A Tourist Spot Search Method Using Similarity of Function Based on Distributed Representations of User Reviews

Tomofumi Yoshida, Daisuke Kitayama, Shinsuke Nakajima and Kazutoshi Sumiya

***Abstract*—Tourist spots have certain functions, such as "spot suitable for seeing the night view" or "spot suitable for meeting point." In this paper, we propose a method for searching tourist spots that uses the similarity of their functional features.**

In our method, we extract distributed representations of a tourist spot as a paragraph vector of user reviews for the tourist spot. We extract distributed representations of location and category in the same way. Next, we extract the functional feature of a tourist spot by combining distributed representations of the tourist spot, locations, and categories. Finally, we search tourist spots using the extracted functional feature and distributed representations of other locations and categories.

In this phase, the most important thing is the way in which locations and categories are represented. We can employ an extraction method using a paragraph vector for user reviews based on a location or a category (the conventional method). On the other hand, we can use the average vector of distributed representations of all spots in a location or a category. In this paper, we compared our method (average vector) with the conventional method (paragraph vector). By this experiment, we evaluated the effectiveness of our method for searching tourist spots.

***Index Terms*—Tourist information, Distributed representations, User reviews**

I. Introduction

In recent years, tourist information Web sites such as TripAdvisor [1] and Jalan [2] have come to be widely used. Tourists can post user reviews of a tourist spot on these Web sites. Therefore, there is much tourist information and many user reviews on the Web. In general, tourist spots have metadata of location tags and category tags on tourist information Web sites. Users can search tourist spots using keywords and these tags to narrow down the list of spots. When wishing to see a historic spot in Kyoto, Japan, most users select the "Kyoto" tag and the "Shrine and Temple" tag.

However, not all tourist requirements are represented by the metadata given to the spot. In other words, users want to search using features other than location and category. When a user is planning a trip with the theme of night viewing, "Tokyo Tower" and "Mt. Hakodate" are suitable spots. "Tokyo Tower" is a communications and observation tower in Tokyo, Japan, and it is one of the most famous night viewing spots in Japan. "Mt. Hakodate" is a mountain in Hokkaido, Japan; it too is one of the most famous night viewing spots in Japan. These spots have the same function, "spot suitable for seeing a night view," although the location tags and category tags of these spots differ.

We call this function of a spot its functional feature. The functional feature does not depend on location and category. For example, shrines and temples are generally in quiet places. However, the shrines and temples category does not indicate the function "he/she can be quiet and calm." In fact, shrines and temples are also present in downtown areas. Therefore, the conventional search method using location tags and category tags is insufficient for narrowing the search by functional feature.

In the above way, we focus on the functional features of tourist spots, which are not extracted from the given metadata and tags. We propose a method for searching tourist spots based on the similarity of their functional features. By this method, a user can search for a suitable spot simply by selecting "well-known" as the spot's functional feature even if the user wants to find spots in an area unfamiliar to the user.

In this paper, we use the paragraph vector model, an efficient method for vectorizing sentences and documents, on user reviews of each tourist spot to generate distributed representation (paragraph vector) of each tourist spot. We extract distributed representations of location tag and category tag in the same way. Then, we generate a query vector that shows the functional feature of a tourist spot by following steps. First, we subtract the location vector and category vector from the target spot vector; this generated vector is considered to represent the functional feature of the spot. Next, the user selects the search location and category. Finally, we add the location vector and the category vector selected by the user to the vector generated in the first step.

In this way, we can generate a query vector for searching for spots that have the same functional feature but in other locations and categories. For example, if a user would like to find a spot in New York having the same functional feature "point suitable for meeting" as "Hachiko," a famous statue of a dog in Tokyo, our method generates a query vector such as $Hachiko - Tokyo + New\ York$, and then calculates the cosine similarity between the query vector and the paragraph vector for each tourist spot. In an ideal retrieval result, our method will retrieve "Grand Central Terminal Clock," a famous meeting point in New York. Figure 1 shows an overview of our proposed search method.

The remainder of this paper is organized as follows. Use

Manuscript received January 8, 2017; revised February 3, 2017.

T. Yoshida is with the Graduate School of Engineering, Kogakuin University, Japan; e-mail: em15020@ns.kogakuin.ac.jp.

D. Kitayama is with the Faculty of Informatics, Kogakuin University, Japan; e-mail: kitayama@cc.kogakuin.ac.jp.

S. Nakajima is with the Faculty of Computer Science and Engineering, Kyoto Sangyo University, Japan; e-mail: nakajima@cc.kyoto-su.ac.jp.

K. Sumiya is with the School of Policy Studies & Research Center for Social Informatics, Kwansei Gakuin University, Japan; e-mail: sumiya@kwansei.jp.

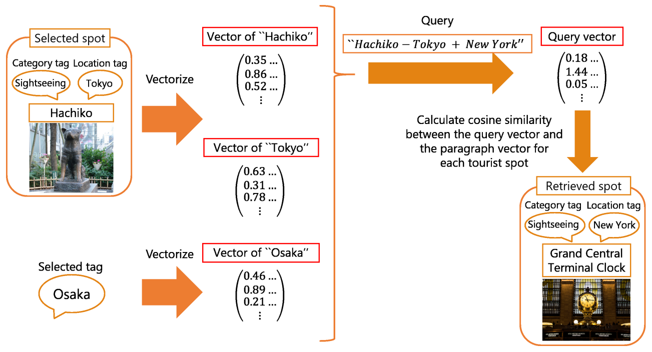

Fig. 1. Overview of our proposed search method

of the paragraph vector model for tourist spots and its problems are presented in Section II. Section III describes the proposed method and an evaluation experiment. Related work is mentioned in Section IV. We conclude the paper in Section V.

II. Use of Paragraph Vector Model for Tourist Spots and Its Problems

A. Paragraph vector for tourist spot

The paragraph vector model was proposed by Le and Mikolov [3]. The idea for this model was based on the study by Mikolov et al. [4], known as word2vec [5]. Based on the distributional hypothesis given by Harris in 1954 [6], the paragraph vector model learns fixed-length feature representations, called paragraph vectors, from variable-length texts such as sentences, paragraphs, and documents by using a neural network. In this paper, we denote the minimum unit of learning data for the paragraph vector model as a "document." Thus, the paragraph vector model learns the distributed representations of each document.

Prior to evaluating our proposed method, we first analyzed how the paragraph vector for each tourist spot represents the feature of the corresponding tourist spot. We hypothesized that the paragraph vector for a given tourist spot represents the following three features of that tourist spot: location feature, category feature, and functional feature. Thus, we examined whether the similarity between the paragraph vectors for tourist spots that have similar location features or category features increased regardless of the similarity of their functional features by preliminary experiment 1, described in Section II-B. Secondly, we analyzed the paragraph vector for the location tag and category tag by preliminary experiment 2, described in Section II-C. On the basis of the results of the two preliminary experiments, we propose a method for calculating the paragraph vector for the location tag and category tag by using the average vector of the paragraph vectors for all spots that have the same location tag or the same category tag.

In addition, we describe the setup for our preliminary experiments. Table I shows the dataset for the preliminary experiments, which was extracted from Jalan, one of the leading tourist information Web sites in Japan. Each tourist spot has four location tags, two category tags, and user reviews. We used these user reviews as training data for the paragraph vector model. Furthermore, to generate the paragraph vectors, we used the implementation of the paragraph

TABLE I
Jalan Dataset

Number of spots	43,759
Number of reviews	1,481,831
Types of location tag	Prefecture, Area, City, Town
Types of category tag	Parent category, Child category

TABLE II
Package Information and Learning Parameters

Version of Python	3.5.2
Version of gensim	0.12.4
Learning model	DBOW
Window size	8
Number of dimensions	300

vector model by gensim [7], an open-source natural language processing toolkit implemented in the Python language. Table II shows the environment and parameter settings for gensim. Values of parameters that are not mentioned in Table II follow gensim's default settings [8].

B. Preliminary experiment 1: Analyzing the paragraph vectors for tourist spots

As the first preliminary experiment, we analyzed how a paragraph vector represents the features of a tourist spot. In this experiment, we treated all user reviews for one tourist spot as one document and generated the paragraph vector for each spot. Table III shows the five tourist spots having the highest cosine similarity between their respective paragraph vectors and the paragraph vector for "Hachiko."

"Hachiko" is one of the most famous meeting points in Japan. This functional feature is hard to guess from its location tags, "Tokyo" and "Shibuya," and the category tags, "Sightseeing" and "Historic Sites". In this table, several famous meeting points in Japan such as "Moyai Statue," "Saigo Takamori Statue," and "Umeda BIGMAN" scores a high cosine similarity to "Hachiko". Many user reviews of these spots mention the reviewers' meeting experiences; therefore, the paragraph vector model generated distributed representations of these spots that contained the functional feature "point suitable for meeting" from the user review contexts.

However, many user reviews also mention the location feature and category feature of the tourist spot; thus, the cosine similarity between the paragraph vector for "Hachiko" and the paragraph vector for other spots also increases when their location features or category features are similar to each other. For example, "Moyai Statue," as well as "NANAKO Statue" and "QFRONT" (which, although not displayed in Table III, came in sixth and ninth place, respectively, in cosine similarity with "Hachiko"), are located near the Shibuya Station, which is also located near "Hachiko." Moreover, many of the category tags in Table III are "Sightseeing" or "Historic Sites," the same category tags as for "Hachiko." On the basis of the above, we confirm our first hypothesis that the similarity between paragraph vectors for tourist spots that have similar location features or category features increases regardless of the similarity of their functional features.

TABLE III
TOURIST SPOTS WITH A HIGH SIMILARITY TO "STATUE OF HACHIKO"

Name of spot	Similarity	Location 1	Location 2	Location 3	Location 4	Category 1	Category 2
Statue of Hachiko	-	Tokyo	Shibuya area	Shibuya-ward	Dogenzaka	Sightseeing	Historic Sites
Moyai Statue	0.7825	Tokyo	Shibuya area	Shibuya-ward	Dogenzaka	Sightseeing	Tourist facilities
Hachiko Family Relief	0.6197	Tokyo	Shibuya area	Shibuya-ward	Dogenzaka	Others	Noted place
Saigo Takamori Statue	0.5719	Tokyo	Asakusa area	Taito-ward	Ueno Park	Sightseeing	Historic Sites
Umeda BIGMAN	0.4992	Osaka	Umeda area	Osaka city	Shibata Kita-ku	Sightseeing	Tourist facilities
Statue of Hachiko (Odate station)	0.4675	Akita	Towada area	Odate city	Onari-cho	Sightseeing	Historic Sites

TABLE IV
TOURIST SPOTS WITH A HIGH SIMILARITY TO THE QUERY VECTOR "HACHIKO - TOKYO SPECIAL WARDS + OSAKA CITY"

Name of spot	Similarity	Location	Category
Moyai Statue	0.5192	Tokyo special wards	Sightseeing
Universal Studios Japan	0.5047	Osaka city	Amusement
Hachiko Family Relief	0.3680	Tokyo special wards	Others
Umeda BIGMAN	0.3530	Osaka city	Sightseeing
Saigo Takamori Statue	0.3487	Tokyo special wards	Sightseeing

C. Preliminary experiment 2: Analyzing the paragraph vectors for location tags and category tags

As the second preliminary experiment, we analyzed the paragraph vectors for location tags and category tags. In this experiment, we treated all user reviews of all tourist spots having the same tag as one document for each tag and generated the paragraph vector for each tag. For this, we treated "location 3" as the location tag and "category 1" as the category tag for each spot. For example, when we generated the paragraph vector for "Kyoto city," we treated all reviews of spots whose location tag was "Kyoto city" as a "Kyoto city" document and generated a distributed representation of "Kyoto city" using the paragraph vector model. As an exception, we treated special wards of Tokyo such as "Shibuya-ku" and "Chiyoda-ku" as one location tag, "Tokyo special wards."

We analyzed whether these paragraph vectors for location tags and category tags are efficient for generating query vectors. Table IV shows the five tourist spots having the highest cosine similarity between their respective paragraph vectors and the query vector $Hachiko - Tokyo\ special\ wards + Osaka\ city$. In this table, a famous amusement park in Osaka city, "Universal Studios Japan," scores a high cosine similarity to the query vector, although an amusement park is not a typical spot for a meeting point. The main reason for this result is that the cosine similarity between the paragraph vector for "Osaka city" and that for "Universal Studios Japan" is 0.7893, the highest similarity to "Osaka city" of any of the spots listed. Therefore, the cosine similarity between the query vector and "Osaka city" has increased. On the basis of the above, we confirm that treating all user reviews of all tourist spots having the same tag as one document does not generate an efficient query vector.

III. EXPERIMENT

We compared extracted tourist spots using query vectors generated by the proposed method and by some simple methods, and we examined the characteristics of each method. We generated paragraph vectors in the same way as for preliminary experiment 1 (Section II-B). We used the following three methods for generating query vectors.

Filter method: We extracted tourist spots that have high cosine similarity to the input spot and added the same location/category tags.

Direct method: We generated a paragraph vector for location/category tags using a set of reviews for a spot having the exact same target location/category tags. This is the same method used in preliminary experiment 2 (Section II-C).

Proposed method: We generated a paragraph vector for location/category tags by calculating the average of the paragraph vectors for spots that have the target location/category tags. This method is explained in Section II-A.

We used two queries, the first one to assess the effectiveness of the location tag and the second one to assess the effectiveness of the category tag:

- $Shinjuku\ Gyoen\ National\ Garden - Tokyo\ special\ wards + Osaka\ city$
- $Meiji\ Shrine - Shrine + Sightseeing$

A. Query 1: Shinjuku Gyoen National Garden - Tokyo special wards + Osaka city

We used $Shinjuku\ Gyoen\ National\ Garden - Tokyo\ special\ wards + Osaka\ city$ as the generated query vector based on location tags. We call this Q_1. "Shinjuku Gyoen National Garden" has "Tokyo special wards" as its location tag and "Sightseeing" as its category tag. With this query, we expect that a nature-rich park in Osaka city will be extracted. Table V shows the five spots having the highest cosine similarity to "Shinjuku Gyoen National Garden." Tables VI, VII, and VIII show the top five spots in the results based on the respective methods.

First, in examining Tables VII and VIII, we observe that nature-rich parks around Osaka such as "Osaka Castle Nishinomaru Garden" and "Kyoto Prefectural Botanical Garden" in the results for the proposed method are ranked more highly than the results for the direct method. Therefore, we consider that the proposed method generates a paragraph vector for location tags that is more suitable than that of the direct method. However, nature parks around Tokyo such as "Cherry Blossoms at Shinjuku Gyoen National Garden" and "Imperial Palace East Gardens" remain in the results for the proposed method. The norm of the paragraph vector for location tags generated by the proposed method tends to be small because it is averaged (we can say that the same is true for category tags). Therefore, we expected that the contribution of the generated paragraph vector for

TABLE V
TOP FIVE SPOTS BASED ON COSINE SIMILARITY TO "SHINJUKU GYOEN NATIONAL GARDEN"

Name of spot	Similarity	Location	Category
Cherry Blossoms at Shinjuku Gyoen National Garden	0.7022	Tokyo special wards	Animal/Plant
Imperial Palace East Gardens	0.4939	Tokyo special wards	Sightseeing
Kyoto Prefectural Botanical Garden	0.4774	Kyoto city	Sightseeing
Yoyogi Park	0.4339	Tokyo special wards	Sightseeing
Otani Natural Park	0.4306	Yamato Takada city (Nara)	Sightseeing

TABLE VI
TOP FIVE SPOTS BASED ON COSINE SIMILARITY TO Q_1 USING FILTER METHOD

Name of spot	Similarity	Location	Category
Osaka Castle Nishinomaru Garden	0.4003	Osaka city	Sightseeing
Nagai Botanical Garden	0.3500	Osaka city	Sightseeing
Nakanoshima Rose Garden	0.3284	Osaka city	Sightseeing
Fujita Manor Park	0.3224	Osaka city	Sightseeing
Tsurumi Ryokuchi	0.3214	Osaka city	Sightseeing

TABLE VII
TOP FIVE SPOTS BASED ON COSINE SIMILARITY TO Q_1 USING DIRECT METHOD

Name of spot	Similarity	Location	Category
Universal Studios Japan	0.5409	Osaka city	Amusement
Cherry Blossoms at Shinjuku Gyoen National Garden	0.3448	Tokyo special wards	Animal/Plant
Universal City Walk Osaka	0.2958	Osaka city	Others
Dahlia of Kurohime Highlands	0.2642	Shinanomachi (Nagano)	Animal/Plant
Cherry Blossoms at Akashi Park	0.2621	Akashi city	Animal/Plant

location/category tags to the query vector would be small. In almost all cases, the results for the proposed method are a fine adjustment of the results according to similarity (see Tables V and VIII). We need to develop a suitable method to amplify the generated paragraph vector for location/category tags.

Second, the filter method was able to extract some nature-rich parks in Osaka such as "Nagai Botanical Garden" and "Nakanoshima rose garden," which are similar to "Shinjuku Gyoen National Garden" (see Table VI). These spots were not able to be extracted by the other methods. The similarity of the paragraph vector for the spot itself includes the similarity of its functional feature. As a result, the filter method with location tags went very well (see Table VI). When a user can select location tags appropriately, we should use the filter method. On the other hand, when a user wants to search for spots on or around selected location tags, our method can extract tourist spots at selected location tags and around the selected location.

B. Query 2: Meiji Shrine - Shrine + Sightseeing

We used $Meiji\ Shrine - Shrine + Sightseeing$ as the generated query vector based on category tags. We call this Q_2. Meiji Shrine has Tokyo special wards as its location tag

TABLE VIII
TOP FIVE SPOTS BASED ON COSINE SIMILARITY TO Q_1 USING PROPOSED METHOD

Name of spot	Similarity	Location	Category
Cherry Blossoms at Shinjuku Gyoen National Garden	0.6754	Tokyo special wards	Animal/Plant
Kyoto Prefectural Botanical Garden	0.4791	Kyoto city	Sightseeing
Imperial Palace East Gardens	0.4564	Tokyo special wards	Sightseeing
Osaka Castle Nishinomaru Garden	0.4370	Osaka city	Sightseeing
Otani Natural Park	0.4123	Yamato Takada city (Nara)	Sightseeing

and Shrine as its category tag. With this query, we expect that historical spots with tourist facilities will be extracted rather than a place of worship.

Table IX shows the five spots having the highest cosine similarity to "Meiji Shrine." Tables X, XI, and XII show the top five spots in the results based on the respective methods. According to the results in Table X, the spots extracted by the filter method (such as "Auxiliary Shrine of Ishigami Jingu" and "Mausoleum of Emperor Jinmu") do not have the category "Shrine." However, they are spots that are practically equivalent to shrines.

We considered that these inappropriate results were due to the adding of the feature tag "Sightseeing." In general, the category to which a spot belongs cannot be determined uniquely, and there are many spots that can belong to a plurality of category tags. Therefore, it is difficult to effectively extract spots using the filter method based on category tags.

On the other hand, as seen in Table XI, the paragraph vector for the category tag learned by the direct method extracted tourist attractions near "Meiji Shrine," such as "Yoyogi Park." In addition, "Nogi Shrine," located in Nogizaka near "Meiji Shrine," was extracted. These spots frequently include reviews such as "I walked from Harajuku to Akasaka by accident," meaning that they visited sightseeing spots near other sightseeing spots. Therefore, we believe that the generated paragraph vector represents the meaning of the category tags.

The results for the proposed method are similar to the cosine similarity results (see Tables IX and XII). This has the same tendencies as in the case of the location tag. Although they are not displayed in Table XII, the similarity of spots near "Meiji Shrine" and highly related to "Sightseeing" tags such as "Yoyogi First Gymnasium" and "Nogi Shrine" have risen, in addition to "Yoyogi Park." That is, only the spots related to the added/subtracted tag have changed. We regard this as a more rigorous result compared with the direct method.

Experimental results show that the query vector using the paragraph vector for the location/category tag generated by the proposed method has the following three features.

- It is particularly effective in cases that include tags that are difficult to classify uniquely, such as category tags.
- The paragraph vectors for the location/category tag based on average vectors make a small contribution to the query vector. Therefore, we need to develop an amplification method for the paragraph vectors for the location/category tag.

TABLE IX
TOP FIVE SPOTS BASED ON COSINE SIMILARITY TO "MEIJI SHRINE"

Name of spot	Similarity	Location	Category
Meiji Shrine Inner Garden	0.5512	Tokyo special wards	Sightseeing
Atsuta Shrine	0.5034	Nagoya city	Shrine
Miyazaki Shrine	0.4855	Miyazaki city	Shrine
Hokkaido Shrine	0.4761	Sapporo city	Shrine
Yajima Shrine	0.4377	Miyazaki city	Shrine

TABLE X
TOP FIVE SPOTS BASED ON COSINE SIMILARITY TO Q_2 USING FILTER METHOD

Name of spot	Similarity	Location	Category
Meiji Shrine Inner Garden	0.5512	Tokyo special wards	Sightseeing
Auxiliary Shrine of Ishigami Jingu	0.3704	Tenri city	Sightseeing
Yoyogi Park	0.3589	Tokyo special wards	Sightseeing
Mausoleum of Emperor Jinmu	0.3535	Kashihara city (Nara)	Sightseeing
Kashihara Forest Garden	0.3417	Kashihara city (Nara)	Sightseeing

TABLE XI
TOP FIVE SPOTS BASED ON COSINE SIMILARITY TO Q_2 USING DIRECT METHOD

Name of spot	Similarity	Location	Category
Meiji Shrine Inner Garden	0.4704	Tokyo special wards	Sightseeing
Nogi Shrine	0.3256	Tokyo special wards	Shrine
Cornerstone of Peace Memorial	0.3227	Itoman city	Others
Yoyogi Park	0.3210	Tokyo special wards	Sightseeing
Dr. Hebron House Memorial	0.3057	Yokohama city	Sightseeing

TABLE XII
TOP FIVE SPOTS BASED ON COSINE SIMILARITY TO Q_2 USING PROPOSED METHOD

Name of spot	Similarity	Location	Category
Meiji Shrine inner garden	0.5389	Tokyo special wards	Sightseeing
Atsuta Shrine	0.4291	Nagoya city	Shrine
Hokkaido Shrine	0.4193	Sapporo city	Shrine
Yoyogi Park	0.4168	Tokyo special wards	Sightseeing
Miyazaki Shrine	0.4103	Miyazaki city	Shrine

- It is possible to extract only spots that are highly relevant to the target tags via addition and subtraction using the generated paragraph vectors.

IV. Related Work

There are many studies on tourism information extraction. First, we mention a method for automatically extracting sightseeing spots using geotagged content posted on a location-based social network service. Crandall et al. [9] proposed a method for extracting a popular spot and landmark. They extracted clusters as a spot by using the large number of photos, the spot's location information, and social tags. Moreover, they showed that the trajectory of a photography route could be obtained from data of the same photographer. Hirota et al. [10] proposed visualization of multiple points for which there are considerable numbers of geotagged photos, and extraction of a landmark's shape by using the shooting directions. Further, Oku and Hattori [11] proposed a method to extract the features of a sightseeing spot based on the tweets in that area. Next, we mention the expansion method for extracting tourist information. Fujii et al. [12] are working on a method to extract information based on real experience from blogs and community Q&As, information that is difficult to know using guidebooks alone. In this study, we used data from a tourist review site, but with our method it seems that it would be possible to generate a paragraph vector from various text data and tag information attached to extracted spots by conventional methods.

There has been research on relative retrieval and analogy retrieval, such as finding content corresponding to one domain in another domain. Nakajima and Tanaka [13] proposed a method of searching by using the relative relationship between two domains by mapping the space of one domain onto the other domain. In addition, Kato et al. [14] focused on two target keywords and a keyword representing their relationship. They proposed a method of automatically extracting the relationship keyword from a Web search index and a method of finding other corresponding keywords using the relationship keyword. These studies are similar to our research in that they are methods for discovering equivalent tourist spots in other locations and categories. In this research, we used a paragraph vector to carry it out and focused on the functional feature.

V. Conclusion

We have proposed a tourist spot search method using similarity of function based on distributed representations of user reviews. We generated a paragraph vector for tourist spots using user reviews on a tourist review site, and we generate a paragraph vector for location/category tags based on the paragraph vector for the spot. In this way, we made it possible to calculate spots and locations/categories. We assessed the characteristics of the generated paragraph vector for a spot by preliminary experiments. We can say that the generated paragraph vector for a spot represents the spot's functional feature; at the same time, however, it includes location features and category features. Therefore, we experimented with a method to generate a paragraph vector for location/category that appropriately subtracts location features and category features from the paragraph vector for a spot. The results confirmed the effectiveness of the proposed method using the average vector for the corresponding spot as the paragraph vector for the location/category tags.

As our future work, we need to develop a suitable method to amplify the generated paragraph vector for location/category tags because the contribution of the generated paragraph vector for location/category tags to the query vector is small. In addition, we will extend the paragraph vector generating method to treat not only location/category tags but any keyword.

Acknowledgment

This work was supported by MEXT KAKENHI of Grant-in-Aid for Scientific Research (B) Grant Number 26280042.

References

[1] www.tripadvisor.com, one of the most famous travel website around the world.
[2] www.jalan.net, one of the largest travel website in Japan.

[3] Q. V. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*, 2014, pp. 1188–1196.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems 26*, 2013, pp. 3111–3119.

[5] code.google.com/archive/p/word2vec, implementation of the algorithm proposed by Mikolov et al.

[6] Z. S. Harris, “Distributional structure,” *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[7] R. Řehůřek and P. Sojka, “Software Framework for Topic Modelling with Large Corpora,” in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 2010, pp. 45–50. [Online]. Available: http://is.muni.cz/publication/884893/en

[8] www.radimrehurek.com/gensim/models/doc2vec.html, gensim’s API Reference.

[9] D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg, “Mapping the world’s photos,” in *Proceedings of the 18th International Conference on World Wide Web*, 2009, pp. 761–770.

[10] M. Hirota, M. Shirai, H. Ishikawa, and S. Yokoyama, “Detecting relations of hotspots using geo-tagged photographs in social media sites,” in *Proceedings of Workshop on Managing and Mining Enriched Geo-Spatial Data*, 2007, pp. 7:1–7:6.

[11] K. Oku and F. Hattori, “Mapping geotagged tweets to tourist spots considering activity region of spot,” in *Tourism Informatics*. Springer Berlin Heidelberg, 2015, vol. 90, pp. 15–30.

[12] K. Fujii, H. Nanba, T. Takezawa, and A. Ishino, “Enriching travel guidebooks with travel blog entries and archives of answered questions,” in *Information and Communication Technologies in Tourism 2016: Proceedings of the International Conference in Bilbao, Spain, February 2-5, 2016*, A. Inversini and R. Schegg, Eds. Springer International Publishing, 2016, pp. 157–171.

[13] S. Nakajima and K. Tanaka, “Relative queries and the relative cluster-mapping method,” in *Database Systems for Advanced Applications: 9th International Conference, DASFAA 2004, Jeju Island, Korea, March 17-19, 2003. Proceedings,*, Y. Lee, J. Li, K.-Y. Whang, and D. Lee, Eds. Springer Berlin Heidelberg, 2004, pp. 843–856.

[14] M. P. Kato, H. Ohshima, S. Oyama, and K. Tanaka, “Query by analogical example: Relational search using web search engine indices,” in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, 2009, pp. 27–36.