 3.4 examples were selected on average. The goodness values were fixed at 1 for all the selected examples.

Next, the volunteers were divided into two groups, and two target maps were allocated; one to each group. They were asked to browse restaurants on the maps for 5 minutes, and to evaluate restaurants on a 5-point scale for each search intention. Table 4 also lists the average Kappa coefficient for the evaluation, which is the degree of agreement by two volunteers. The scores are higher than 0.80, which indicates almost perfect agreement on the evaluation.

Now, we have 20 queries and two data sets to be retrieved. In our experiment, the combinations, i.e., 40 tests, were tried with each method described in the next section.

5.4 Settings

Table 4: Search intentions, source maps, and target maps.

Search intentions	
ID	Content
1	Restaurants for around 1,000 yen.
2	Restaurants serving spicy food.
3	Restaurants serving sea food.
4	Expensive restaurants
5	Restaurants serving special local foods at modestly high prices

Source maps		
ID	Area	# of entities
1	Tokyo	55
2	Nagoya	55
3	Osaka	59
4	Sapporo	51

Target maps			
ID	Area	# of entities	Avg of Kappa
1	Kyoto	49	0.94
2	Kobe	50	0.86

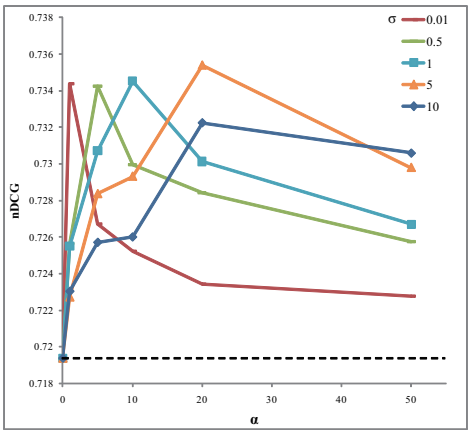

Figure 6: nDCG for the proposed method.

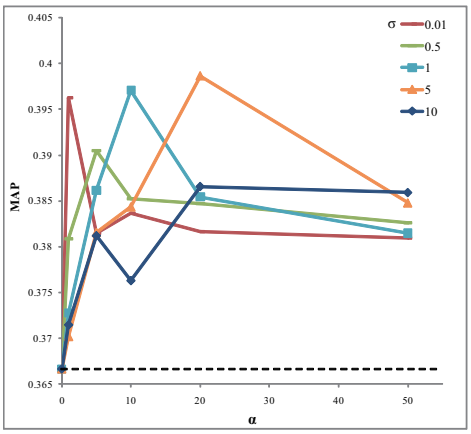

Figure 7: MAP for the proposed method.

To validate the effectiveness of our method, baseline methods were used for comparison. The first baseline was a simple method with a fixed distance function, which uses the standard matrix $\hat{\mathbf{W}}$ to calculate similarity (**Baseline**.) The second baseline was MindReader, explained in Section 3 (**MindReader**.) In this experiment, the distance matrix of MindReader was limited to a diagonal matrix, and the elements in the matrix were given by inverse variances (weighted by goodness values). This method cannot work with only one example given, so a unit matrix was assigned to the distance matrix, as mentioned by Ishikawa et al. Our proposed method is represented by **Proposed**.

The metrics we used were mean average precision (MAP) and normalized discounted cumulative gain (nDCG). The highest grade of 5 is regarded as relevant for MAP since it is a method for data with binary relevance.

5.5 Experimental Results

Some parameters were determined by comparing each value before the main experiment; a covariance matrix $\mathbf{\Sigma}$ for 2-D Gaussian distribution, a weight α for implicit negative examples, and a weight ρ for the penalty. As the two parameters, $\mathbf{\Sigma}$ and α, are related to each other, the combinations were tried to decide them. The covariance matrix is a diagonal 2×2 matrix, and proportional to a unit matrix, i.e., $\mathbf{\Sigma} = \sigma\mathbf{I}$. This is because a range of a known area is assumed a circle, not an ellipse.

The results of the preliminary experiment for σ and α are shown in Figure 6 and 7, which describe nDCG and MAP for the proposed method. The results with $\alpha = 0$ represents results of no weight for implicit negative feedback (illustrated as dotted line,) and the improvement by using the implicit negative feedback was confirmed. The local optimal combination is $\sigma = 5.0$ and $\alpha = 20.0$. However, the value of σ is not reasonable for the proposed method of finding implicit negative feedback. The value 5.0 is too large for the real world, and it gives almost the same values to entities within 1 km square area. It means that all the non-selected examples are treated as implicit negative examples. Therefore, we confirmed the effectiveness of implicit negative feedback in the query by example method, however, could not verify the validity of the method for estimating implicit negative feedback. One of the reason why σ was estimated as so a large value is that almost all of the restaurants relevant to given search intentions are selected as a query in creating the test set. It might cause the large value to be the best for the parameter σ.

The results of each value for the weight ρ is shown in Figure 8 and 9, where $\sigma = 5.0$ and $\alpha = 20.0$. $\rho = 0$ represents results of no weight for the penalty, and the proposed method with enough large ρ is almost the same as the baseline method. The local optimal value for the parameter ρ is 1.0. The results indicate that neither a method with only implicit negative feedback, nor one with only the standard similarity is not effective for estimating user's notion of similarity.

The main experimental result is shown in Figure 10 and 11. Our proposed method got the highest nDCG and MAP scores in the three methods, whereas MindReader was the worst method in our experiment. This was because the small number of given examples made it difficult to predict the similarity adapted to each search intention. The baseline method got a little lower scores than the proposed one, and the result supports that the standard similarity is required for entity similarity.

We could see some characteristics for each search intention. For the search intention 1, MindReader worked better, and emphasized

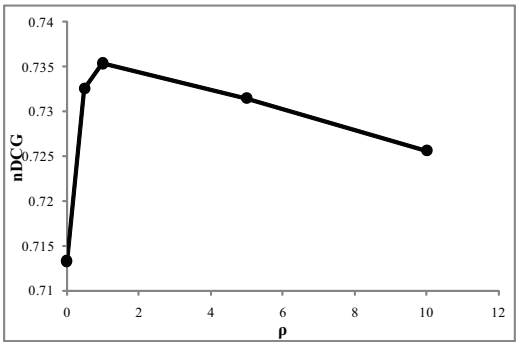

Figure 8: nDCG of each ρ for the proposed method.

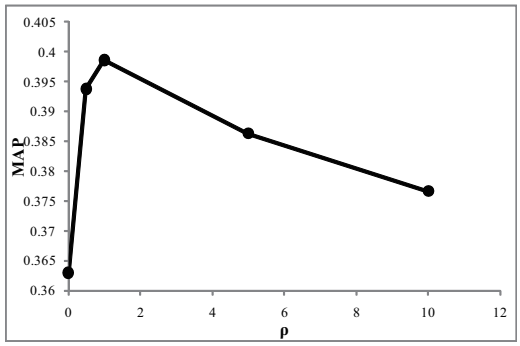

Figure 9: MAP of each ρ for the proposed method.

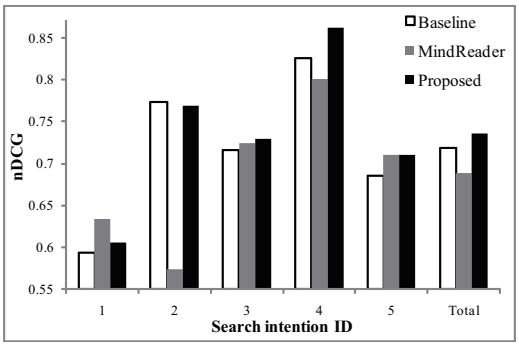

Figure 10: nDCG for the proposed method and the baselines.

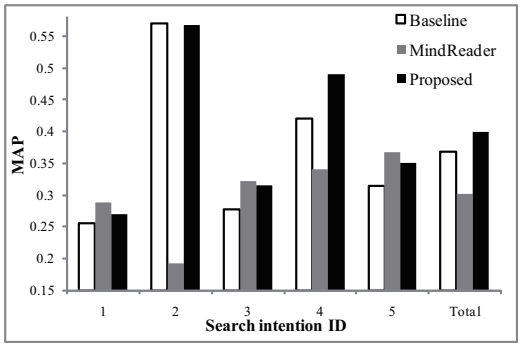

Figure 11: MAP for the proposed method and the baselines.

the price similarity. However, for the search intention 4, where the price similarity should be weighted, MindReader failed to achieve high scores, but our proposed method could regarded the price as an important attribute for calculating the similarity. The proposed method is also robust to a search with relative similarity because of the following reason. For retrieving expensive restaurants, a user selects restaurant for relatively high price in a map, and budgets of the selected restaurants are not always close to each other, for example, restaurants whose budgets are more than $100. In this case, the previously proposed method, MindReader neglects the price similarity since it takes account into only selected examples. On the contrary, our method makes use of not only selected examples, but also non-selected examples, and the difference between selected examples and non-selected ones is used to estimate the similarity. Thus, in our method, the price similarity is weighted so as to make the non-selected examples far from selected examples.

For the search intention 2, the standard method and our proposed method got much higher scores than MindReader. Serving spicy food is a noticeable feature for restaurants, and the standard similarity performs well for a previously predictable similarity.

6. CONCLUSION

We proposed a method of searching for geographic entities in an unknown place with a query by example in a known place. The proposed method enables users to use analogies for finding relevant entities in unfamiliar places. Even if a user does not have any knowledge about a place where he/she wants to find information, he/she can make a query by selecting relevant examples in a well known place. We raised two problems for calculating similarity in the proposed method; the notion of similarity can vary for different contexts, and accurately measuring similarity between entities in different domains. We tackled the problem of the variability of similarity with the idea of implicit negative feedback, and improved previously proposed method using only positive examples. The experimental result showed the effectiveness of the implicit negative feedback. However, we could not verify the validity of the method for predicting implicit negative examples in a map interface.

There are still some problems remaining in the query by example method for geographic entity search. We plan to address the problem of similarity in different domains by findings from a study of analogy.

7. ACKNOWLEDGMENTS

This work was supported in part by the following projects and institutions: Grants-in-Aid for Scientific Research (Nos. 18049041, 21700105 and 21700106) from MEXT of Japan, a Kyoto University GCOE Program entitled “Informatics Education and Research for Knowledge-Circulating Society,” and the National Institute of Information and Communications Technology, Japan.

8. REFERENCES

[1] Bing maps. http://www.bing.com/maps/.
[2] Booking.com. http://www.booking.com/.
[3] Google maps. http://maps.google.com/.
[4] GourNavi. http://www.gnavi.co.jp/.
[5] Yahoo! Local Maps. http://maps.yahoo.com/.
[6] E. Amitay, N. Har’El, R. Sivan, and A. Soffer. Web-a-where: geotagging web content. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2004)*, pages 273–280, 2004.

[7] D. Angluin and C. Smith. Inductive inference: Theory and methods. *ACM Computing Surveys (CSUR)*, 15(3):237–269, 1983.

[8] T. V. Ashwin, R. Gupta, and S. Ghosal. Adaptable similarity search using non-relevant information. In *Proceedings of the 28th international conference on Very Large Data Bases (VLDB 2002)*, pages 47–58, 2002.

[9] A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization*. Society for Industrial and Applied Mathematics, 2001.

[10] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.

[11] N. Chang and K. Fu. Query-by-pictorial-example. *IEEE Transactions on Software Engineering*, 6:519–524, 1980.

[12] Y. Chen, T. Suel, and A. Markowetz. Efficient query processing in geographic web search engines. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data (SIGMOD 2006)*, pages 277–288, 2006.

[13] R. Hiramoto and K. Sumiya. Web information retrieval based on user operation on digital maps. In *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems (GIS 2006)*, pages 99–106, 2006.

[14] Y. Ishikawa, R. Subramanya, and C. Faloutsos. Mindreader: Querying databases through multiple examples. In *Proceedings of the International Conference on Very Large Data Bases (VLDB 1998)*, pages 218–227, 1998.

[15] M. P. Kato, H. Ohshima, S. Oyama, and K. Tanaka. Query by analogical example: Relational search using web search engine indices. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009)*, pages 27–36, 2009.

[16] D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. In *ACM SIGIR Forum*, volume 37, pages 18–28, 2003.

[17] R. R. Larson. Geographic information retrieval and spatial browsing. *GIS and Libraries: Patrons, Maps and Spatial Information*, pages 81–124, 1996.

[18] M. Lieberman, H. Samet, J. Sankaranarayanan, and J. Sperling. STEWARD: architecture of a spatio-textual search engine. In *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems (GIS 2007)*, 2007.

[19] A. Markowetz, Y. Chen, T. Suel, X. Long, and B. Seeger. Design and implementation of a geographic search engine. In *Proceedings of the 8th International Workshop on the Web and Databases (WebDB 2005)*, pages 19–24, 2005.

[20] K. McCurley. Geospatial mapping and navigation of the web. In *Proceedings of the 10th international conference on World Wide Web (WWW 2001)*, pages 221–229, 2001.

[21] S. Nakajima and K. Tanaka. Relative queries and the relative cluster-mapping method. In *Proceedings of the 9th International Conference on Database Systems for Advances Applications (DASFAA 2004)*, pages 843–856, 2004.

[22] Y. Rui, T. Huang, and S. Mehrotra. Content-based image retrieval with relevance feedback in MARS. In *Proceedings of IEEE International Conference on Image Processing*, volume 81, pages 815–818, 1997.

[23] Y. Rui, T. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Transactions on circuits and systems for video technology*, 8(5):644–655, 1998.

[24] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on pattern analysis and machine intelligence*, 22(12):1349–1380, 2000.

[25] A. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *Proceedings of the AAAI Workshop on Artificial Intelligence for Web Search*, pages 58–64, 2000.

[26] A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.

[27] V. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, 1995.

[28] R. White, I. Ruthven, and J. Jose. A study of factors affecting the utility of implicit relevance feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2005)*, pages 35–42, 2005.

[29] A. Yoshitaka and T. Ichikawa. A survey on content-based retrieval for multimedia databases. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):81–93, 1999.