Content-based Retrieval for Heterogeneous Domains: Domain Adaptation by Relative Aggregation Points

Makoto P. Kato  Hiroaki Ohshima  Katsumi Tanaka

Kyoto University, Japan
{kato,ohshima,tanaka}@dl.kuis.kyoto-u.ac.jp

ABSTRACT
We introduce the problem of domain adaptation for content-based retrieval and propose a domain adaptation method based on relative aggregation points (RAPs). Content-based retrieval including image retrieval and spoken document retrieval enables a user to input examples as a query, and retrieves relevant data based on the similarity to the examples. However, input examples and relevant data can be dissimilar, especially when domains from which the user selects examples and from which the system retrieves data are different. In content-based geographic object retrieval, for example, suppose that a user who lives in Beijing visits Kyoto, Japan, and wants to search for relatively inexpensive restaurants serving popular local dishes by means of a content-based retrieval system. Since such restaurants in Beijing and Kyoto are dissimilar due to the difference in the average cost and areas' popular dishes, it is difficult to find relevant restaurants in Kyoto based on examples selected in Beijing. We propose a solution for this problem by assuming that RAPs in different domains correspond, which may be dissimilar but play the same role. A RAP is defined as the expectation of instances in a domain that are classified into a certain class, e.g. the most expensive restaurant, average restaurant, and restaurant serving the most popular dishes. Our proposed method constructs a new feature space based on RAPs estimated in each domain and bridges the domain difference for improving content-based retrieval in heterogeneous domains. To verify the effectiveness of our proposed method, we evaluated various methods with a test collection developed for content-based geographic object retrieval. Experimental results show that our proposed method achieved significant improvements over baseline methods. Moreover, we observed that the search performance of content-based retrieval in heterogeneous domains was significantly lower than that in homogeneous domains. This finding suggests that relevant data for the same search intent depend on the search context, that is, the location where the user searches and the domain from which the system retrieves data.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms
Algorithms, Experimentation

Keywords
Content-based retrieval, domain adaptation

1. INTRODUCTION
Content-based retrieval enables a user to input examples as a query, and retrieves relevant data based on the similarity to the examples. In multimedia retrieval, it is difficult to represent complex search intents by using keywords due to the difference in medium between input and output; thus, content-based retrieval has been successfully applied to a wide range of multimedia retrieval including image retrieval [17] and spoken document retrieval [9], as well as geographic object retrieval [15].

Content-based geographic object retrieval proposed by Kato et al. [15] provides a means by which a user can select geographic objects (e.g. restaurants) as a query in a domain s/he knows well (denoted by source domain $D^{(S)}$) for retrieving objects from a domain s/he does not know well (denoted by target domain $D^{(T)}$). This search method is helpful, especially when the searcher has little knowledge of the target domain, as it is usually difficult to make a concrete query without knowledge on the domain in which the user wants to search [2]. Suppose that a user who lives in Beijing visits Kyoto, Japan, and wants to search for restaurants, say, relatively inexpensive restaurants serving popular local dishes. If the user has little knowledge of Kyoto and no idea about the average cost and what the popular local dishes are there, it might be difficult for him/her to specify attributes (e.g. a category and price range of restaurants), or make a keyword query that represents his/her search intent. By using a content-based geographic object retrieval system, a user is able to make an example query without knowledge of the target domain by selecting familiar source domain examples that are relevant to his/her search intent.

A challenge in content-based geographic object retrieval is to find relevant instances from a target domain $D^{(T)}$ based on examples selected from a heterogeneous source domain $D^{(S)}$. When the two domains are dissimilar, examples selected from $D^{(S)}$ are not always similar to relevant instances from $D^{(T)}$. For example, assume that the user who visits Kyoto selects relatively inexpensive restaurants serving popular local dishes in Beijing. Such restaurants in Beijing and Kyoto are dissimilar since the average cost and popular local dishes are different in the two domains. Therefore, the content-based geographic object retrieval system has to find instances semantically similar to the given examples against superficial dissimilarity between instances from heterogeneous domains. Although similar problems have been recognized as domain
adaptation [20], domain adaptation for content-based retrieval has not been addressed in the literature. We discuss related work on domain adaptation in Section 2, and formalize and characterize domain adaptation for content-based retrieval in Section 3.

We propose a method for accurately predicting relevant instances in a domain based on selected examples in another heterogeneous domain. We assume that different domains include corresponding points that may not be the same in terms of their representation but play the same role in heterogeneous domains. For the Kyoto and Beijing example, both domains include the most expensive restaurant, the average restaurant, and a restaurant serving the most popular dishes there. These kinds of points are identified through aggregation based on relative and domain-independent operation; we call these points relative aggregation points (RAPs). We assume that RAPs in different domains correspond, and construct a new feature representation for each instance based on RAPs from the source and target domains. Several kinds of similarities are measured between RAPs and instances, each of which is used to represent instances in the new feature space. Incorporating features represented by RAPs bridges the gap between heterogeneous domains and enables us to capture the place of an instance in a domain.

Our experiment was designed to provide answers to two research questions on domain adaptation for content-based retrieval. The first research question is whether domain adaptation for content-based retrieval is necessary. In other words, are relevant data different for the same search intent in different locations and datasets? We answered this research question based on a test collection developed for content-based geographic object retrieval, which includes relevant data for 100 queries (20 search intents in five domains). Our experimental result showed that mean average precision (MAP) and normalized discounted cumulative gain (nDCG) [12] in out-domain settings ($D(S) \neq D(T)$) were significantly lower than those in in-domain settings ($D(S) = D(T)$). This finding suggests that relevant data can vary in different search locations and datasets and has important implication for relevance judgment in information retrieval tasks. The second research question is whether domain adaptation based on RAPs can bridge the domain difference and significantly improve the search performance of content-based retrieval in heterogeneous domains. In our experiment based on the aforementioned test collection, our proposed method showed significant improvements on MAP and nDCG over baseline methods.

The remainder of this paper is structured as follows. We describe related work on domain adaptation in Section 2, and formalize and characterize domain adaptation for content-based retrieval in Section 3. In Section 4, we propose a method for domain adaptation based on RAPs. In Section 5, we describe the test collection developed for content-based geographic object retrieval and show experimental results in Section 6. We conclude with a summary of our findings and future work in Section 7.

2. RELATED WORK

Domain adaptation [20] has been addressed in several tasks such as part of speech tagging [5], text classification [4, 16, 27, 28], and learning to rank [7, 11, 26]. The motivation behind domain adaptation is that a model trained in a source domain $D(S)$ does not work well in a heterogeneous target domain $D(T)$. The aforementioned content-based retrieval problem is also a domain adaptation problem since it is a process for modeling a search intent from examples selected in a domain, and inferring relevant instances in another by using the model. In this section, we describe related work on domain adaptation and clarify our contributions of this paper.

Blitzer et al. proposed structural correspondence learning (SCL) for domain adaptation [4, 5]. SCL first extracts pivot features, which behave in the same way in different domains. The correlation with the pivot features is then used to identify correspondences among features from different domains. Each instance is represented by the correspondence together with its own features. SCL has been successfully applied to part of speech tagging and sentiment classification, and is considered a state-of-the-art method for domain adaptation. Since our proposed method is similar to SCL in terms of incorporating corresponding points not corresponding features, we included SCL as a baseline and compared it with our proposed method.

Several domain adaptation methods for document classification tasks have been proposed recently. Dai et al. proposed a domain adaptation method based on co-clustering [10]. The method clusters words and documents simultaneously in a target domain, restricting word clusters by labels from a source domain and transferring the labels to documents in the target domain. Ling et al. incorporated spectral clustering for domain adaptation to construct a new feature space based on the clustering results [16]. Spectral clustering is applied to documents in a source and target domain under the source domain supervision, so that documents in the target domain are separated as much as possible. Their method then represents documents by their vector that encodes clusters to which a document belongs. Xue et al. proposed a method based on probabilistic latent semantic analysis (pLSA) for domain adaptation [28]. They assumed that different domains include common latent topics, applied pLSA to documents under the source domain supervision, and classified documents from a target domain using common topics obtained through the pLSA. Wang et al. tackled a problem of cross-language document classification [27]. They argued that cross-language document classification is a challenge because of cultural discrepancies and translation ambiguities. It is an interesting and similar motivation to our geographic object retrieval example in which the cultural discrepancy may prevent training an accurate model.

In addition to document classification tasks, some studies have focused on domain adaptation for learning to rank tasks. Cai et al. proposed active query selection for ranking adaptation, which seeks instances to be labeled in the target domain for an effective training [7]. Wang et al. tackled the problem of learning to rank, assuming that there is a common latent space in different domains, and mapped instances onto the space preserving the label order [26]. Gao et al. proposed a method based on re-weighting labeled instances for learning to rank in heterogeneous domains [11]. The main idea of their approach is that the label information of source domain instances similar to ones from a target domain is more helpful in predicting the label of target domain instances than dissimilar source domain instances. Although we also focus on an information retrieval task, our problem described in the following section uses pairs of an instance and binary label; thus, we cannot simply adapt their methods to our problem and compare them with our proposed method.

Nakajima and Tanaka proposed a relative query processing method [18]. A relative query consists of a set of instances and an instance in the set. Their relative query processing method represents an instance $x$ in a set $X$ as a vector from the centroid of $X$ to $x$. The differences between domain adaptation for content-based retrieval and the relative query processing method are the following two aspects: (i) a content-based retrieval system returns ranked instances, while the relative query aims to find only a relevant instance, and (ii) domain adaptation for content-based retrieval addresses a problem that the same features may have dif-
different meanings in different domains, while the relative query processing method does not and only addresses the relativity of the feature value. Moreover, we included their method as a baseline in our experiment and showed the difference in effectiveness between our proposed method and the relative query processing method.

We assume that different domains include points that play the same role, like the assumption of common features in most of the previous work. However, we would like to emphasize a fundamental difference: we do not assume that common features in different domains correlate with labels in the same way. The reason for this is that even the same features may indicate different meanings in different domains. For example, sushi restaurants in Beijing and Kyoto that are exactly the same may be relevant in Kyoto, but irrelevant in Beijing for a user who wants a restaurant serving local dishes. Moreover, a test collection we developed for the content-based geographic object retrieval is different from commonly used test collections such as 20 Newsgroups\(^1\) in some aspects. Our test collection includes different types of features: continuous, such as the cost and distance from the nearest station, discrete, such as category information, and high-dimensional, such as the description of a restaurant. The label distribution is usually balanced in the sentiment and category classification tasks addressed in previous work, while the amount of relevant instances to a search intent is relatively small in our test collection and is different in different search intents and domains. The unbalanced label distribution might impair the performance of previously proposed methods, as revealed in our experiment.

3. DOMAIN ADAPTATION FOR CONTENT-BASED RETRIEVAL

In this section, we define the problem of domain adaptation for content-based retrieval and discuss domain difference.

A domain consists of a feature space \(X\) and a marginal distribution \(P(X)\) \([20]\); thus, we represent a domain as \(D = (X, P(X))\). A label space is denoted by \(Y\). Instances \(X \subset X\) are sampled following \(P(X)\), and a label \(y \in Y\) is assigned to each instance \(x \in X\). Label-instance pairs are denoted by \(D = \{(x_1, y_1), (x_2, y_2), \ldots\}\). We set two domains in content-based retrieval: a source domain \(D^{(S)} = (X^{(S)}, P(X^{(S)}))\), from which instances to be selected as a query are sampled, and a target domain \(D^{(T)} = (X^{(T)}, P(X^{(T)}))\), from which instances to be retrieved are sampled. From instances \(X^{(S)}\) sampled from the source domain \(D^{(S)}\), the user can select a subset \(Q \subset X^{(S)}\) that consists of instances relevant to his/her search intent. The example selection can be interpreted as a labeling process on the instances \(X^{(S)}\), and label-instance pairs are denoted by \(D^{(S)} = \{(x^{(S)}_i, +1) \mid x^{(S)} \in Q\} \cup \{(x^{(S)}_i, -1) \mid x^{(S)} \in (X^{(S)} - Q)\}\), where a label space \(Y^{(S)}\) comprises \(-1\) (nonselected examples) and \(+1\) (selected examples). Therefore, a query in content-based retrieval is represented as the set of label-instance pairs \(D^{(S)}\). On another front, instances sampled from the target domain \(D^{(T)}\) are denoted by \(X^{(T)}\), and label-instance pairs are denoted by \(D^{(T)}\), in which a label represents binary relevance or graded relevance for a search intent; thus, a label space \(Y^{(T)}\) is a set of integer values. A content-based retrieval system ranks the instances \(X^{(T)}\) based on a ranking function \(f : X^{(T)} \rightarrow \mathbb{R}\) trained by label-instance pairs \(D^{(S)}\) and target instances \(X^{(T)}\), where a total order \(\prec\) on the instances \(X^{(T)}\) is given by \(x^{(T)}_i \prec x^{(T)}_j \iff f(x^{(T)}_i) \leq f(x^{(T)}_j)\). To evaluate content-based retrieval systems, an arbitrary information retrieval metric is applied to a totally ordered instance set \((X^{(T)}, \prec)\).

For example of content-based geographic object retrieval, a user can select a region (e.g. Beijing) as the source domain \(D^{(S)}\) and another (e.g. Kyoto) as the target domain \(D^{(T)}\). In this case, their feature spaces \(X^{(S)}\) and \(X^{(T)}\) are the same: a set of all possible restaurant instances. A set of restaurant instances in \(D^{(S)}\) is denoted by \(X^{(S)} = \{x_1^{(S)}, x_2^{(S)}, x_3^{(S)}\}\), and we assume that the set is obtained by sampling instances following the marginal distribution \(P(X^{(S)})\). The user can select a subset \(Q = \{x_1^{(S)}, x_2^{(S)}\}\) of \(X^{(S)}\), which can be represented as \(D^{(S)} = \{(x_1^{(S)}, +1), (x_2^{(S)}, +1), (x_3^{(S)}, -1)\}\). A content-based geographic object retrieval system receives \(D^{(S)}\) as a query, and ranks a set of restaurant instances \(X^{(T)} = \{x_1^{(T)}, x_2^{(T)}, x_3^{(T)}\}\) sampled from \(D^{(T)}\).

The most important difference from domain adaptation tasks addressed in previous work is that no label information in the target domain is available during the training process in content-based retrieval. Relevant instances in the target domain depend on the user’s search intents; thus, it is not realistic to prepare labels for all possible search intents. Note that in many cases all the instances from the target domain are not available for training in ad-hoc information retrieval due to the amount of data. Instead of using all the data, one can sample some data from the target domain or re-rank top ranked instances using them as target instances. For content-based geographic object retrieval, it is feasible to use all the target instances because the size of instances in a region the user focuses on is generally small (1,000 at most).

The difference between domains is characterized by the difference in feature spaces and marginal distributions. In content-based geographic object retrieval, for example, features spaces in different domains are the same, i.e. \(X^{(S)} = X^{(T)}\). A restaurant instance is represented as three elements in any domain: the price range, category, and description of a restaurant in natural language. On the other hand, marginal distributions in different domains can be different, i.e. \(P(X^{(S)}) \neq P(X^{(T)})\). For example, there are many expensive restaurants in Tokyo, many Japanese-style restaurants in Kyoto, and many restaurants serving spicy dishes in Beijing. Recent studies proposed methods for quantifying the domain difference by using \(A\)-distance for estimating the learning bound for domain adaptation \([3, 21]\). In our experiment, we demonstrated the domain difference measured by \(A\)-distance \(d_A\) for ensuring that domains we used are different from each other. We used the inequality \(D^{(S)} \neq D^{(T)}\) if and only if the \(A\)-distance \(d_A(D^{(S)}, D^{(T)})\) is greater than a threshold \(\epsilon\).

In summary, domain adaptation for content-based retrieval is defined as follows: given label-instance pairs \(D^{(S)}\) from a source domain \(D^{(S)}\) and instances \(X^{(T)}\) from a target domain \(D^{(T)}\) such that \(D^{(S)} \neq D^{(T)}\), domain adaptation for content-based retrieval is the problem of predicting a ranking function \(f\) that maximizes an information retrieval metric of a totally ordered instance set \((X^{(T)}, \prec)\).

4. DOMAIN ADAPTATION BASED ON RELATIVE AGGREGATION POINTS

We propose a domain adaptation method for content-based retrieval in this section. Our assumption is that different domains include corresponding points that may not be the same in terms of their representation, but play the same role in heterogeneous domains. We first introduce RAPs that we believe to be such points and then propose a method for constructing a new feature representation for each instance based on RAPs. We also introduce several

\(^1\)http://people.csail.mit.edu/jrennie/20Newsgroups/
examples of the RAP for geographic object retrieval at the end of this section. Although this domain adaptation method is a framework and can be applied to any type of instances, we use an example of a restaurant instances for simplicity. A simplified restaurant instance \( x \) has two attributes: the cost \( x_c \) (continuous) and category \( x_c \) (a set of category values such as Japanese-style and sushi), and can be represented as a d-dimensional vector \( x \) if needed.

### 4.1 Feature Representation based on Relative Aggregation Points

A RAP is the expectation of instances in a domain that are classified into a certain class. Let us consider the maximum cost point in a domain \( D = \{ X, P(X) \} \), which is the most expensive restaurant. Given a set of instances \( X \) sampled from the domain \( D \), we can easily identify the most expensive restaurants within \( X \), and represent these restaurants as \( \psi_{\text{MAX}}(X) = \arg\max_{x \in X} x_c \), where \( x_c \) is the cost of \( x \). However, a set of instances \( X \) might not be representatives sampled from the domain \( D \). For example, the set of instances \( X \) may include an extremely expensive out-lier despite the low average cost of the domain. Therefore, we estimate the expectation of the most expensive instances in all possible instance sets sampled from a domain.

Given a domain \( D \) and a subset function \( \psi : 2^X \rightarrow 2^X \) (\( 2^X \) is a collection of subsets of \( X \)), the RAP \( a_\psi \) is defined as follows:

\[
a_\psi = \int_{x \in X} x P(x | \psi) dx,
\]

where \( x \) is the vector representation of an instance \( x \), and \( P(x | \psi) \) is the probability that \( x \) is sampled from a subset \( \psi(X) \) of any instance set \( X \) sampled from the domain \( D \). The probability \( P(x | \psi) \) is given by marginalizing out a set of instances \( X \):

\[
P(x | \psi) = \int_{X \in 2^X} P(x | \psi, X) P(X) dX,
\]

where \( P(x | \psi, X) \) is the probability that \( x \) is sampled from a subset \( \psi(X) \). Intuitively, the subset function \( \psi \) indicates the deterministic membership of an instance, while probability \( P(x | \psi) \) indicates probabilistic membership, which is more stable than the deterministic one.

RAPs are relative because they can vary in different domains for the same subset function \( \psi \). Figure 1 illustrates RAPs as black stars in two domains, where the feature space includes two dimensions: cost and category (e.g. the Japanese-ness of a restaurant). Black star 1 is the minimum cost point, star 2 is the average cost point, and star 3 is the maximum cost point, all of which are different between the two domains. We assume that RAPs in different domains correspond and features of RAPs also correspond. In Figure 1, we regard costs of the minimum cost points as the same, i.e. \( b_1^{(S)} = b_1^{(T)} \), and categories of the maximum cost points in different domains as the same, i.e. \( c_3^{(S)} = c_3^{(T)} \). For the example of Kyoto and Beijing, the maximum cost point in Kyoto and that in Beijing are different: a restaurant serving Kyoto cuisine for 10,000 JPY and one serving Beijing duck for 5,000 JPY. By using the maximum cost point in Kyoto and Beijing, we assume that 10,000 JPY corresponds to 5,000 JPY and Kyoto cuisine corresponds to Beijing duck. Kyoto cuisine in Kyoto is considered the same as Beijing duck in Beijing since those dishes are delicacies in each domain. Based on the above assumption, we can model each instance by the similarity to RAPs, where instances in different domains are similar if their features are similar to ones of RAPs.

We give a concrete feature representation based on RAPs. Let \( H = \{ h_1, h_2, \ldots, h_m \} \) denote a set of similarity functions between two instances, and \( A = \{ a_1, a_2, \ldots, a_n \} \) denote a set of RAPs in a domain \( D \). For example, \( H \) includes the similarity in terms of the cost and that in terms of the category. By using a function \( \phi : X \rightarrow \mathbb{R}^{mn} \), we transform an instance \( x \in X \) into a \( mn \)-dimensional vector \( \phi(x) \). The function \( \phi \) is defined as follows:

\[
\phi(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_n(x))
\]

where \( \phi_i(x) = (h_1(x, a_i), h_2(x, a_i), \ldots, h_m(x, a_i)) \) and each dimension corresponds to the similarity to a RAP measured by a similarity function. Combining the vector representation \( x \) of an instance \( x \) and the feature representation based on RAPs \( \phi(x) \), we obtain an augmented feature representation \( (x, \lambda \phi(x)) \), where \( \lambda \) is a parameter that determines the importance of the feature representation based on RAPs. Note that the vector \( x \) and \( mn \)-dimensional vector \( \phi(x) \) are normalized by their \( L_2 \)-norm in practice.

RAPs are estimated separately in the source and target domains. Letting \( \phi^{(S)} \) denote a function based on RAPs estimated in the source domain and \( \phi^{(T)} \) denote one in the target domain, a source instance \( x^{(S)} \in X^{(S)} \) is converted into \( (x^{(S)}, \lambda \phi^{(S)}(x^{(S)})) \), while \( x^{(T)} \in X^{(T)} \) is converted into \( (x^{(T)}, \lambda \phi^{(T)}(x^{(T)})) \). Thus, our method enables us to take into account ordinary similarity (i.e. the similarity between \( x^{(S)} \) and \( x^{(T)} \)), as well as the similarity based on RAPs (i.e. the similarity between \( \phi^{(S)}(x^{(S)}) \) and \( \phi^{(T)}(x^{(T)}) \)).

The motivating example, relatively inexpensive restaurants serving popular local dishes, is processed by using RAPs in the following way. Relatively inexpensive restaurants should be similar to the minimum cost point in terms of the cost, and restaurants serving local dishes may be similar to restaurants of a category that frequently appears in a domain. It is also possible that expensive or average restaurants serve popular local dishes, and the similarity to such restaurants in terms of the category is helpful to capture what the popular local dishes are. Therefore, we can find relevant restaurants in another heterogeneous domains by calculating the similarity between similarities to those RAPs.

### 4.2 Examples of Relative Aggregation Point

Our proposed method is a framework, and effective RAPs highly depend on the task. We present some examples of the RAPs that are effective for the content-based geographic object retrieval. Note that methods for estimating RAPs from finite samples are not essential for the purpose of this paper; thus, we describe these methods in Appendix A.

**AVG.** The most normal restaurant may be considered equivalent in different domains. The average (AVG) RAP is the average of instances in a domain. The subset function \( \psi_{\text{AVG}} \) is defined as \( \psi_{\text{AVG}}(X) = \{ x | x \in X \} \); thus, the probability \( P(x | \psi_{\text{AVG}}) \) is equal to the marginal distribution \( P(x) \). Therefore, the AVG RAP is defined as follows:

\[
a_{\psi_{\text{AVG}}} = \int_{x \in X} x P(x) dx.
\]
MAX/MIN. The maximum (MAX) (and minimum (MIN)) cost may correspond in different domains as mentioned earlier. The MAX RAP is the expectation of instances that has the maximum cost in a domain. The subset function \( \psi \) is defined as \( \psi \) \( x \in X \) where \( x \) is the cost of an instance \( x \). The probability \( P(x|\psi_{\text{MAX}}) \) is given by \( P(x|\psi_{\text{MAX}}) = \int_{S} P(x|b)P(b|\psi_{\text{MAX}})db \), using the probability \( P(b|\psi_{\text{MAX}}) \) of cost \( b \), which is the maximum cost in any instance set, and the probability of \( x \) conditioned by \( b \). Therefore, the MAX RAP is defined as follows: \( a_{\psi_{\text{MAX}}} = \int_{X} P(x|b)P(b|\psi_{\text{MAX}})dbdx \).

The MIN RAP is obtained in the same way as the MAX RAP, where the subset function \( \psi_{\text{MIN}} \) is defined as \( \psi_{\text{MIN}}(X) = \arg\min_{x \in X} x_b \).

FREQ. Relatively frequent (FREQ) categories may correspond in different domains. For example, Japanese-style in Kyoto and Chinese-style in Beijing represent the same meaning, i.e. a popular category in a domain. The subset function \( \psi_{\text{FREQ}} \) is defined as \( \psi_{\text{FREQ}}(X) = \{x | x \in X \land (\forall c \in x_i) \psi_{\text{FREQ}}(c) > \psi_{\text{FREQ}}(c) \} \), where \( \psi_{\text{FREQ}}(c) \) is the frequency of category value \( c \) in a set of instances \( X \), and \( A \) is a set of instances sampled from the entire domain (or all the domains we have). This subset function \( \psi_{\text{FREQ}} \) returns instances with categories that are more frequent in \( X \) than those in \( A \). A FREQ RAP is obtained as follows in a similar manner to MAX/MIN: \( a_{\psi_{\text{FREQ}}} = \int_{X} \int_{C} P(x|c)P(c|\psi_{\text{FREQ}})dCdx \), where \( C \) is a collection of subsets of category values.

CLS. RAPs can be automatically defined through clustering. The subset function \( \psi_{\text{CLS}} \) is defined as \( \psi_{\text{CLS}}(X) = \{x | x \in X \land x \in S \} \), where a set of instances \( S \) is obtained through clustering applied to instances in both the source and target domains. Since \( P(x|\psi_{\text{CLS}}) \) is obtained from a way of sampling, the CLS RAP is simply defined as follows: \( a_{\psi_{\text{CLS}}} = \int_{S} P(x|S)dx \).

5. TEST COLLECTION

In this section we describe a method for developing our test collection for content-based geographic object retrieval\(^2\), and show the statistics of the test collection including the degree of differences between domains.

5.1 Instances

Our test collection includes 46,945 restaurant instances crawled from a Japanese restaurant Web site GourNavi\(^3\), which is one of the biggest sites that provide restaurant information in Japan. We opted to target at restaurant data for several reasons. Restaurant search is usually conducted with an ambiguous search intent that cannot be easily expressed by using keywords. Moreover, the difference between domains (or regions) is originally derived from the cultural difference and inherits its properties; thus, the difference is generally large and difficult to understand. Therefore, restaurant search is one of the most attractive applications of content-based geographic object retrieval.

Restaurant instances in our test collection include several attributes such as the cost, category, name, genre, and introduction of a restaurant. The name (e.g. Fujiyama), genre (e.g. traditional Japanese sushi bar), and introduction (a detailed description of a restaurant written in natural language, which comprises about 200 words) are text, which were processed by MeCab\(^4\), a Japanese morphological analyzer. Only nouns and adjectives were used to construct a text vector \( x \), whose elements represent the term frequency in those text attributes. Feature selection was carried out excluding terms that occurred less than three times in our test collection. The category attribute is a set of category values, e.g. [bar, sushi, Japanese-style]. We treated the category attribute the same as text attributes (i.e. bag-of-words), and represented a category vector as \( c \).

Although category values are hierarchically structured in the GourNavi Web service (e.g. top categories include Japanese-style and Western-style, and sub categories under the Japanese-style category include sushi and tempura), we did not use the hierarchical structure for simplicity. The text and category attributes were separately processed and were weighted by the tf-idf method. To be more specific, the \( i \)-th element of the text vector \( x \) for an instance \( x \) is \( tf(t_i) log(N/dl(t_i)) \), where \( tf(t_i) \) is the frequency of a term \( t_i \) in the instance \( x \), \( df(t_i) \) is the number of instances that include the term \( t_i \), and \( N \) is the number of instances in our test collection. The category vector \( c \) is weighted in the same manner. To process the cost attribute together with text and category attributes, we discretized the cost and represented it as a probability distribution on discrete values. The cost \( b \) was assumed to follow a normal distribution \( N(z;b,\sigma) \), and was discretized by the probability of each \( z \) in a set of real values \( Z \). The cost \( b \) was represented by a \( |Z| \)-dimensional vector \( x_b \), whose element corresponds to each element \( z \) in \( Z \) and is \( N(z;b,\sigma) \). In our test collection, \( Z \) includes every 500 from \(-5,000 \) to \( 40,000 \), and \( \sigma = 1,000 \). This discretization is not essential but is convenient for computation of the similarity between restaurant instances. Through the above processes, an instance was represented as a \( d \)-dimensional vector \( x = (x_1, x_2, x_3, x_4) \), where \( d = 15,670 \) in our test collection.

5.2 Domains

We chose five major cities (Kyoto, Tokyo, Sapporo, Fukuoka, and Nagoya) in Japan. We selected a region from each city that contain about 500 restaurant instances, and used the five regions as domains in our test collection. Table 1 shows the number of instances, the average cost, and relatively frequent category values in each domain. The average cost in Tokyo is especially high, that in Kyoto follows, and there is no significant difference across other three domains. There are many Japanese-style restaurants in Kyoto, many Western-style ones in Tokyo, and many izakaya ones in Nagoya. The number of seafood restaurants is relatively large in Sapporo and Fukuoka, and that of nabe restaurants is slightly large in Fukuoka. Those differences in the average cost and category indicate the difference between marginal distributions, and may affect the performance of content-based geographic object retrieval.

To measure the difference between the five domains and to ensure that domains in our test collection are significantly different from each other, the \( \mathcal{A} \)-distance \( d_{\mathcal{A}} \) was calculated between all the pairs of domains in our test collection. The \( \mathcal{A} \)-distance was introduced by Ben-David et al. [3], and was defined as the upper bound of the difference between the two marginal distributions \( P(X|S) \)

\(^2\)The dataset is available at http://www.mpkato.net/datasets/.
\(^3\)http://www.gnavi.co.jp/
\(^4\)http://mecab.sourceforge.net/
and $P(X^{(T)})$. To compute the $A$-distance from finite samples of the source and target domains, we approximated the distance by a method introduced by Rai et al. [21].

We calculated the $A$-distance of all the pairs of domains in our test collection, which ranges from 0.537 (Fukuoka-Nagoya) to 0.745 (Kyoto-Tokyo). Rai et al. [21] reported the $A$-distance across eight types of user reviews in the multi-domain sentiment dataset $^3$, and the $A$-distance ranges from 0.0459 (Kitchen-Apparel) to 0.762 (DVD-Book). Comparing the $A$-distance in our test collection and the multi-domain sentiment dataset, we found that our domains are as different from each other as domains used in some previous work [4, 21], even though all the instances are restaurants and all the domains are domestic cities. Therefore, we will use the inequality $D^{(S)} \neq D^{(T)}$ for any pair of the domains hereafter since it was revealed that any pair of the domains is significantly different, i.e. $d_{A}(D^{(S)}, D^{(T)}) > \epsilon$, as mentioned in Section 3.

5.3 Search Intents
To select relevant instances in our test collection, which were used as a query in the source domain and as the ground truth in the target domain, search intents had to be documented for being understood by subjects. To this end, we manually gathered questions from the restaurant category in Yahoo! Chiebukuro $^6$ (Japanese Yahoo! Answers), where the asker asks restaurants that are relevant to his intents. We opted to use Yahoo! Chiebukuro as the resource of search intents because complex and realistic intents are posted there. A hundred of questions were extracted as search intents from Yahoo! Chiebukuro, and were classified into two types in terms of domain-dependency for characterizing each search intent. Domain dependency is the degree of variability of a search intent in different domains, and was subjectively evaluated by two assessors we hired only for this task. We asked them to score the extracted search intent on a scale from one (independent from the domain) to five (dependent on the domain). As their agreement on 100 questions was low: 0.233 in terms of quadratic weighted kappa [24], we separated the labels into domain-dependent (a score is three or more) and domain-independent (a score is less than three), and excluded all the questions on which the two assessors disagreed. As a result, 57 questions remained, of which 20 (35.1%) questions were domain-dependent and 37 (64.9%) questions were domain-independent. We selected ten questions each from the sets of domain-dependent and domain-independent questions, and included them as 20 search intents in our test collection.

An example of domain-dependent search intents is “I’m going to look for business trip with my boss. Do you have any recommendations for dinner?” It would be great if you would tell us restaurants for from 3,000 to 5,000 JPY that provide <v>-specific dishes with Japanese sake. We are 30’s and 40’s. Either meat or fish dish is OK.” On the other hand, an example of domain-independent search intents is “Please show me Western restaurants in <v>. I want a Italian or French restaurant that has a lunch course for 5,000 JPY without wine,” where the variable <v> is replaced with the name of a domain, or a region (e.g. Kyoto).

5.4 Procedure of Relevance Judgment
We recruited 1,000 subjects living in five regions through an online questionnaire service, and asked them to select instances relevant to each search intent in their home area. The five regions are Kyoto city in Kyoto, 23 special wards in Tokyo, Sapporo city in Hokkaido, Fukuoka city in Fukuoka, and Nagoya city in Aichi, which correspond to the five domains, Kyoto, Tokyo, Sapporo, Fukuoka, and Nagoya. We asked the subjects to select relevant instances in a domain corresponding to their home area because it is difficult to judge the relevance without domain knowledge. Note that the recruitment was controlled so that the number of subjects is uniform for each sex, age, and region combination, i.e. 20 subjects were sampled from each (male and female), (20’s, 30’s, 40’s, 50’s, and 60’s), and (Kyoto, Tokyo, Sapporo, Fukuoka, and Nagoya) combination. The online questionnaire service we used is hosted by a domestic company, which delivered a mail to users and asked to visit our Web system. Subjects were rewarded through the company if they finished the whole task we instructed.

The task we asked the subjects to work on consists of two steps. In the first step, the subjects were asked to examine 30 restaurant instances randomly sampled from a domain for subjects’ better understanding of the domain. To make the subjects carefully examine each restaurant instance, we asked subjects to find a couple of restaurants they wanted to visit. In the second step, two search intents were assigned and presented in random order to each subject. The subjects were asked to select as many restaurants relevant to a presented search intent as possible. The subjects were allowed to use a simple search engine, and can search for restaurants by specifying the upper and/or lower limits of the cost, the top category (e.g. Japanese-style, Western-style, and Chinese-style), and sub category (e.g. sake, Italian, and French). The search engine returns a list of restaurant instances that satisfy all the specified conditions, and the search result ranking was randomized to reduce the ranking bias. Relevance judgments for each search intent in each domain were conducted by 20 subjects, where two subjects were selected from each sex and age group.

5.5 Post-Processing of Relevance Judgments
As Alonso and Baeza-Yates indicated [1], it is important to keep the quality of relevance judgments when the assessment is conducted through an online service, or crowd sourcing. We ensured the quality of our test collection by conducting the following two steps. First, subjects who finished a task within five minutes were filtered out, as it was supposed to take more than five minutes in a preliminary system test. Second, we manually assessed all the relevant instances for each search intent in each domain, and removed subjects who made obviously wrong relevance judgments. For example, we removed a subject who selected a restaurant for 1,000 JPY as relevant when the subject was asked to find restaurants for 5,000 JPY. After the removal of low-quality subjects, there remained 579 subjects and 6,082 relevance judgments. On average, 11.6 subjects worked on each search intent in each domain, and made 121 relevance judgments. Finally, we merged duplicate relevance judgments and generate graded relevance by regarding the multiplicity as the grade. The average number of relevant instances per domain and search intent was 42.1, and the standard deviation was 33.3.

Our test collection contains many subjective search intents and needed as many subjects as possible for avoiding the individual bias, whereas it was difficult to ask individual to evaluate all the restaurant instances due to the cost limitation. To gain the coverage of relevance judgments, we assigned 20 subjects to each search intent and randomized the ranking result returned by the simple search system. However, the relevance judgment we conducted is not always complete since it is not guaranteed that all the restaurant instances in a domain were examined by the subjects. As Buckley and Voorhees’s study on retrieval evaluation with incomplete information suggests [6], traditional evaluation metrics such as MAP...
and precision at ten are not robust to substantially incomplete relevance judgments.

To assess the completeness of our relevance judgments, we used a method of estimating the population size by multiple census proposed by Schumacher and Eschmeyer [23]. Their method conducts multiple samplings, which correspond to individual’s relevance judgments in our case, and estimates the population size, which corresponds to the number of relevant instances in our case. Having estimated the number of relevant instances, we can estimate the fraction of judged relevant instances to all the relevant instances. In our test collection, 62.9% of relevant instances were estimated to be actually judged by the subject. According to the Buckley and Voorhees’s work [6], the moderate completeness of our test collection possibly affects the system ranking based on the evaluation metric: the Kendall correlation between the system ranking based on the evaluation metric: the Kendall correlation between the system ranking based on MAP with incomplete relevance judgments and that with complete relevance judgments was 0.9 on average in three test collections from the Text REtrieval Conference (TREC). Although we cannot argue that our test collection is complete and is able to accurately estimate the effectiveness of methods, the effect of the moderate completeness of our test collection is supposed to be small. Moreover, we used an evaluation metric for graded relevance, where relevant instances with a low grade make less impact on the evaluation result than ones with a high grade. As the maximum grade per search intent and region is 4.34 on average in our test collection, relevant instances that were not judged by any subjects are not seriously affect the evaluation metric compared to the binary relevance case.

6. EXPERIMENT

We answer the following two research questions through our experiment: whether domain adaptation for content-based retrieval is necessary, and whether domain adaptation based on RAPs can significantly improve the search performance of content-based retrieval in heterogeneous domains.

6.1 Experimental Settings

We selected a domain as source $D^{(S)}$ and another as target $D^{(T)}$ from our test collection. The input in our problem was a set of pairs of a binary label and instance $D^{(S)}$ from the source domain $D^{(S)}$ as mentioned in Section 3; thus, a label +1 was assigned to relevant (selected) source domain instances, while −1 was assigned to irrelevant (nonselected) source domain instances, where the grade for relevance was ignored since the user does not grade but just selects instances as a query in content-based retrieval. A content-based retrieval method receives $D^{(S)}$ and target instances $X^{(T)}$ and predicts an optimal ranking function $f$ to rank instances $X^{(T)}$ from the target domain $D^{(T)}$. We used both MAP and nDCG [12] in our experiment for evaluating the results in terms of binary relevance and graded relevance. For MAP, relevant instances were defined as ones that were judged as relevant by at least one subject. For nDCG, we set the rank threshold to 10 because it is the common number of search results shown in a search engine result page.

Simple baselines are methods without any knowledge of the domain: nearest neighbor search (NNS), one-class SVM (OSVM), and SVM. NNS ranks target instances based on the similarity to the centroid of selected instances in $D^{(S)}$. OSVM is a method incorporating the one-class SVM [8, 22], which trains a model for binary classification using only selected instances in $D^{(S)}$, and the ranking function $f$ is defined as $f(x) = w^T x - \rho$. Note that $x$ is a vector representation of an instance $x$, and $w$ and $\rho$ are parameters that determine the decision boundary of the one-class SVM. The NNS and OSVM methods do not use nons-elected examples, which are considered typical content-based retrieval methods. SVM is a method based on a linear support vector machine using label-instance pairs $D^{(S)}$, which trains a model for binary classification, and the ranking function $f$ is defined as $f(x) = w^T x + b$, where $w$ and $b$ are parameters that determine the decision boundary of the SVM.

Baseline domain adaptation methods are transductive support vector machine (TSVM), structural correspondence learning (SCL), and relative cluster mapping (RCM), all of which use knowledge of both the source and target domains. TSVM proposed by Joachims [13] has been used as a baseline for domain adaptation tasks [16, 28]. The ranking function of this baseline is predicted in the same way as the SVM method. SCL was also compared with our proposed method, which is a state-of-the-art method for domain adaptation [4, 5]. SCL constructs a new feature space and represents each instance with additional features generated based on pivots (features that behave similarly in the source and target domains). We applied the SVM method to augmented vectors of instances for predicting the ranking function $f$. RCM was selected as a baseline since the problem definition and motivation are similar to ours [18]. RCM represents an instance by the vector from the centroid of a set of instances to a vector representing the instance itself. We then applied the SVM in the same manner as SCL.

Our proposed RAP-based domain adaptation method includes a parameter $\lambda$, and has variants that use different RAPs. We set $\lambda = 1$ (default) and included all the RAPs mentioned in Section 4.2 for the comparison. The best setting for those factors were explored after comparison. A set of similarity functions involves the cosine of the cost $x_0$, that of the category $x_c$, that of the text $x_t$, and that of $x$. The CLS RAPs were obtained by using CLUTO7 with default parameters. The number of clusters was set to 30. The training was conducted using the SVM in the same manner as the baseline domain adaptation methods.

The SVM and TSVM methods were implemented using SVMlight and the OSVM was implemented using LIBSVM9. All parameters were set to default by the software. SCL was implemented according to the paper authored by Blitzer et al. [4, 5]10. For SCL, we selected features that occur more than five times in the source and target domains and showed the highest mutual information with respect to labels in the source domain. The number of dimensions $h$ was set to 50 according to a study using SCL for comparison [19], and the number of pivots was set to 100, which is smaller than the one used in the comparison study, due to the lack of frequently occurring features in our test collection.

6.2 Is Domain Adaptation Necessary for Content-based Retrieval?

We first present an answer to the first research question: is domain adaptation for content-based retrieval necessary? The SVM method, which is a baseline that does not use any domain adaptation technique, was applied to all the search intents. In an in-domain setting ($D^{(S)} = D^{(S)}_k$), we used a method similar to k-fold cross validation, where instances in a domain are split into k bins, of which $k-1$ bins are used as the source and the rest are used as the target. In our study, we set $k$ to 5 and recorded the average

7 http://glaros.dtc.umn.edu/gkhome/views/cluto/
8 http://svmlight.joachims.org/
9 http://www.csie.ntu.edu.tw/~cjlin/libsvm/
10 To test our implementation of SCL, we applied it to a dataset Blitzer et al. developed [4], which is the one we referred to in Section 5.2. The average accuracy was 0.774, which is close to the average accuracy Blitzer et al. reported.
Table 2: The average nDCG@10 over all the search intents in each combination of the source and target domains. The bold font indicates in-domain settings, while the others are out-domain settings.

<table>
<thead>
<tr>
<th>Source</th>
<th>Kyoto</th>
<th>Tokyo</th>
<th>Sapporo</th>
<th>Fukuoka</th>
<th>Nagoya</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tokyo</td>
<td>0.630</td>
<td>0.463</td>
<td>0.520</td>
<td>0.511</td>
<td>0.431</td>
</tr>
<tr>
<td>Sapporo</td>
<td>0.492</td>
<td>0.432</td>
<td>0.638</td>
<td>0.524</td>
<td>0.506</td>
</tr>
<tr>
<td>Fukuoka</td>
<td>0.505</td>
<td>0.472</td>
<td>0.510</td>
<td>0.588</td>
<td>0.477</td>
</tr>
<tr>
<td>Nagoya</td>
<td>0.476</td>
<td>0.466</td>
<td>0.538</td>
<td>0.527</td>
<td>0.599</td>
</tr>
</tbody>
</table>

nDCG@10 over k runs. For an out-domain setting \( (D(S) \neq D(T)) \), we applied a similar method to the in-domain setting for comparison. A domain was used as the source and another was used as the target, where instances in the source and the target domains were split into k bins, of which k - 1 bins in the source domain were used as input and a bin in the target domain was used as instances to be retrieved. The query in our experiment was different for the combination of the source and search intent. The number of unique queries was 100 (5 domains \( \times \) 20 intents). The target domain was selected from the five domains; thus, we conducted 500 runs (5 domains \( \times \) 100 queries).

Table 2 lists the average nDCG@10 over all the search intents in each pair of the source and target domains. It is clear that the average nDCG@10 in out-domain settings \( (D(S) \neq D(T)) \) was lower than that in in-domain settings \( (D(S) = D(T)) \). There was a negative correlation between the average nDCG@10 and the A-distance (Pearson's coefficient \( r = -0.678 \), \( p < 0.05 \)). The negative correlation indicates that the more different domains are, the more difficult it is to achieve high search performance. The average nDCG@10 for the in-domain and out-domain settings are 0.608 and 0.482, respectively. With a Welch's t-test, we found a significant difference between the in-domain and out-domain settings, \( t(179) = 5.89, p < 0.001 \). Moreover, the effect size of the domain setting was 0.588 in terms of Cohen's \( d \), which is considered medium.

The significant difference in search performance between the in-domain and out-domain settings suggests that domain adaptation for content-based retrieval is necessary. This result also implies that relevant data can vary even for the same search intent when the search context is different. The difference in search context in our case consists of the difference between datasets from which the system retrieves data, the difference between search locations, and the difference between user’s home areas. Recently, studies have suggested that relevant data can vary for different users [25] and different search devices [14]. In addition to such hypotheses, our results pose new hypotheses stating that relevant data can vary for different datasets and/or locations where the user searches, which are still open questions since we did not separate the two effects.

6.3 How Effective Is Domain Adaptation based on Relative Aggregation Points?

We compared our RAP-based domain adaptation method with baselines, NNS, OSVM, SVM, TSVM, SCL, and RCM, to demonstrate its effectiveness. We conducted 400 runs (4 out-domains \( \times \) 100 queries) in this experiment. Tables 3 and 4 list the average MAP and nDCG@10 for each search intent in the out-domain setting, respectively. “Transductive” methods use knowledge of the target domain, while “Inductive” methods do not. The bold font indicates the highest score per search intent. One-way ANOVA showed a significant effect of methods on MAP and nDCG, \( F(6, 2793) = 16.1 \), \( p < 0.001 \) and \( F(6, 2793) = 40.0 \), \( p < 0.001 \), respectively. With a paired t-test12, we found significant differences between baselines and the RAP-based domain adaptation method at \( \alpha = 0.05 \), which are indicated by a dagger \( \dagger \) at the “Total” row in Tables 3 and 4.

Our proposed RAP-based domain adaptation method achieved significant improvements over all the baseline methods in terms of MAP, and over NNS, OSVM, and TSVM in terms of nDCG@10. It can be seen that the RAP-based domain adaptation method achieved high MAP and nDCG@10 especially for domain-dependent search intents. SCL achieved comparable nDCG@10 scores but failed to improve the MAP. SCL selected some features as pivots such as “private room” and “seafood,” which are considered common in any domains. The MAP and nDCG@10 of

---

11We also tested the difference between the in-domain and out-domain settings in terms of MAP (0.609 vs. 0.479) and found a significant difference, \( t(157) = 5.35, p < 0.001 \), Cohen's \( d = 0.585 \).

12Holm’s method was used to adjust the significance level for multiple comparisons.
RCM was almost the same as SVM, since the centroid vector of instances had small values for all the dimensions in our case: the value distribution was flattened by taking an average over all the instance vectors. The T SVM was worse than SVM in many of the search intents, probably because the number of relevant instances per search intent was not balanced across domains, the same label distribution in the source and target domains is assumed with T SVM. SVM showed higher MAP and nDCG@10 than methods using only selected examples, i.e. NNS and OSVM. The difference between SVM and those methods is that SVM did use non-selected examples. Therefore, this result suggests that using not only selected examples but also nonselected examples can improve the search performance of content-based retrieval when most of the relevant instances are selected in the source domain.

Our proposed RAP-based domain adaptation method showed considerable improvements of nDCG@10 for search intents 5, 11, and 19. Search intent 5 was “Please tell me a tranquil restaurant appropriate for entertaining foreign guests in <v>.” Our budget is around 8,000 JPY per person,” search intent 11 was “I am looking for Western restaurants in <v> for Christmas dinner for 7,500 JPY per person (but not picky about the cost),” and search intent 19 was “Please show me Japanese restaurants or ones serving meat for celebrating my father’s birthday in <v>.” He cannot eat Italian dishes or drink alcohol. Our maximum budget is 40,000 JPY for three people.” These intents include ambiguous conditions, e.g. a category in search intent 5, and budget in the search intent 11, which can be different in different domains. For example, Japanese restaurants were often judged as relevant in Kyoto for search intent 5, while Western restaurants were usually relevant in Tokyo. As shown in Table 1, Kyoto include many Japanese restaurant and is famous for Japanese-style restaurants because it is one of the oldest cities in Japan. On the other hand, there are many Western-style restaurants in Tokyo because it is a modern international city in Japan. The difference in frequent categories might be lessened by RAPs in our proposed method, which leads to high search performance. Similarly, relevant restaurants were more expensive in Tokyo than those in Kyoto for search intent 11, and the cost difference was probably bridged by RAPs related to the cost.

### 6.4 Parameter Tuning

There are a parameter λ and choices of RAPs in our proposed RAP-based domain adaptation method. We tested several values for λ using all the RAPs mentioned in the experimental setting. The average nDCG@10 over all the search intents are plotted in Figure 2. The setting λ = 0.5 was relatively close to the SVM (no RAP was used), while the setting λ = 2.0 was relatively close to where only features derived from RAPs were used. The best λ was 1.3 in our test collection. These results suggest that both original and domain-dependent features are necessary to better model the similarity between instances in different domains.

We also tested the effectiveness of each RAP. Table 5 lists the relative MAP and nDCG@10 for all combinations of RAPs. The percentage indicates the gain from (None), a method with no RAP, normalized by the maximum gain of MAP or nDCG@10. The FREQ RAP was the least effective in our test collection and rather impaired search performance. In contrast, the CLS RAP was the most effective in our experiment and improved search performance even when only CLS was used. The MAX/MIN and AVG achieved high MAP scores when they were used together. The best performance in terms of both MAP and nDCG@10 was obtained by the combination of CLS, MAX/MIN, and AVG. As future work, we will develop a method of finding effective RAPs for a given task.

### 7. CONCLUSIONS

We introduced the problem of domain adaptation for content-based retrieval, and proposed a domain adaptation method based on RAPs. We developed a test collection for content-based geographic object retrieval, and provided answers to two research questions.

The first research question was whether domain adaptation for content-based retrieval is necessary. Our experimental results showed that evaluation metrics in an out-domain setting (D(S) ≠ D(T)) were significantly lower than those in an in-domain setting (D(S) = D(T)). Therefore, we conclude that the answer to the first research question is yes: search performance without domain adaptation significantly decreases when the source and target domains are heterogeneous, and domain adaptation for content-based retrieval is necessary.

The second research question was whether domain adaptation based on RAPs can bridge the domain difference and significantly improve the search performance of content-based retrieval in heterogeneous domains. Our experimental results answered the second research question by showing significant improvements over baseline methods.

Future work involves applying the RAP-based method to other datasets. In practice, developing a method for finding effective RAPs is the most important part of future work, as it is difficult to find effective RAPs manually. Although we assume that feature spaces in the source and target domains are the same in this paper, it is possible to apply our proposed RAP-based domain adaptation method to heterogeneous feature spaces such as image and music feature spaces. The RAP-based method requires only a within-domain similarity function not a between-domain similarity function for domain adaptation, and enables a user to find popular music by selecting popular pictures. There is also a need to further study the effect of the domain differences, such as the difference between datasets from which the system retrieves data, between search locations, and between user’s profiles, on search tasks.
8. ACKNOWLEDGMENTS

This work was supported in part by the following projects: Grants-in-Aid for Scientific Research (Nos. 24240013, 24680008, and 22-4687) from MEXT of Japan, and a Kyoto University GCOE Program entitled “Informatics Education and Research for Knowledge-Circulating Society.”

9. REFERENCES


APPENDIX

A. ESTIMATION OF RELATIVE AGGREGATION POINTS

We approximately estimated RAPs by using finite samples X from a domain. Expectation $E[x]$, can be approximated as follows:

$$E[x] = \frac{1}{n} \sum_{i=1}^{n} x_i$$

where $|X|$ is the number of instances X. This approximation is directly applied to the equation for the AVG RAP.

Similarly, expectation $E[x|S]$ can be approximated by instances $X_S = X \cap S$ as follows:

$$E[x|S] = \frac{1}{|X_S|} \sum_{x \in X_S} x$$

which is applied to the equation for the CWS RAP.

The equations for the MAX/MIN and FREQ RAPs are the same form and can be written as follows:

$$\max_{x \in X} \frac{1}{|X_S|} \sum_{x \in X_S} x$$

for the MAX/MIN, and

$$\min_{x \in X} \frac{1}{|X_S|} \sum_{x \in X_S} x$$

for the MIN RAP. The function $P(b|\psi_{\text{MIN}})$ of the MIN RAP is defined as $P(b|\psi_{\text{MIN}}) = 1 - (1 - G(b))^l$, where $G(b)$ is the cumulative distribution function of the normal distribution. For the MIN RAP, the probability $P(b|\psi_{\text{MIN}})$ is defined as $P(b|\psi_{\text{MIN}}) = 1 - (1 - G(b))^l$.

For the FREQ RAP, we assume that the frequency of a category value $c$ follows a binomial distribution $B$, and is independent from the frequency of other values. In addition, the number of sampled instances from a domain is assumed to be constant $l$. The probability of the cost $b$ that is the maximum cost in a set of sampled instance is then defined as $P(b|\psi_{\text{MAX}}) = \{G(b)\}^l$, where $G(b)$ is the cumulative distribution function of the normal distribution. For the MIN RAP, the probability $P(b|\psi_{\text{MIN}})$ is defined as $P(b|\psi_{\text{MIN}}) = 1 - (1 - G(b))^l$.