How to Find a Place Suitable for "Guitar Practice": Purpose-oriented Geographic Entity Retrieval by Using Online Review Graph Analysis

Yui Maekawa*
Tokyo Institute of Technology
Tokyo, Japan
maekawa@sw.it.aoyama.ac.jp

Yoshiyuki Shoji
Aoyama Gakuin University
Kanagawa, Japan
shoji@it.aoyama.ac.jp

Martin J. Dürst
Aoyama Gakuin University
Kanagawa, Japan
duerst@it.aoyama.ac.jp

ABSTRACT

This paper proposes a method of ranking geographic entities (places) where the purpose given as a query can be achieved. Most existing map search engines accept only the name of a place or the type of a place. Thus when searchers want to find a suitable place for "guitar practice", they have to input a place type such as "music studio". To create such a query, prior knowledge (*i.e.*, that a music studio is suitable for playing guitar) is required. Our proposed method uses online review information on places to enable direct place retrieval from a given purpose query. Our method creates a bipartite graph consisting of places and the words that appear in the reviews of these places. The relevance between the given keyword query and a place is calculated by using the Random Walk with Restart algorithm. Additionally, we expand the graph with three hypotheses; 1) places that are suitable for the same purpose are similar to each other, and purposes that can be achieved in the same place are similar to each other, 2) the same purpose can be achieved in places with similar metadata, and 3) purposes which have semantically similar meaning can be achieved in the same places. Through an experiment using real review data taken from Google Maps, the usefulness of the proposed method was demonstrated. In particular, it was found that the expansion by places' metadata is effective for finding more relevant places.

KEYWORDS

Place Search, Random Walk with Restart, Online Review

1 INTRODUCTION

In recent years, geographic information retrieval is becoming more and more popular. A wide range of people uses place search services (*e.g.*, Google Maps) to find stores, facilities, and other places. The users of place search include children who do not have sufficient prior knowledge of places and elderly people who are not good at searching. Nowadays, with such a wide range of people using geographic information retrieval services, there is a growing demand for geographic information retrieval algorithms that allow users with little prior knowledge of geographic information to search successfully.

Conventional geographic search systems only accept the name of a place or the type of a place as a query. Therefore, in order to find a place where a specific purpose can be achieved, the user needs to enter the type of the venue, or characteristics of the place he or she wants to find. For example, if you want to buy a book, you must search for "bookstore"; if you want to use a delivery service, you must enter the query "post office". Let us imagine the case that a user is looking for a place where he can achieve "guitar practice". In this case, normally, it is necessary to search with a place type query, such as "music studio". However, in order to create this query, users need prior knowledge, such as "we can practice guitar in a music studio". Therefore, it is impossible to input any facility where they can practice guitar without that prior knowledge. This problem can be solved if we can search places by purpose, using "guitar practice" as a query.

In addition, even if a searcher has prior knowledge that guitar can be practiced in a music studio, the query "music studio" may not find a large number of places where the searcher can practice guitar. A music studio is not the only place where a guitar can be played. There are many places where this is possible: parks, karaoke rooms, riversides, and so on. It is not reasonable to list all these places in the search query.

In this research, we propose a new search algorithm that ranks places by the possibility that they can achieve the purpose indicated by the query. For example, if the user enters "guitar practice", the system will rank specific places such as "Studio FOO Tokyo branch", "BAR Karaoke Tokyo branch", or "Tokyo central park". The aim of our search algorithm is to allow the user to input a purpose, so that a wide range of users can search places more easily, regardless of prior knowledge.

The reason for the effectiveness of such a search model is that the search difficulty is asymmetric. It is easy to determine if a place makes it possible to achieve a purpose by accessing the official Website or by calling the place. However, it is difficult to make a list of candidates. If the places with a high likelihood of achieving the purpose can be ranked, users can find a suitable place in very few steps.

In this research, we focused on online reviews about places to realize our search algorithm. Some geographic information services, such as Google Maps, allow users to post reviews of a certain place. Such reviews include many actual and feasible actions taken by users at the place.

Although these online reviews taken from geographic information sites are an important information resource, they are not sufficient to implement the proposed search algorithm directly. One of the reasons is the limited comprehensiveness of the reviews. The review information does not always describe all the actions that can be performed at a place. For instance, not all places where you can practice guitar have a review that says "I practiced my guitar

*Yui Maekawa contributed to this research while at Aoyama Gakuin University until March 2020

here". Traditional information retrieval methods based on simple string matching can therefore not take advantage of the reviews.

Therefore, we propose a graph-based algorithm that links given purpose queries and places, by setting up the following three hypotheses:

H1 Mutual Recursive Deduction:
Places that are suitable for the same purpose are similar to each other, and purposes that can be achieved in the same place are similar to each other.

H2 Expansion by Place Type:
The same purpose can be achieved in places with similar metadata. For instance, if you were able to play guitar in a certain Karaoke room, there is a high probability that you can play guitar in another Karaoke room.

H3 Expansion by Word Semantics:
Purposes which have semantically similar meaning can be achieved in the same places. For instance, if a certain park gets a review saying "This place is suitable for playing ukulele", this park should also be suitable for playing guitar.

The proposed method performs Random Walk with Restart (RWR) link analysis on a bipartite graph. This graph is composed of places and the words that appear in reviews for these places. In order to clarify the effectiveness of our method, an experiment using real data was conducted. For the experiment, we implemented an actual place search system that uses review data obtained from Google Maps. In this system, when a searcher inputs a purpose as a query, they can obtain the ranking of places suitable for achieving that purpose. The accuracy of the method was checked by performing actual searches with pre-prepared queries and manually labeling the results.

The structure of this paper is as follows. In Section 2, we discuss existing research related to our method. Section 3 describes the details of our search algorithm. In Section 4, the proposed method is evaluated through an experiment. Section 5 discusses the results of the experiment, and Section 6 presents conclusions and future work.

2 RELATED WORK

This research is part of the research on purpose-oriented search algorithms. We adopt a graph approach, extend places by metadata, and extend purposes with synonyms. Therefore, this research is closely related to the existing research of geographic information retrieval, expansion of purpose, and locality recommendation.

2.1 Geographic Information Retrieval

Major geographic information search systems typically accept place names, place types, and addresses as queries for finding places. Therefore, a lot of research has been done on a search to enable more flexible input, such as extending the query.

Pat *et al.* [7] developed a geographic information retrieval system that collects location information (geotagged posts) from social networking sites such as Twitter and Instagram, and represents the results in terms of territory. By focusing on geotagged posts on social networking sites, they attempted to make the normally static geographic information database dynamic.

Shoji *et al.* [9] also proposed a method using geotagged tweets for finding places. Their method, named "location2vec", is based on a word2vec-like algorithm, and it can find similar places by comparing tweets around different places. However, since many users post their tweets with automatic geotagging by the SNS system, posts about a place made after moving somewhere else have the wrong geotag. Therefore, the accuracy of the information in geotagged posts is questionable. This research is similar to the present studies because they also focus on social data for geographic information retrieval. However, we chose review information for a place instead of SNS posts, because compared with SNS posts, there is a much higher likelihood that they contain information related to the place.

Bauer *et al.* [2] analyzed offline purchasing needs and proposed a search method for physical brick-and-mortar stores where actual purchases can be made, while online mail-order sales are now common. This is accomplished by querying the keywords representing the object to be purchased and vectorizing the locations, respectively, and ranking them by cosine similarity. This research is similar to our research in that the search targets are actual objects. However, our research does not use a simple similarity calculation in a vector space model, but a link analysis on a two-part graph. The difference is that we aimed to widen the range of input data in the search. As a result, we can find not only places where people can buy something from a retailer, but also other places.

Kato *et al.* [5] expanded the input of place search to allow examples as queries. In their method, searchers can input a certain place, and the system finds similar places. It can help find places by purpose, but the searchers need to know an example place that is suitable for their purpose.

2.2 Expansion of Purpose

In this research, the goal is to improve the recall of search results by extending viable objectives at the same place by inference. In other words, it is possible to search for local products and other stores in the same chain that does not include query words in their reviews.

As an example of extending purposes, Pothirattanachaikul *et al.* [8] proposed a method for extracting alternatives that can achieve the same objectives from community Question Answering (cQA) sites. For example, "taking sleeping pills" and "drinking warm milk" are alternative behaviors that can achieve the same goal of "falling asleep easily". Their research uses a bipartite graph consisting of the question and answer information extracted from the cQA site. By analyzing this graph, they were able to find alternative behaviors by ranking similarity levels.

The expansion of purpose is also a big problem in research on cQA. Jiwoon *et al.* [3] proposed a method of finding questions with a similar purpose in a cQA site. It can help people who have a purpose but do not want to ask a question on a cQA site. This method focused on how to calculate the similarity of questions. The ideas used are related to ours, such as that places suitable for the same purpose are similar to each other, and purposes that can be achieved in the same place are similar to each other. Wang *et al.* [12] also tackle this problem. They used a natural language processing-based approach that uses syntactic trees. Our method has to consider both purpose similarity and place similarity.

2.3 Locality Recommendation

This research aims to find a place where the user's objectives can be achieved. For the same purpose, there are studies that extract the characteristics of places and solve the problem by recommendations and other approaches.

Kurashima *et al.* [6] proposed a method for extracting features of a place by extracting information from a blog and visualizing the experience of a place on a map by topic modeling. This research uses a more exhaustive but less descriptive review to estimate what can be done at a location in order to discover geographical objects from a query.

As another research on recommending places, Wang *et al.* [11] extended the Bookmark-coloring algorithm to represent information about past behavior on social media sites, location information, relationships between users, and user similarity as a graph. By using the similarity between users, they can recommend the next place the user is likely to visit with higher accuracy than conventional recommendations.

Many studies have been conducted to estimate the nature of a place from information gathered from social media and CGM (Consumer Generated Media) sites. Among them, many studies use Location-Based Social Networking (LBSN) such as Twitter [10]. The most typical example is real-world event detection or travel assistance. For instance, Dong et al. citedong2015multiscale proposed a method of finding events by using Flikr photos. As a task to estimate the nature of a place, Zhang *et al.* [13] integrate social media information to estimate the atmosphere and usage of a street.

POI discovery is another important element in the geographic recommendation. The discovery of spots that attract people's attention from social media is close to the discovery of places that are suitable for achieving objectives in this research. Some research uses social media information and detects POIs and their usage or category [4]. Some studies have used review information as well as this study[1].

3 METHOD PROPOSED

This section describes a new algorithm: a method that ranks places suitable for a purpose directly given as a query. In order to realize such a retrieval model, we extract places and the actions which were taken at the place from reviews of these places. Not all actions that can be taken at a place are described in reviews of this place. Therefore, to search for places that do not have a direct purpose in their reviews or that are not reviewed, the method has to deduce and extend objectives of what can be done in that place.

As the expansions of purpose, we adopt the following three hypotheses into a graph-based algorithm:

H1 Mutual Recursive Deduction,
H2 Expansion by Place Type, and
H3 Expansion by Word Semantics.

The first hypothesis is a substantial part of our algorithm. The places where people can achieve the same purpose are similar to each other. For instance, a park and a river beach are similar places, because you can do the same things (*e.g.*, playing a musical instrument, jogging, playing catch) in both of them. In addition, the purposes that can be achieved in the same place are similar to each other. For instance, eating hot-dog and drinking beer are similar purposes,

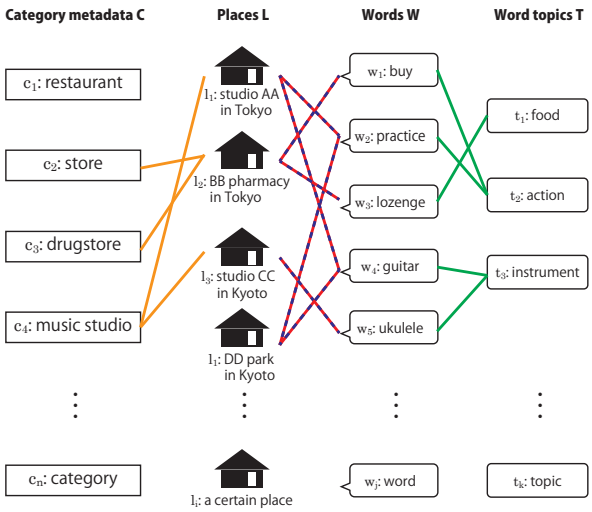

Figure 1: A graph representation of the whole place-review dataset

because both of them can be achieved in the same places (*e.g.*, diners, beer halls, baseball stadiums). To reflect this hypothesis, the method creates a bipartite graph consisting of places and purposes. Thus, a reciprocal recurrence calculation is performed by link analysis. The second hypothesis stands on the idea that the same purpose can be achieved in places that have a similar type. For instance, if you were able to buy a burrito at a certain Starbucks, you would be able to buy it at another branch of Starbucks. In addition, you might be able to buy burritos in other coffee shops. To integrate this hypothesis, our method modifies the bipartite graph by adding links between places and places. The last hypothesis means that purposes which have semantically similar meaning can be achieved in the same places. For instance, if it is possible to buy toilet paper at a certain store, it will be possible to buy tissue paper at the same store, because Our method integrates these hypotheses by adding virtual links between purposes.

3.1 Creating the Bipartite Graph for Mutual Recursion

Our method uses the review information about places as the data source that reflects purposes that can be achieved at each place. First, our method makes a bipartite graph that consists of words and places to express the first hypothesis. The words that appear in reviews of the same place are likely to be similar to each other, and places with reviews containing the same words are similar to each other. The graph contains two types of nodes: all the places in the dataset, and all the words in all the reviews for these places.

A schematic diagram of the entire dataset is shown in Figure 1. The review data is represented as the relationship between a place l_i and a word w_j that appear in the review for that place. Furthermore, there exists a relationship between a place and the metadata about that place, and a relationship between a word and its topics.

First, we create a weighted directed bipartite graph, focusing on the relationship between a place and the words in the review about

it. As a pre-processing step, each review sentence was divided into words. For cleansing the review data written in natural language, word selection by word-class was performed. Only verbs, nouns, and adjectives were treated as nodes. Each word was lemmatized, all verbs were straightened to the standard form, and all word changes (*i.e.*, plurals) were removed. Cleansing by frequency was also done. Words that appeared too frequently or very rarely were removed. Finally, places and words were linked by edges if the word appeared in the review for the place. The bipartite graph created in this phase is shown as a subgraph in the middle of Figure 1, with red and blue lines as edges.

Second, we create the adjacency matrix $\boldsymbol{M}$ from the created graph. Figure 2 shows a schematic diagram of the final shape of the adjacency matrix $\boldsymbol{M}$, where L is the set of all the place nodes in the graph and W is the set of all the word nodes in the graph. The matrix $\boldsymbol{M}$ is a square matrix of dimension $(|L| + |W|)$, where $|L|$ and $|W|$ denote the number of elements in the sets.

Here, the value of each element m_{ij} of the matrix $\boldsymbol{M}$ is defined as below. Figure 2 shows the overall structure of matrix $\boldsymbol{M}$. Let $N_w(l_i)$ be the subset of W connected to l_i, and $N_l(w_i)$ be the subset of L connected to w_i. In Figure 2, the links from places to words are located in the lower left blue part (*i.e.*, $i > |L|$ and $j \leq |L|$). m_{ij} is set to 1 if w_i is an element of $N_w(l_j)$, and is 0 otherwise. Similarly, the upper right red part of the matrix in Figure 2 represents the links from words to places. m_{ij} is set to 1 if l_i is an element of $N_l(w_j)$, and 0 otherwise. That is, in the lower left and upper right part of the matrix in Figure 2, m_{ij} will be 1 if the i-th node and the j-th node are connected by an edge, and 0 otherwise. The review information was rearranged into four relationships, which can be represented as a single adjacency matrix.

Finally, we normalized the weight of the edges that connect words to places. As the number of edges increases in the unweighted state, the value increases cyclically in dense parts in the graph. Therefore, we divide the weight of an edge by the number of outgoing edges of the source node.

3.2 Calculating Place Similarity for Place Type Expansion

Next, in order to adopt Hypothesis 2 (Expansion by Place Type), we added information about the relationships between places to the graph. We hypothesize that the same purpose can be achieved at similar places, for instance, "Starbucks in Tokyo" and "Starbucks in Kyoto", which are separate branches of an affiliated store. By considering the similarity between places, it becomes possible to find places that are not directly reviewed. Therefore, we extend the graph to take into account the similarity between places by comparing their metadata.

In most online map applications, such as Google Maps, there exists metadata for each place. A typical kind of metadata is the category information of a place, such as "restaurant" or "hospital". In this research, we used such categorical information about places as a feature of places. We calculated the degree of association between places by using metadata that indicates the relationship between them, and added the similarity into the graph. There are various methods for calculating the degree of association between objects.

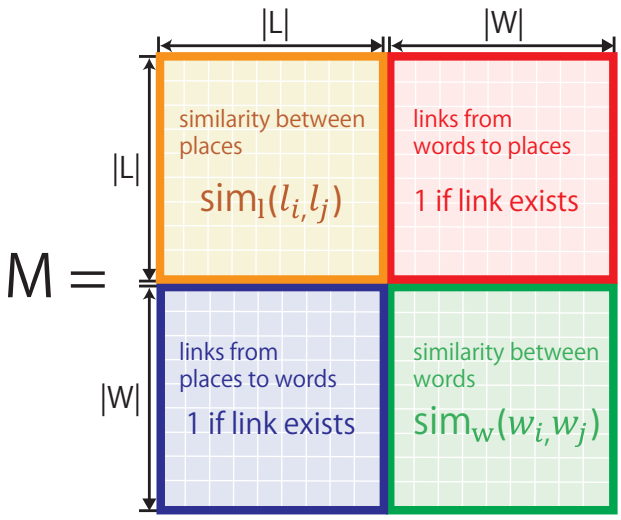

Figure 2: An overview of the expanded adjacency matrix M, which represents the relationships between places (L) and words (W).

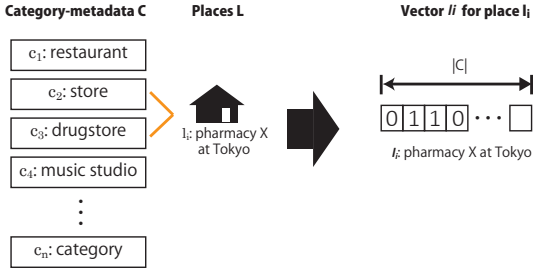

Figure 3: Vector representation of a place by using category metadata

In this research, we adopt the cosine similarity of their category, as the most straightforward approach.

The metadata for a place can be considered a Boolean value vector. This allows us to compute the similarity between places as a distance in a vector space. The vector $\mathbf{l_i}$ of the place l_i is a $|C|$-dimensional vector where the set of all metadata is defined as C. Each element is set to 1 for the j-th element of the vector if there is a link to the metadata $c_j \in C$, or to 0 otherwise (see Figure 3).

The similarity $\mathrm{sim}_l(l_j, l_i)$ between the places l_i and l_j is defined as

$$\mathrm{sim}_l(l_i, l_j) = \frac{\boldsymbol{l_i} \cdot \boldsymbol{l_j}}{|\boldsymbol{l_i}||\boldsymbol{l_j}|}, \tag{1}$$

which is based on cosine similarity.

The calculation cost is a big problem for the actual link analysis calculation. In most cases, the number of category tags that are linked to a place is as few as 1 to 5, and the number of category tags is less than 100. Most places have a few tags, and some of the tags are used too frequently. We eliminated frequent tags that have

no explanatory ability. In our implementation, we set a threshold and cut off some links.

Thus, we used metadata consistency to extend the graph by attaching virtual edges between places. In the upper left part (orange part) of the matrix of Figure 2, m_{ij} is set to 1 if the metadata of places l_i and l_j are highly similar; otherwise it is set to 0.

3.3 Calculating Word Similarity for Purpose Expansion

Next, we extend the graph by focusing on Hypothesis 3 (Expansion by Word Semantics). The degree of association between words is calculated, and added to the graph. For example, guitar and ukulele are lexically close in their meaning. Therefore, we can extend the result so that where you can achieve "guitar practice", you can achieve "ukulele practice". Thus, we added a virtual link between them. This expansion aims to allow reviews that do not contain their purpose directly to be reflected in the rankings of places. Here we extend the graphs to take into account the similarity between words.

The computation of semantic similarity between words is a general problem, and it can be solved by vectorization with methods such as LDA, LSI, or Word2Vec. Our method utilizes a similarity calculation using Word2Vec. A Wikipedia corpus was used for learning the Word2Vec model, because encyclopedic sites are suitable resources to calculate lexical similarity.

The word similarity $\mathrm{sim_w}(w_i, w_j)$ (where w_i is the i-th word) can be used to weigh the links between words in the graph. The distributed representation of a word w_i is defined as follows:

$$\boldsymbol{w_i} = \mathrm{w2v}(w_i). \tag{2}$$

By using this vector, the similarity between the words w_i and w_j can be defined as

$$\mathrm{sim_w}(w_i, w_j) = \frac{\boldsymbol{w_i} \cdot \boldsymbol{w_j}}{|\boldsymbol{w_i}||\boldsymbol{w_j}|}, \tag{3}$$

which is the cosine similarity between the two vectors.

The similarity $\mathrm{sim_w}(w_i, w_j)$ takes a value between 0 and 1. As with the case of similarity between places, we treat this value with a threshold to reduce the computational cost. Finally, we used $\mathrm{sim_w}(w_i, w_j)$ as a Boolean value for calculation. The right bottom part of Figure 2 represents $\mathrm{sim_w}(w_i, w_j)$ for each word in the dataset.

3.4 Ranking Places by Random Walk with Restart

So far, creating the matrix $\boldsymbol{M}$ that represents the expanded graph shown in Figure 4 has been accomplished; it contains all the necessary relationships between places and words, relationships among places, and relationships among words. By processing this matrix, it is possible to compute the relevance of the nodes in the graph. The relevance between a word node and a place reflects how the words in the query are related to the place. In other words, it can rank the places that can achieve the purpose. We adopted Random walk with Restart (RWR) as the algorithm for calculating the degree of association between nodes in our graph. First, in order to perform relevance calculations with RWR, we transformed the graph matrix $\boldsymbol{M}$ into a transition probability matrix. The transformation to the

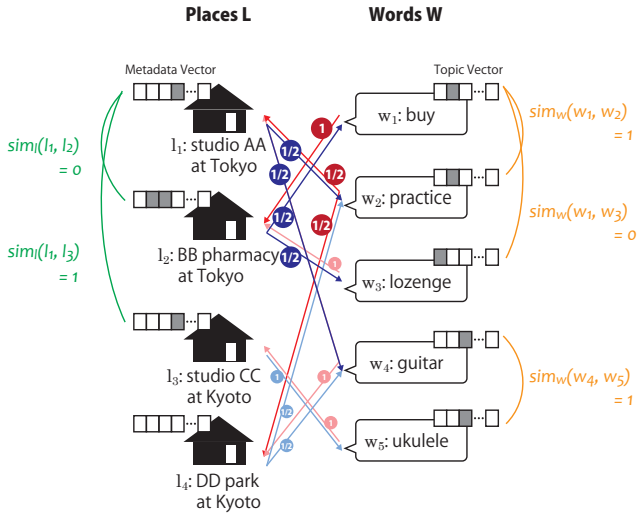

Figure 4: Place-word graph expanded with word semantic similarity and place metadata similarity

transition probability matrix was done by normalizing the matrix by columns, that is by dividing each entry by the sum of the weights of the exit edges. Therefore, we need to consider $\boldsymbol{M}$ as a directed graph; the link from a place to a word and the reverse link has different weights.

Note that you can change the weights for each hypothesis here. For example, if you want to increase only the similarity score between places, you can apply a weight only to the elements in the upper left part of Figure 2 before this transformation.

The formulation for the actual calculation is as below. Let L be the set of all geographic nodes in the graph and W be the set of all word nodes in the graph. $|L|$ and $|W|$ represent the number of elements in each set. $N_w(l_i)$ is the subset of W connected to the edges exiting l_i, and $N_l(w_i)$ is the subset of L connected to the links exiting w_i. The function $\mathrm{sim_l}(l_i, l_j)$ means the similarity between the i-th place and the j-th place, and the function $\mathrm{sim_w}(w_i, w_j)$ means the similarity between the i-th word and the j-th word. The matrix which represents the graph structure $\boldsymbol{M}$ is defined as

$$\boldsymbol{m_{ij}} = \begin{cases} (\text{if } i > |L|) \begin{cases} (\text{if } j > |L|) & : \beta \mathrm{sim_w}(w_i, w_j) \\ (\text{if } j \le |L|) & \begin{cases} (\text{if } w_i \in \mathrm{N_w}(l_j)) & : \frac{1}{|\mathrm{N_w}(l_j)|} \\ (\text{otherwise}) & : 0 \end{cases} \end{cases} \\ (\text{if } i \le |L|) \begin{cases} (\text{if } j > |L|) & \begin{cases} (\text{if } l_i \in \mathrm{N_l}(w_j)) & : \frac{1}{|\mathrm{N_l}(w_j)|} \\ (\text{otherwise}) & : 0 \end{cases} \\ (\text{if } j \le |L|) & : \alpha \mathrm{sim_l}(l_i, l_j) \end{cases} \end{cases} \tag{4}$$

where α and β are weights for each hypothesis (α for **H2**, β for **H3**), both of them taking values from 0 to 1, and $\alpha + \beta \le 1$. The transition probability matrix $\boldsymbol{M'}$ which is $\boldsymbol{M}$ normalized by its rows is defined by

$$\boldsymbol{m'_{ij}} = \frac{\boldsymbol{m_{ij}}}{\sum_{k=1}^{|L|+|W|} \boldsymbol{m_{kj}}}, \tag{5}$$

where m'_{ij} is an element of $\boldsymbol{M'}$.

RWR is an algorithm to compute the degree of association between nodes by performing a random walk on the graph and randomly jumping to the initial node with a fixed probability at each step. Normally, to represent the jumping probability for the initial node q, a one-hot vector $\boldsymbol{q}$ with the q-th element being 1 and the other elements being 0 is used. The nodes of the words that appear in the given query can be used as the initial nodes.

However, in this research, we have to consider the case where the query consists of multiple words, such as "guitar practice". If the given query consists of two or more words, a random jump to all the words in the query will give high relevance to place nodes that are not related to the query. This is because the words in the query are not independent. For example, the query "practice guitar" can be split into two-word nodes, "practice" and "guitar". If these two words are independently used as start nodes, the search results will be a mixture of places associated with "guitar" and places associated with "practice". The result will be similar to the result of an OR search on a traditional search engine. A place node that is highly associated with "practice" may not be a suitable place for "guitar practice". It might be suitable for other kinds of "practice", such as "baseball practice" or "painting practice". Likewise, not all "guitar" related places are suitable for "guitar practice"; some of them may be good places to fix a guitar, or to buy a new guitar.

The solution to this problem (*i.e.*, realizing AND search) is to set the initial nodes to place nodes instead of word nodes. We set the initial nodes to only the place nodes where all the words in the query appear together in a single review. If there is more than one corresponding place, we randomly jump to all these place nodes with equal probability. This enables the algorithm to increase the number of search results for long queries without a loss of accuracy.

The set of initial nodes is represented as a vector of $\boldsymbol{r}$ of $|L|+|W|$ dimensions. Each dimension $\boldsymbol{r_i}$ is 1 in case the i-th node meets the condition, and 0 otherwise. To convert $\boldsymbol{r_i}$ to a probability vector, it is normalized.

The RWR score for each node is calculated by the power method, repeating the equation below:

$$\boldsymbol{p} = (1-c)\boldsymbol{M'}\boldsymbol{p} + c\boldsymbol{r}. \tag{6}$$

As the initial value of $\boldsymbol{p}$, we used $\boldsymbol{r}$. Repeating is continued until $\boldsymbol{p}$ converges. After the convergence, the values of each element $\boldsymbol{p_u}$ in the final $\boldsymbol{p}$ can be used as relevance of the u-th node for the given purpose query. The search result ranking is obtained by sorting all places $l_i \in L$ by $\boldsymbol{p_i}$ in descending order.

4 EXPERIMENT

We evaluated the method's usefulness in an experiment using real data collected from Google Maps. The search results of five methods for nine pre-prepared purpose queries were manually evaluated. An evaluator manually evaluated each of the top-ranked places.

The number of evaluators was one, because it is objectively possible to determine whether an action is feasible in a given place. When the evaluator was unsure about the decision for a place, they accessed the official Website of that place, or called and inquired if people were able to achieve their purposes there.

4.1 Dataset

For the experiment, we used the review data of places and place metadata collected with the Places API of Google Maps. First, we used the Places API of Google Maps to collect review information about places and their correlations. Google Maps puts a quantitative limit on the data that can be collected in a certain period of time. Therefore, we limited the search to about 80km^2 in a densely populated area of Tokyo, Japan, mainly in the Shinjuku, Shibuya, and Chiyoda wards, and we collected all the places (*i.e.*, geographic entities like shops, facilities, and so on) contained in this area.

The list of places in an area and the reviews for them had to be collected via different APIs. The Google Find Place API limits the collectible number of places to only the top 32 results within the specified area. Therefore, we recursively called the API by dividing bigger ranges into four quadrants when the number of included objects reached the upper limit. Finally, by reducing the area to 25m square, 261,492 places were obtained. The reviews for these objects were collected using the Place Details API. Due to API limitations, only the top five reviews for each site were obtained. This resulted in 85,942 places with at least one review with text.

4.2 Implementation

The reviews of 85,942 places in Google Maps were divided into words by using the Japanese morphological analyzer MeCab (Since words in a sentence in Japanese are not separated by spaces). We used the dictionary called mecab-ipadic-NEologd, which includes neologisms frequently used in social media services. The words used in our experiment were limited to verbs, nouns, and adjectives, and the verbs were unified to the standard form. Word cleansing was done by word frequency: rarely used words and words that appeared too often were removed. We removed words that appeared in less than 50 of the 85,942 reviews and words that appeared in more than 40 percent of the reviews. In the end, 9,816 words were considered as nodes in the graph.

Next, we pre-calculated the degree of similarity between places. In order to calculate the similarity between places, we used category tags. Each place in Google Maps has a maximum of five category tags. We used 97 categories assigned to the collected places, excluding categories that occur frequently (*i.e.*, establishment and point_of_interest) for generating a vector consisting of Boolean values. By using this vector, we were able to compute the cosine similarity in the vector space. In this experiment, due to the computational complexity, we used only places with three or more categories of similarity and whose vectors are exactly the same as each other.

The similarity of the words was calculated in advance. In the proposed method, the words in the graph are connected to each other by virtual edges to account for semantically similar objectives. We computed the similarity between all combinations of words for 9,816 word nodes. As a data source for learning the word2vec model, we used Wikipedia data. As an implementation of Word2Vec, gensim, Python's topic analysis library, was used. In order to keep the matrix sparse to reduce computational effort, only combinations with similarity greater than or equal to 0.5 were adopted, and other combinations were treated as having zero similarity.

Finally, we computed the actual Random Walk with Restart and the fit between the query and the ground objects. To speed up the computation of a square matrix of 95,758 dimensions consisting of objects and words, the Python library SciPy was used.

4.3 Comparative Methods

To analyze the effectiveness of the three hypotheses, we prepared the following five methods:

- **All** (H1, H2, H3) is the method proposed that considers all hypotheses, *i.e.,* ($\alpha = 0.1$, $\beta = 0.1$),
- **Place Only** (H1, H2) is a variant method which only considers place type similarity, *i.e.,* ($\alpha = 0.1$, $\beta = 0$),
- **Word Only** (H1, H3) is another variant method which only considers semantic similarity of words, *i.e.,* ($\alpha = 0$, $\beta = 0.1$),
- **No Expansion** (H1 only) is a plain method which does not consider similarity of places and words, *i.e.,* ($\alpha = 0$, $\beta = 0$), and,
- **Baseline** is a traditional search algorithm that only finds places which have reviews directly containing all query words.

A set of places for evaluation were created for pre-prepared queries by these five methods. The top 20 rankings obtained from each method were evaluated. The search results were sorted randomly.

4.4 Answer Labeling

Nine queries were prepared (see Table 1). For these queries, the search result rankings were obtained for the five methods above. The places ranked in the top 20 of these search results were manually labeled with binary values: 1 if it was possible to achieve the purpose there, 0 otherwise. Since the search result of the baseline method is not a ranking, 20 randomly selected places in its result were evaluated.

Labeling was performed by a single evaluator, because it is objectively possible to determine whether or not the purpose is achievable at a given place. If in doubt about whether a purpose was achievable, the evaluator was allowed to check the websites or make a phone call to the place.

Note that this research does not consider the time of day or season (*i.e.*, methods ignore the timestamps of reviews). For this reason, places whose purpose is achievable during a certain time of the year (*e.g.*, a swimming pool that is open only in summer) were labeled as correct. Similarly, places where it was possible in the past to achieve the purpose (*e.g.*, places that changed their business, or closed) were also labeled as correct.

4.5 Result

We describe the method-by-method and query-by-query precision and ranking evaluations, and the actual output. Table 1 shows the p@k (precision at k) and nDCG (normalized Discounted Cumulative Gain) obtained by the nine queries used in the experiment. (However, nDCG cannot be computed for the **Baseline** because it is a Boolean search, not a ranking.)

As the overall result, all proposed methods achieved higher precision than **Baseline**. For the average results of all queries, **Place Only** obtained the highest score.

The highest precision of the **All** method was achieved when the queries were "enjoy afternoon tea" and "buy pizza". For these queries, **All** greatly outperformed precision and nDCG of **Baseline** and **No Expansion**. When the query was "buy computer", all methods obtained low precision. However, even for such a difficult search task, **All** and **Place Only** performed better than **Baseline**.

5 DISCUSSION

This section discusses the nature of each method, and the usefulness of the search results.

To discuss the nature of the proposed methods based on the experimental results, a comparison of the advantages of each method is needed. Across the board, **Place Only** was the most effective for both precision and nDCG. Method **All**, with all expansions added, showed higher precision than the baseline. When focusing on nDCG, every expansion was more effective than **No Expansion**.

We discuss the quality of the obtained results. The proposed method was able to find many places that were not found in the baseline. Many of the places found were judged as suitable for the purpose. The actual search results included different places depending on the expansions used. This suggests that each of the expansions affected finding more relevant places.

We focus on the cases in which the proposed method did not work effectively. If the search task itself was too difficult, or conversely, too easy, all our methods were relatively ineffective. For instance, in the task of finding a place suitable for eating pizza, it was possible to find a large number of places using conventional methods. In such cases, finding more places by inference has conversely reduced accuracy.

Finally, individual cases will be discussed. An example where the expansion by the **Place type** deduction worked properly is the search task of "Buy Computer". In this task, our method deduced that you can buy a computer at an electronics store. Even though a store has no reviews, our method was able to guess that the store sells computers by using place type metadata.

Similarly, the extension by place type was highly accurate for the query "have a BBQ". The search results of the traditional method showed a lot of noise, such as "purchased BBQ sauce flavored food". Inference by place types, such as barbecue sites or campgrounds, was effective. In other words, restaurants offering barbecue sauce-flavored food were ranked lower. because among the places with reviews about BBQ, there were only a few restaurants that offer barbecue sauce-flavored food, and more campgrounds.

For some queries, the proposed method had a lower precision than the baseline method. However, the search results for these queries included places that were not found by traditional methods. For example, for the query "guitar practice", **Baseline** found only three places, all of which were music classes, because only these places contained the query words directly in their reviews. More music classes were found by the **No Expansion** method. In a more extended approach, it was possible to find shops, such as music stores that offered guitar lessons or had a performance space attached to them. In these cases, it was possible to rank more suitable places by combining extensions in both words and places.

Table 1: evaluation result of 5 methods for 9 queries

	All (proposed)		Place Only		Word Only		No Expansion		Baseline	
	p@20	nDCG	p@20	nDCG	p@20	nDCG	p@20	nDCG	p@20	(# found)
Guitar Practice	0.30	0.40	0.35	0.43	0.40	0.54	**0.54**	**0.57**	0.15	4
Buy Computer	**0.45**	**0.59**	**0.45**	**0.59**	0.35	0.38	0.40	0.41	**0.45**	49
Fix Computer	0.70	0.76	**0.75**	**0.79**	0.70	0.76	**0.75**	**0.79**	0.65	13
Eat Pizza	0.75	0.64	0.80	0.68	0.85	**0.84**	0.80	0.81	**0.95**	466
Buy Pizza	**0.80**	**0.87**	0.75	0.84	0.70	0.68	0.70	0.68	0.65	32
Catch a Fish	**0.25**	0.27	**0.25**	0.28	**0.25**	**0.35**	**0.25**	0.32	**0.25**	23
Have a BBQ	0.70	0.66	**0.75**	**0.68**	0.60	0.58	0.50	0.48	0.30	124
Enjoy Afternoon Tea	**0.90**	**0.94**	**0.90**	**0.94**	**0.90**	0.79	0.80	0.76	0.75	91
Swimming	0.05	0.03	0.20	0.14	0.15	0.10	**0.25**	**0.21**	0.20	78
Average	0.54	0.57	**0.58**	**0.60**	0.54	0.56	0.54	0.56	0.48	-

From these results, the extended method that applied only place type-based inference had the highest performance. However, it can be said that each extension has different strengths.

6 CONCLUSION

In this research, we proposed a new search algorithm that ranks the places that can achieve a given purpose. In a conventional retrieval system, searchers have to input the type of business and the characteristics of the place to be searched as a query. This makes it difficult to find a place, such as a place for "guitar practice", by objective. Therefore, by using geographical review information such as Google Maps, we made the search system able to accept the purpose directly. By extending it with three types of hypotheses, searchers can search for places by inputting their purpose. We implemented a web application based on the Random Walk with Restart-based graph analysis method. The experimental result shows that our method can find more suitable places than existing place search methods.

As a future challenge, an increase in the accuracy of the search results is needed. Also, the amount of calculation is another important problem. Our method requires the creation of a graph and convergence calculations each time a query is entered. In order to operate the search model as an actual Web service, it is necessary to improve the speed of the service by grouping similar places and purposes in advance. In the future, it is necessary to conduct more advanced research to realize such a search as an actual web service.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grants Number 18K18161, 21H03775 and 18H03243.

REFERENCES

[1] Ramesh Baral, XiaoLong Zhu, S. S. Iyengar, and Tao Li. 2018. ReEL: Review Aware Explanation of Location Recommendation. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization* (Singapore, Singapore) *(UMAP '18)*. Association for Computing Machinery, New York, NY, USA, 23–32.

[2] Sandro Bauer, Filip Radlinski, and Ryen W White. 2016. Where Can I Buy a Boulder?: Searching for Offline Retail Locations. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1225–1235.

[3] Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. 2005. Finding Similar Questions in Large Question and Answer Archives. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management* (Bremen, Germany) *(CIKM '05)*. Association for Computing Machinery, New York, NY, USA, 84–90. https://doi.org/10.1145/1099554.1099572

[4] Shuhui Jiang, Xueming Qian, Jialie Shen, Yun Fu, and Tao Mei. 2015. Author Topic Model-Based Collaborative Filtering for Personalized POI Recommendations. *IEEE Transactions on Multimedia* 17, 6 (2015), 907–918. https://doi.org/10.1109/TMM.2015.2417506

[5] Makoto P. Kato, Satoshi Oyama, Ohshima Hiroaki, and Katsumi Tanaka. 2010. Query by Example for Geographic Entity Search with Implicit Negative Feedback. In *Proceedings of the 4th International Conference on Uniquitous Information Management and Communication* (Suwon, Republic of Korea) *(ICUIMC '10)*. Association for Computing Machinery, New York, NY, USA, Article 45, 10 pages. https://doi.org/10.1145/2108616.2108671

[6] Takeshi Kurashima, Taro Tezuka, and Katsumi Tanaka. 2005. Blog map of experiences: Extracting and geographically mapping visitor experiences from urban blogs. In *International Conference on Web Information Systems Engineering*. Springer, 496–503.

[7] Barak Pat, Yaron Kanza, and Mor Naaman. 2015. Geosocial search: Finding places based on geotagged social-media posts. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 231–234.

[8] Suppanut Pothirattanachaikul, Takehiro Yamamoto, Sumio Fujita, Akira Tajima, Katsumi Tanaka, and Masatoshi Yoshikawa. 2018. Mining Alternative Actions from Community Q&A Corpus. *Journal of Information Processing* 26 (2018), 427–438.

[9] Yoshiyuki Shoji, Katsurou Takahashi, Martin J Dürst, Yusuke Yamamoto, and Hiroaki Ohshima. 2018. Location2vec: Generating distributed representation of location by using geo-tagged microblog posts. In *International Conference on Social Informatics*. Springer, 261–270.

[10] Kristin Stock. 2018. Mining location from social media: A systematic review. *Computers, Environment and Urban Systems* 71 (2018), 209–240.

[11] Hao Wang, Manolis Terrovitis, and Nikos Mamoulis. 2013. Location recommendation in location-based social networks using user check-in data. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 374–383.

[12] Kai Wang, Zhaoyan Ming, and Tat-Seng Chua. 2009. A Syntactic Tree Matching Approach to Finding Similar Questions in Community-Based Qa Services. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Boston, MA, USA) *(SIGIR '09)*. Association for Computing Machinery, New York, NY, USA, 187–194. https://doi.org/10.1145/1571941.1571975

[13] Yihong Zhang, Panote Siriaraya, Yukiko Kawai, and Adam Jatowt. 2020. Automatic latent street type discovery from web open data. *Information Systems* 92 (2020), 101536.