POI Atmosphere Categorization Using Web Search Session Behavior

Kota Tsubouchi*
Yahoo Japan Corporation
Tokyo, Japan
ktsubouc@yahoo-corp.jp

Hayato Kobayashi*
Yahoo Japan Corporation
Tokyo, Japan
hakobaya@yahoo-corp.jp

Toru Shimizu*
Yahoo Japan Corporation
Tokyo, Japan
toshimiz@yahoo-corp.jp

ABSTRACT

Point Of Interest (POI) categorization is to group POIs into several categories and make them easy-to-use in geospatial applications. Previous studies mainly used geospatial features, such as check-in sequences and satellite images, to group POIs into pre-defined rough categories. However, each POI has its own "atmosphere" beyond its geospatial features, which represents what kinds of people tend to visit it and how they spend their time there. This subtle atmosphere is important for users to decide whether to visit the POI, so considering it may be critical when providing commercial services, such as a property search service. In this paper, we propose a new POI categorization method that can capture the POI atmosphere by using user behavior on a web search engine. Our key observation is that the next queries of a search query about a POI tend to contain the user's purpose for visiting it. We harness this observation to train a neural encoder that maps POIs to continuous vectors (called embeddings) via next-query prediction with a deep structured semantic model (DSSM). Experimental results indicate that our method performs well for POI atmosphere categorization of parks as a case study. We believe that our method complements the existing POI categorization methods.

CCS CONCEPTS

• **Information systems** → **Spatial-temporal systems**; *Web mining*; • **Computing methodologies** → **Learning latent representations**.

KEYWORDS

POI Representations, Atmosphere Categorization, Geographic Knowledge Discovery, Web Search Session Behavior

1 INTRODUCTION

Imagine that there are two parks almost the same size in a residential area. One park has a lot of benches and cigarette butts strewn about, and many office workers gather there at lunch time. The other park has table-tennis tables on the lawns. Elderly people and homemakers play table tennis at lunch time, and children also play there after school. These two parks clearly have different atmospheres (i.e., "business-like" versus "family-like"), but they will be categorized into the same category (i.e., "parks located in a residential area") by existing POI (point of interest) categorization methods, which group POIs into pre-defined rough categories with geospatial features such as check-in sequences and satellite images. This problem is critical for commercial services such as property-search services, as shown in Figure 1. This figure shows a

Figure 1: Example of a property search service, as an application of POI categorization.

possible scenario where a user is searching for a property near a park because they want to frequently play with their children there. In this scenario, if park B has a "business" atmosphere, and park A is the nearest park with the "family" atmosphere, this result is completely misleading for such a user, since the property is actually far from the intended "park."

In this paper, we address POI categorization that can distinguish the above-mentioned *atmospheres*. Our approach to capturing the POI atmosphere is to leverage user behavior on a web search engine to train a neural encoder that maps POIs to continuous vectors (called embeddings), which will be used for POI categorization. Nowadays, people tend to get information from a search engine no matter what they do, and POIs are no exception. For example, if a user wants to go to a park for the purpose of playing with their children on a playground equipment, they will post a series of queries (called search session) such as "[PARK NAME] playground equipment" to learn the details of the equipment and "playground equipment safety" to confirm the basic safety of the equipment, as well as "[PARK NAME]" to obtain overall information about the park. Such search sessions on a POI must contain enough information to represent the POI atmosphere, because most people have prior knowledge of the POI and only post its relevant queries.

We use a deep structured semantic model (DSSM) [11] to train the neural encoder with the search session behavior. DSSM is a variant of the latent semantic model based on deep neural networks, which was originally proposed in the information retrieval field and has been practically used in commercial search engines to obtain relevant documents given a search query. The core idea of DSSM is to train an encoder that maps both queries and documents into a semantic (embedding) space so that each query is close to its relevant documents. This training is conducted in a supervised fashion with many pairs of a query and its relevant document. Although we do not have any explicit supervised information for

*These authors contributed equally.

our task, we solve this problem by regarding the search session behavior as implicit supervised information under the assumption that the next query of a query about a POI tends to represent the POI atmosphere. In other words, we regard the next queries as the relevant documents and train the encoder of DSSM with a lot of pairs of a query and its next query. This approach is expected to perform better than using a skip-gram model, which is often used in the existing POI embedding methods such as POI2vec [2], since DSSM is more expressive than the skip-gram model, and its usefulness has been confirmed in the information retrieval field. Furthermore, DSSM can basically handle any queries if we use a character-based encoder, whereas the skip-gram model can only handle words (or sequences of words) in a pre-defined dictionary.

Our approach is a new one compared to the ones taken by previous studies on POI categorization or representation. In particular, the previous studies represented POIs in mainly two ways. One way uses user behavior in a geographical space [2, 25] (e.g., sequence of POIs a user visited), while the other uses geospatial features of a POI [6, 18, 23] (e.g., a satellite image of the POI). These two approaches will not be enough for our task since their resources do not directly contain information indicating the atmosphere of the POI such as "why users visit there," although they are useful for ordinary POI categorization. Considering atmosphere of the POI is critical for commercial services, as mentioned above, so our study will help put the findings of the previous studies into practical use since POI embeddings obtained by our approach can be easily used with the existing ones. There have been few studies purely based on linguistic resources similar to ours, because it has been considered that ordinary word embeddings lack the ability to capture geographic features of POIs, as pointed by Yan et al. [23]. We show empirical evidence that search session behavior can also capture several geographic features of POIs, such as area and functionality, so our study will be important as a bridge between two different fields: geographic information processing and natural language processing.

The contributions of this paper are summarized as follows:

- We propose a new POI categorization method that can represent atmosphere of a POI beyond its geospatial features with utilizing user behavior on a web search engine. Our method uses next-query prediction in the DSSM framework to train a neural encoder that maps POIs into real-valued embeddings.
- We demonstrate the effectiveness of our method through a case study on POI atmosphere categorization of parks. The experimental results show that our POI embeddings correspond to the real world and are better than ordinary word embeddings (by word2vec) in terms of separability.

The rest of this paper is organized as follows. In Section 2, we explain how to prepare the dataset based on search session behavior and how to train the neural encoder based on DSSM. In Section 3, we describe the case study on POI categorization of parks to confirm the effectiveness of our method. In Section 4, we describe the related work in POI embedding and query classification technologies. In Section 5, we conclude with a summary of the key points.

2 PROPOSED METHOD

2.1 System Overview

Figure 2 shows an overview of our POI atmosphere categorization system, which consists of two parts: training and inference. The training part makes a query encoder that maps search queries to embeddings in Euclidean space, where the queries may include POI names. The query encoder is based on a recurrent neural network and trained by next query prediction in a deep structured semantic model (DSSM) framework [11]. We will explain the data preparation for training in Section 2.2 and give an overview of DSSM training in Section 2.3.

In the inference part, we simply use the query encoder trained in the training part in order to obtain POI embeddings.[1] We only need names (e.g., "New York") since the search engine log basically covers common POI names that can be used for commercial services. Conversely, if a POI name is not used in a general search engine, it will not be used in other specific services. We demonstrate the power of our encoder by addressing POI categorization of parks as a case study in Section 3.

2.2 Data Preparation

Table 1 shows examples of search queries about parks in Japan, sampled from our search engine service, Yahoo! JAPAN. Each block separated by double lines shows consecutive queries (called a search session) about a park, which were posted by a single user in a short period. Our key observation is that we can guess the atmosphere of each park from the next queries without any prior knowledge of the park. For example, we can guess that Yoyogi Park and Shiroyama Park in sessions 1 and 2 may have a family atmosphere. Similarly, we can guess that the parks in sessions 3 and 4 have a river where we can play, and those in sessions 5 and 6 have some scenic places. This observation allows us to think up an idea of training a query encoder so that the next query can be well predicted from the embedding of a query. Below, we describe how to prepare the dataset for this training based on next query prediction.

First, we extract search sessions, each of which consists of consecutive queries posted in the same context, from users' search query histories. The purpose of this extraction is to reduce the noise of predicting the next query that is not relevant to the current one. We assume that queries posted in a short time period have the same context and define a search session as follows.

Definition 2.1 (Search Session). A search session is defined as a consecutive subsequence by splitting the query history of a single user when the difference in posting time between adjacent queries is longer than a given interval δ. Formally, let $H = (Q_1, Q_2, \ldots, Q_{|H|})$ be the query history in chronological order. A consecutive subsequence $(Q_I, \ldots, Q_J)$ of the history H is called a search session if and only if the following conditions hold:

$$t(Q_{i+1}) - t(Q_i) \leq \delta, \qquad (I \leq i < J), \tag{1}$$

$$t(Q_I) - t(Q_{I-1}) > \delta, \tag{2}$$

$$t(Q_{J+1}) - t(Q_J) > \delta, \tag{3}$$

[1] We provide our system as a relatively standard API for several services, where the input is a park name, and the output is the corresponding embedding.

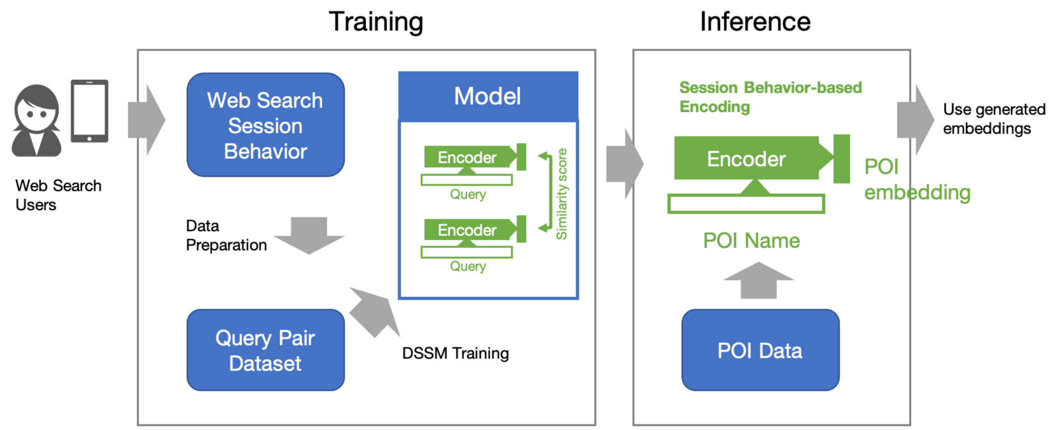


Figure 2: Overview of POI atmosphere categorization system.

Table 1: Examples of search sessions, where each block separated by double lines represents a session, and each row represents a query.

Session	Original Query	English Translation
1	代々木公園	Yoyogi Park
1	代々木公園　ランチ	Yoyogi Park, lunch
1	代々木公園　ランチ　子連れ	Yoyogi Park, lunch, with children
2	城山公園	Shiroyama Park
2	城山公園　子供	Shiroyama Park, children
2	メルヘン館　鹿児島	Fairy Tale Museum, Kagoshima (prefecture)
3	城南島海浜公園	Jonanjima Kaihin Park
3	城南島海浜公園　釣り	Jonanjima Kaihin Park, fishing
4	くじら運動公園	Kujira Undou Park
4	くじら運動公園　川遊び	Kujira Undou Park, swimming in river
4	くじら運動公園　バーベキュー	Kujira Undou Park, barbecue
5	光が丘公園	Hikarigaoka Park
5	東京　紅葉スポット	Tokyo, autumn leaves spots
6	豊洲ぐるり公園	Toyosu Gururi Park
6	豊洲ぐるり公園　夜景	Toyosu Gururi Park, night view

where $t(Q)$ represents the posting time of a query Q, setting $t(Q_0) = -\infty$ and $t(Q_{|H|+1}) = \infty$.

We manually set the interval δ to two minutes through a preliminary experiment.[2] We discarded search sessions with only one query since we need the next query for training. For each search session, we kept up to ten queries from the beginning and discarded the rest to avoid unfavorable context drift and computational cost. Finally, we extracted all pairs of adjacent queries from the filtered search sessions. We did not use non-adjacent pairs since the number of such pairs is quadratic to the session length, and adjacent pairs can implicitly consider non-adjacent pairs. More specifically, given a search session (Q_1, Q_2, Q_3), if the prediction of Q_3 by Q_2 is correct, it is also informative for the prediction of Q_2 by Q_1. We divided them into three sets, training, validation, and test sets, as shown in Table 2 and collectively constructed a dataset, called the *query pair dataset*. The average length of queries in the dataset was 9.14 Japanese characters, the average number of terms (i.e., chunks of characters delimited by white spaces) in each query was 1.74, and the number of unique characters was 26,383.

[2]We manually checked whether each query pair is in the same context and confirmed that the accuracy was about 95% for 100 random pairs.

Table 2: Details of query pair dataset.

	Training	Validation	Test
# of query pairs	299,923,615	16,629	33,375

2.3 DSSM Training

Figure 3 shows the overview of DSSM training. It exemplifies a training procedure for predicting the true next query "Cherry blossoms" given a query "Sumida Park", assuming that we have a search session (..., "Sumida Park", "Cherry blossoms", ...). Let Q and Q'_0 be the input query and the true next query, which correspond to a pair (i.e., (Q, Q'_0)) in the query pair dataset. Our goal is to obtain a query encoder $f : \mathbf{Q} \rightarrow \mathcal{R}^n$ that maps a query $Q \in \mathbf{Q}$ to an embedding or n-dimensional continuous vector on $\mathcal{R}^n$. We use a

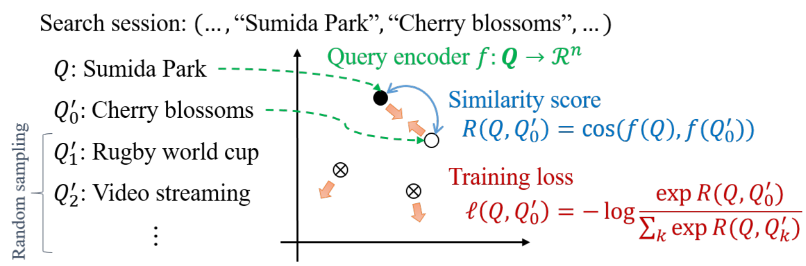


Figure 3: Overview of DSSM training for next query prediction. The left list shows an input query Q and the next query candidates $\{Q'_k\}$, where Q'_0 is a positive example (true next query), and the others Q'_k $(k > 0)$ are randomly sampled pseudo negative examples. The right plot shows the embedding space created by the query encoder f. The training is conducted with a categorical cross entropy loss $\ell(Q, Q'_0)$ based on similarity scores $R(Q, Q'_k)$ from the input query Q. Overall, this training can adjust the query encoder f so that the input query Q is close to the next query Q'_0 and far from random queries Q'_k $(k > 0)$.

recurrent neural network-based encoder as the query encoder, so f returns the linear transformation of the last hidden state calculated in a recurrent manner. Since this procedure is not essential in our study, we will omit the details.

The next query prediction task is a kind of multi-class classification with randomly sampled pseudo negative examples. Let $\{Q'_k\}_{1\le k\le K}$ be the pseudo negative examples, where K is the number of negative examples. Formally, the task is to correctly select the true next query Q'_0 from candidate queries $\{Q'_k\}_{0\le k\le K}$ (including Q'_0) given a query Q. The difference from the ordinary multi-class classification with predefined class labels is that the number of class labels is extremely large, since we need to consider all available queries (basically, more than tens of millions). In the DSSM framework, we can solve this problem by encoding class labels (candidate queries) as well as the input query, in addition to sampling negative examples as mentioned above. More specifically, we define a similarity score $R(Q, Q')$ between two queries, Q and Q', on the embedding space and calculate the probability $p(Q'_0|Q)$ of the true next query based on the scores of the candidate queries for the input query.

In this paper, we define $p(Q'_0|Q)$ as a softmax probability and $R(Q, Q')$ as a cosine similarity, as follows.

$$p(Q'_0|Q) = \frac{\exp(\beta R(Q, Q'_0))}{\sum_{k=0}^{K} \exp(\beta R(Q, Q'_k))}, \tag{4}$$

$$R(Q, Q') = \frac{f(Q)^\top f(Q')}{||f(Q)||\,||f(Q')||}, \tag{5}$$

where β is a hyperparamter (i.e., the inverse temperature coefficient) to shape the probability distribution. Note that Figure 3 represents a simple case of $\beta = 1$. Using the true label probability $p(Q'_0|Q)$, we can train the query encoder in a similar fashion as ordinary multi-class classification. Let $\ell(Q, Q'_0)$ be the categorical cross entropy loss, as follows.

$$\ell(Q, Q'_0) = -\log p(Q'_0|Q). \tag{6}$$

Note that Q'_0 is always the true label. Finally, we have the following total loss for the query pair data including i-th pair $(Q^{(i)}, Q_0'^{(i)})$:

$$L_\Theta = \sum_i \ell(Q^{(i)}, Q_0'^{(i)}), \tag{7}$$

where Θ represents the learning parameters in the query encoder. We can minimize L_Θ by using a (stochastic) gradient descent optimizer and obtain a query encoder f that maps a query Q to an embedding so that Q is close to the next query Q'_0 and far from the other random queries Q'_k $(k > 0)$ on the embedding space, as shown in Figure 3.

Implementation Note. We used a language modeling loss along with the above-mentioned DSSM loss to stabilize the training procedure in the experiments. Note that we chose character-based language modeling (next character prediction) rather than word-based one, following the report [17] that character-based one is a better choice for Japanese texts, especially when processing rare words. Let Q be a sequence of Japanese characters, i.e., $Q = (c_1, \ldots, c_{|Q|})$. The language modeling loss is defined as the standard negative log likelihood, as follows.

$$L^{\mathrm{LM}}_{\Theta^{\mathrm{LM}}} = \sum_i \ell^{\mathrm{LM}}(Q^{(i)}), \tag{8}$$

$$\ell^{\mathrm{LM}}(Q) = -\sum_{t=1}^{|Q|} \log p(c_t|\, c_1, \ldots, c_{t-1}). \tag{9}$$

The whole training proceeds by alternately optimizing the DSSM loss and the language modeling loss.

3 EXPERIMENTS

We experimented with POI categorization of parks as a case study in order to confirm the effectiveness of our method, especially for POI atmosphere. Here, we explain the experimental settings in Section 3.1 and demonstrate the characteristics of the query encoder itself for general queries in Section 3.2. We conducted three experiments focusing on parks in the Tokyo metropolitan area, since we can use the detailed categories that were manually created by the government as (pseudo) correct answers. In Section 3.3, we describe the experiments on categorization of parks and other POIs to confirm the performance in the case of general POIs. In Section 3.4, we describe the experiments on atmosphere categorization of parks with the detailed categories to show the ability of our method to distinguish POI atmospheres. In Section 3.5, we observed park atmosphere in more detail and discovered several connections between the obtained POI atmosphere and the real world.

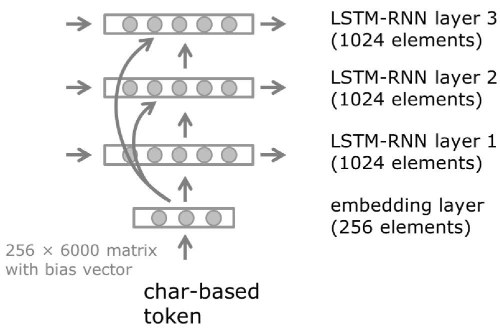


Figure 4: Model settings of query encoder based on LSTMs.

3.1 Experimental Settings

3.1.1 Model Structure. The model structure of our query encoder was a three-layer LSTM stacked on an embedding layer, as shown in Figure 4. We used the LSTM formulation by Graves [3]. The embedding layer receives an input token representing a character in the vocabulary of 6000 characters and gives a corresponding 256-dimensional vector, and this vector is fed to each recurrent layer. The sizes of the recurrent layers are all set to 1024, and the hidden and cell state vectors of an LSTM layer are always initialized with zeros. The model has an additional fully connected layer and produces the 128-dimensional vector for the next query prediction. The total number of parameters was 29M.

3.1.2 Training. We used the Adam [7] optimizer with hyperparameters ($\alpha = 0.0001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$), along with the gradient clipping with threshold 0.1, for training the query encoder. The inverse temperature coefficient β was set to 10, and the number K of negative examples was 4, meaning the task was to select the true next query from 5 candidates. The batch size for next-query prediction was 96, and that for language modeling was 512. Note that we used 480M single queries in the query pair dataset for language modeling. The maximum number of iterations was 5M, where each iteration consumed two batches for both tasks. The training took about 50 days with one GPU (Tesla V100). All the hyperparameters were determined through several preliminary experiments.

3.1.3 Baseline: Skip-gram Model. We chose the skip-gram model (also known as the word2vec's model [10]) as a baseline, as it has been often used in POI embedding methods such as POI2vec. This model has a log-bilinear structure and is trained with word sequences assuming the distributional hypothesis in linguistics that the meaning of a word can be represented by its surrounding words. After the training, we obtained a fixed mapping from the words in the pre-defined dictionary to embeddings.

We used the implementation in the gensim tool [15] and trained the skip-gram model with all the search sessions containing 480M queries, as word sequences. The hyperparameters were as follows: the embedding size was 128, the window size was 5, the minimum count was 1, the number of training epochs was 10, and the others were left at their defaults. There were no unknown POI names in our experiments; that is, all the POI names were included in the search sessions. Note that we also tried pretrained embeddings trained with Wikipedia data,[3] but we did not use them since their predifined dictionary had only 6.39% of the parks used in Section 3.4.

Table 3: Detailed information about nine parks, where each park has its type, area (m^2), and the principle component scores (x, y) in Figure 7a.

Id	Park Name	Type	Area	x	y
α	Komaba	Nei.	40,396	0.089	0.271
β	Jingudoori	Nei.	3,128	-0.473	0.057
γ	Minami Ikebukuro	Urb.	7,818	-0.047	-0.109
δ	Futako Tamagawa	Dis.	63,000	0.420	-0.191
ϵ	Denen Chofu Seseragi	Spo.	30,300	0.218	0.102
ζ	Setagaya	Dis.	78,957	0.158	0.002
η	Senzoku Ike	Gen.	40,000	0.076	0.386
θ	Utsukushi Ga Oka	Nei.	21,832	0.263	0.351
ι	Aobadai	Nei.	38,000	0.173	0.240

3.2 Characteristics of Query Encoder

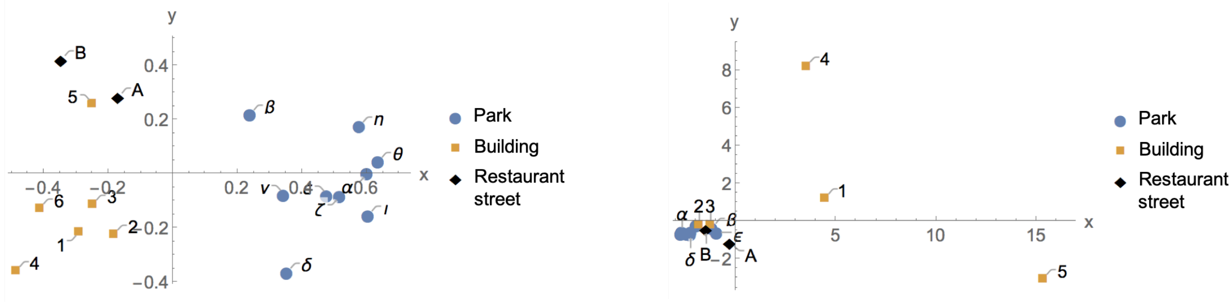
We calculated cosine similarities for several POI-related queries to demonstrate characteristics of the query encoder. In summary, the similarity was reasonably high when two POIs belonged to the same category and low otherwise. For instance, the similarity between two urban areas “Shinjuku” and “Yokohama” was high (score: 0.70), while that for “Shinjuku” and a relatively rural area “Hachinohe” was relatively low (score: 0.52). “Hachinohe” was dissimilar to the spa resort “Atami” (score: 0.38), but “Atami” was very similar to another spa resort “Yufuin” (score: 0.77). Note that here we use English translations of Japanese queries and POI names only to explain the experiments.

One might think that character-based encoding will make mistakes when two words look similar but have completely different meanings. One such case is a mountain name “富士山(Mt. Fuji)” and a company name “富士通(Fujitsu),” which share two Japanese characters representing “Fuji,” making them look similar. However, their similarity was significantly low (score: 0.25), which implies that the next queries of those two words are naturally different. Conversely, there is also a case where two words look different but have almost the same meaning. For example, the similarity of a park “美しが丘公園(Utsukushi Ga Oka Park)” and possibly the same but misspelled one “美しくしがおかこうえん(Utsukushi Kushi Ga Oka Park)” was interestingly high (score: 0.79). This demonstrates a positive effect of our character-based encoding in that our encoder can flexibly handle users' misspellings in real applications.

3.3 Categorization of Parks and other POIs

We experimented on categorization of parks, buildings, and streets with many restaurants (restaurant streets) in the Tokyo metropolitan area to confirm the performance of our encoder for general POIs. We prepared the nine parks shown in Table 3 ($\alpha, \ldots, \iota$), six buildings (1: Shibuya Cast, 2: Roppongi Hills, 3: Tokyo Midtown, 4: Ginza Six, 5: Shibuya Hikarie, and 6: Hibiya Midtown), and two restaurant streets (A: Golden Street, B: Corridor Street). We obtained the embeddings of all the POIs by using our method and the

[3] http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/



(a) Proposed method. **(b) Skip-gram.**

Figure 5: Two-dimensional PCA-based visualizations for embeddings of parks, buildings, and restaurant streets created by our method (a) and the baseline (b).

baseline and visualized them with principle component analysis (PCA).

Figure 5 shows the two-dimensional PCA-based visualizations; figures (a) and (b) are those for our method and the baselines, respectively. According to the results, our method convincingly grouped the POIs into three categories (i.e., circles of "Park," squares of "Building," and diamonds of "Restaurant street"), whereas the baseline could not form recognizable groups. Although the building of "5: Shibuya Hikarie" is unexpectedly closer to the " Restaurant street" group than the "Building" group, this result is actually intuitive because the Hikarie building has a very large restaurant area inside, and people frequently use it for lunch and dinner. Of course, the other five buildings also have restaurants, but these buildings have many company offices and are mainly utilized for business purposes.

3.4 Atmosphere Categorization of Parks

We experimented on atmosphere categorization focusing on parks to show the ability of our method to distinguish POI atmosphere. We prepared 532 parks, which were the ones remaining after filtering out very small parks (such as block parks) from all the 3031 parks in the Tokyo metropolitan area. As (pseudo) correct answers for POI atmosphere, we chose the classification scheme of parks established by the government, where each park is given one of the detailed types depending on its location, area, and purpose. Since they are manually annotated considering the purpose of use, they were expected to reflect POI atmosphere information such as "why users visit there." The major types (major categories of the detailed types) are as follows.

- Type 1: Parks that are usually in the suburbs and utilized for sports activities and recreation.
- Type 2: Green belts in urban spaces for improving the cityscape.
- Type 3: Greenways and forests for improving the safety and comfort of city life.

We conducted qualitative and quantitative comparisons between our method and the baseline in terms of these major types.

3.4.1 Qualitative Comparison. Figure 6 shows two-dimensional visualizations of the embeddings produced by our method (a) and the baseline (b). Here, we used t-SNE (t-distributed stochastic neighbor embedding) [19] for the visualization since it was difficult to extract principle components by PCA in this complicated case. t-SNE is a dimensionality reduction method via a non-linear projection based on the Student t-distribution and commonly used for visualizing word embeddings since it has the ability to preserve local structure; roughly speaking, points that are close to each other in the original space are also close in a low-dimensional space.[4]

Looking at figure (a), we can see that our method clearly separated Type 1 (circles) and Type 3 (diamonds). This result is intuitive since Types 1 and 3 correspond to the artificial and natural atmospheres, respectively. As for Type 2 (squares), the corresponding parks seem scattered at first glance, but they are actually closer to or surrounding Type 3 and even separated from Type 1, with some exceptions. Note that only local similarities are meaningful in t-SNE, so the result does not mean that the concept of Type 2 surrounds or includes that of Type 3. This result also reflects Type 2's modest natural atmosphere well. On the other hand, figure (b) shows that the baseline could not obtain any meaningful result. Overall, we confirmed that our method outperformed the baseline in terms of POI atmosphere categorization.

3.4.2 Quantitative Comparison. We conducted quantitative comparison of our method and the baseline in terms of separability. We set the separability measure as the average accuracy of 10 runs of a simple classifier since we have the correct answers in this task, although there are various unsupervised separability definitions. We prepared n-layer classifiers with n fully connected layers for this evaluation. Given the 128-dimensional embedding of each park, the n-layer classifier returns the softmax probability of each type for the park after processing with the n layers. We trained the model with the standard cross entropy loss with the 532 parks with their types, where the optimizer was stochastic gradient descent with the learning rate $\alpha = 1$.

Table 4 shows the separability results of our method and the baseline via one-layer and two-layer classifiers. The one-layer classifier has only one linear transformation and heavily depends on the expressiveness of the embeddings, while the two-layer classifier is relatively robust to the input. The results show that the separability of our method was clearly higher than that of the baseline, which quantitatively shows the superiority of our method. Since

[4]Note that the axes x and y shown here are automatically created by t-SNE, and it is difficult to determine what they mean.

Table 4: Separability and its standard deviation of our method and the baselines via one-layer and two-layer classifiers.

	One-layer	Two-layer
Proposed method	0.929 ± 0.002	0.962 ± 0.003
Skip-gram	0.822 ± 0.000	0.861 ± 0.008

there were no big differences between the results of the one-layer and two-layer classifiers, we conjecture that similar results will be obtained with other classifiers.

3.5 Connecting Park Atmosphere to the Real World

Finally, we took a closer look at the nine parks in Section 3.3 to find connections between the obtained POI embeddings (or atmosphere) and the real world. Table 3 includes detailed information on the nine parks, where each park has its detailed type and area. We created the following three categories by grouping similar types to simplify this analysis.

- District: District parks (Dis.) are designed to improve the living environment of local residents.
- Sports and general: Sports parks (Spo.) and general parks (Gen.) provide a variety of recreational facilities that supplement the park's main purpose.
- Neighboring and urban: Neighborhood parks (Nei.) and urban parks (Urb.) mean nearly the same thing; they are located in a residential or urban area and are intended to be used by neighboring residents. The area of a park of this type is usually small.

Figure 7 shows the two-dimensional PCA-based visualizations produced by our method and the baseline, where the size and color of each point means the area and category of the corresponding park, respectively. Note that we chose PCA in this analysis because the number of parks was relatively small. We also put the principle component scores (x, y) in Table 3. Looking at the results of our method (a), we can guess what each axis represents. The first principal component (x-axis) separates very small "Neighboring and urban" parks from the rest, and the second principal component (y-axis) seems to divide the rest into two groups; the upper one is for medium-sized parks that are located near large-scale apartment complexes, and the lower one is for large "District" parks to which people come possibly from distant places by car or train. Although area is not included in the park atmosphere, it is important that our encoder expressed area-related features in the embeddings without any prior knowledge of their areas, despite that it was trained only with search queries.

Figure 8 shows the satellite images of the nine parks as supplementary information. As the reader can see, each park has different geographical features, which are much richer information than the area feature in the previous paragraph. We qualitatively confirmed whether our method reflected such features and deeper atmospheres by examining the official information on each park and associating its image and PCA result. The three observations below are proof that our method succeeded in capturing the geographical features and atmosphere.

- The "β: Jingudori" park is located in a crowded business district, and office workers often take breaks on its benches. The public toilet in the small park is often used by neighbors and passers-by. On the other hand, the other eight parks have open spaces where people can play sports, so it is intuitive that the Jingudori park was separated from the other parks.
- Only in the "η: Senzoku Ike" and "θ: Utsukushi Ga Oka" parks can we enjoy fishing. Although these parks are defined as different categories, they are utilized as the same purpose. Indeed, η and θ are located near each other in Figure 7a.
- The "δ: Futako Tamagawa" park is separated from the other parks and located opposite to "β: Jingudori" park in Figure 7a. Most users of the Futako Tamagawa park are families with children, since there are many playgrounds for children in the park. As mentioned above, because the Jingudori park is for business persons, these two parks are very different from each other, which corresponds to the PCA results.
- The "γ: Minami-Ikebukuro" park is located in a similar business district and has a similar small area to the "β: Jingudori" park, but the way users spend their time in each park is completely different. This fact is well reflected in the visualization, which means that even similar parks in geographical features can be successfully distinguished by their "atmospheres."

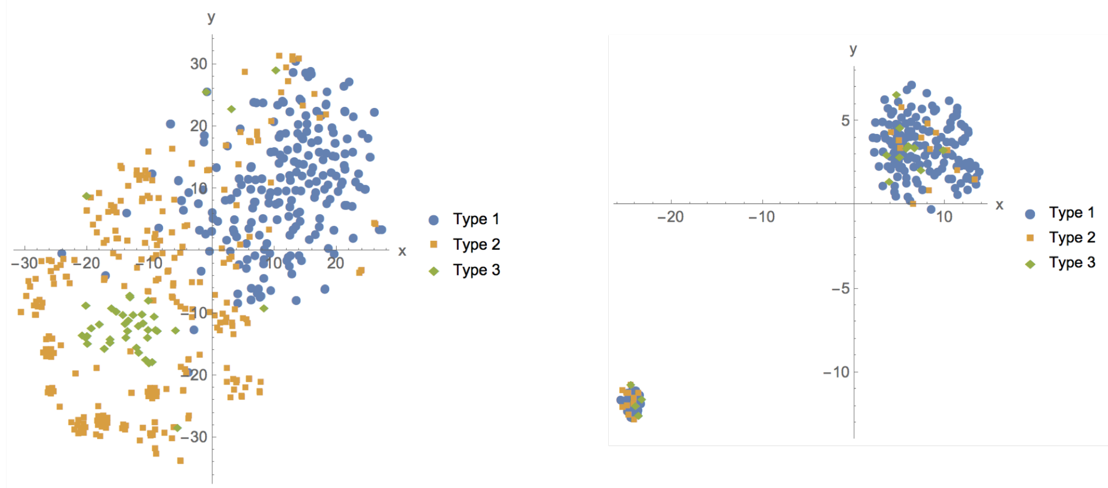
It should be emphasized that our method leads to these observations without using any geographical features, although some of them could have been identified by analyzing the satellite images.

4 RELATED WORK

We clarify the novelty of our work by explaining several related studies in two fields: POI embedding and query classification. In summary, our work is essentially different from them in that we create POI embeddings from search queries and show their effectiveness via a case study for industrial geospatial applications, whereas previous studies on POI embedding and query classification were basically conducted sorely using geospatial resources and search queries, respectively.

4.1 POI Embedding

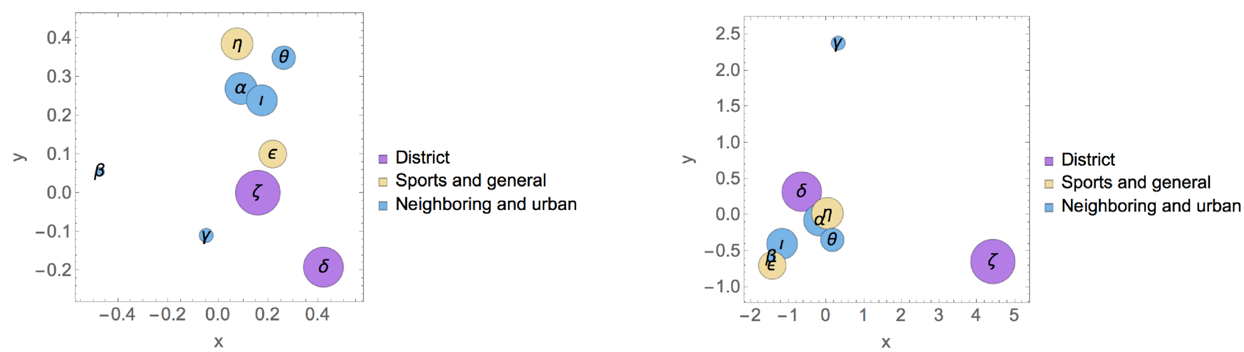
There have been many studies on POI embedding, which are roughly divided into two groups; one group uses user behavior in a geographical space, and the other uses geospatial features of a POI. In the first group with user behavior, typical studies used check-in sequences on location-based social networks such as Foursquare to train word2vec-like models. For example, Zhao et al. [25] proposed Geo-Teaser, a variant of the skip-gram model with temporal states (i.e., weekday and weekend), and Feng et al. [2] proposed POI2Vec, a variant of continuous bag-of-words (CBOW) model [10] that incorporates the geographical influence of POIs, where people tend to visit nearby locations. As a graph-based approach, Wang et al. [20] proposed a variant of the DeepWalk model [12] trained with shortest paths instead of randomly sampled paths to practically learn embeddings of intersections on road networks. In the second group with geospatial features, Jean et al. [6] and Spruyt [18] proposed Tile2Vec and Loc2Vec, respectively, that map satellite images to embeddings. Each method consisted of a convolutional neural



(a) Proposed method.

(b) Skip-gram.

Figure 6: Two-dimensional t-SNE-based visualizations for embeddings of parks with three major types by our method (a) and the baseline (b).



(a) Proposed method.

(b) Skip-gram.

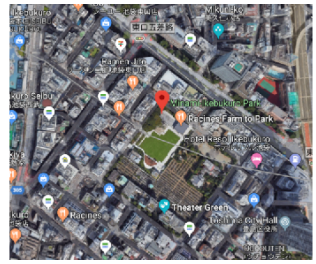
Figure 7: Two-dimensional PCA visualizations for embeddings of nine parks produced by our method (a) and the baseline (b). The size and color of each point represent the area and category of the corresponding park, respectively.

network-based encoder trained with a triplet loss, which minimizes the distance between the embeddings of the anchor location and the neighbor location while maximizing the distance between the anchor and distant locations. Furthermore, there have been several studies using resources with geotags, such as Place2Vec [23] trained with geotagged Yelp venues and GPS2Vec [24] with geotagged Flickr images, where Place2Vec used a distance-binned skip-gram model, and GPS2Vec used a neural encoder that predicts softmax probabilities of semantic features. All of these studies were confirmed to be useful for their target tasks such as ordinary POI categorization, but they will not be enough for our task since their resources do not directly contain information indicating POI atmospheres such as “why users visit there,” as explained in Section 1.

Several studies improved the quality of POI embeddings by using textual information. For example, Chang et al. [1] proposed a hierarchical skip-gram model that can simultaneously consider text content and check-in sequences on Instagram. Wei et al. [22] proposed a graph-based method LeGo-CM that embeds texts, geotags, and time stamps on Twitter into the same latent space. Their findings are useful in that our embeddings will also improve other geospatial models, but our purpose is essentially different from theirs. As mentioned in Section 1, there have been few studies purely based on linguistic resources. Quercini and Samet [13] used the Wikipedia link structure to the relatedness of a concept to a location, but they did not address learning embeddings. A study of Konkol et al. [8] is the most related one, where they examined if embeddings of city names are related to their locations in the real world. However, since they used the skip-gram model with Wikipedia for general POIs, their settings are completely different from ours (DSSM with search queries for POI atmosphere). In addition, we confirmed that our DSSM model performed better than the skip-gram model in Section 3.4.

α: Komaba

β: Jingudori



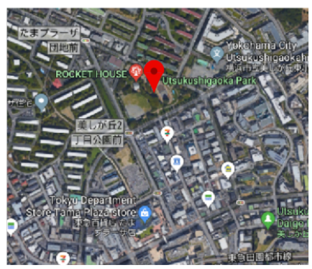
γ: Minami Ikebukuro

δ: Futako Tamagawa

ϵ: Denen Chofu Seseragi

ζ: Setagaya

η: Senzoku Ike



θ: Utsukushi Ga Oka

ι: Aobadai

Figure 8: Satellite photos of the nine parks (image credit: ©Google Earth 2019).

4.2 Query Classification

Our work can be regarded as query classification in the natural language processing or information retrieval fields, when considering a specific task that classifies only POI queries into predefined categories, as in Section 3.4. Note that we used the term "categorization" in this paper since our purpose is to group similar POIs either explicitly or implicitly, rather than just classify them. There have been many studies on query classification [4, 5, 9, 14, 16, 21], since understanding the intents of users' queries is essential to improving the quality of a search engine from a practical standpoint. However, there are no studies connecting embeddings based on search query logs to geospatial features in the real world, like ours.

We explain previous studies on query classification in more detail, which are roughly divided into three groups [26] with respect to training features: query expression, retrieved content, and user behavior. The first group based on query expression uses common text features, such as the types, frequencies, and lengths of words and characters in a query [5, 16]. An advantage of this information is that we can develop a classification system at a low cost by only considering query texts. The second group based on retrieved content involves anchor links, snippets, and display ads as well as documents retrieved by a query [4, 9]. The third group based on user behavior is often an enhanced version of the second group, which includes user feedback for retrieved content, such as the click-through rate, dwell time, and past action history [14, 21]. These rich features of the second and third groups can improve classification performance, but it will be costly to conduct experiments and develop systems. Our approach with search session behavior belongs to both first and third groups, but we only use the query encoder in inference time, whose input is a single query. This means that the deployment cost is the same as that of the first group, so our approach is promising for industrial applications.

5 CONCLUSION

We proposed a new POI categorization method that can represent the atmosphere of a POI beyond its geospatial features. Our method involves a neural query encoder that maps POIs to embeddings, which was trained via next-query prediction within the DSSM framework. Moreover, considering users' search behavior helps us to extract the context of users; thus, we could successfully represent POI atmospheres without any prior knowledge. We addressed POI categorization of parks in a case study and demonstrated the effectiveness of our method, especially for POI atmosphere in comparison with the widely used skip-gram models. Furthermore, we discovered several connections between the obtained embeddings (atmosphere) and the real world by referring to the official information and satellite images of each park. These results indicate that search queries are highly valuable for geographic information processing, as well as natural language processing. We believe that our method complements the existing methods based on geographic features, such as check-in sequences and satellite images, and will encourage their use in the real world. Our future work will include checking if our method can distinguish atmospheres for POIs other than parks. Our method is general enough to cover any POIs used as queries in a search engine, so we expect that our method will improve with various findings from other geographical resources.

ACKNOWLEDGMENTS

We thank Nobuyuki Shimizu and Junji Saikawa for thoughtful discussions and the anonymous reviewers for their valuable feedback.

REFERENCES

[1] Buru Chang, Yonggyu Park, Donghyeon Park, Seongsoon Kim, and Jaewoo Kang. 2018. Content-Aware Hierarchical Point-of-Interest Embedding Model for Successive POI Recommendation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*. IJCAI Organization, 3301–3307. https://doi.org/10.24963/ijcai.2018/458

[2] Shanshan Feng, Gao Cong, Bo An, and Yeow Meng Chee. 2017. POI2Vec: Geographical Latent Representation for Predicting Future Visitors. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI 2017)*. AAAI Press, 102–108. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14902

[3] Alex Graves. 2013. Generating Sequences With Recurrent Neural Networks. *CoRR* abs/1308.0850 (2013). http://arxiv.org/abs/1308.0850

[4] Mohamed Farouk Abdel Hady, Rania Ibrahim, and Ahmed Ashour. 2013. Domain adaptation for cross-lingual query classification using search query logs and document classification. In *Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN 2013)*. IEEE, 1–8. https://doi.org/10.1109/IJCNN.2013.6707061

[5] Homa B. Hashemi, Amir Asiaee, and Reiner Kraft. 2016. Query intent detection using convolutional neural networks. In *International Conference on Web Search and Data Mining, Workshop on Query Understanding*. http://people.cs.pitt.edu/~hashemi/papers/QRUMS2016_HBHashemi.pdf

[6] Neal Jean, Sherrie Wang, Anshul Samar, George Azzari, David B. Lobell, and Stefano Ermon. 2019. Tile2Vec: Unsupervised Representation Learning for Spatially Distributed Data. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*. AAAI Press, 3967–3974. https://doi.org/10.1609/aaai.v33i01.33013967

[7] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). http://arxiv.org/abs/1412.6980

[8] Michal Konkol, Tomáš Brychcín, Michal Nykl, and Tomáš Hercig. 2017. Geographical Evaluation of Word Embeddings. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP 2017)*. AFNLP, 224–232. https://www.aclweb.org/anthology/I17-1023

[9] Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation Learning Using Multi-Task Deep Neural Networks for Semantic Classification and Information Retrieval. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2015)*. ACL, 912–921. http://www.aclweb.org/anthology/N15-1092

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the 1st International Conference on Learning Representations (ICLR 2013)*. http://arxiv.org/abs/1301.3781

[11] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab K. Ward. 2014. Semantic Modelling with Long-Short-Term Memory for Information Retrieval. *CoRR* abs/1412.6629 (2014). http://arxiv.org/abs/1412.6629

[12] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2014)*. ACM, 701–710. https://doi.org/10.1145/2623330.2623732

[13] Gianluca Quercini and Hanan Samet. 2014. Uncovering the Spatial Relatedness in Wikipedia. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL 2014)*. ACM, 153–162. https://doi.org/10.1145/2666310.2666398

[14] Filip Radlinski, Martin Szummer, and Nick Craswell. 2010. Inferring Query Intent from Reformulations and Clicks. In *Proceedings of the 19th International Conference on World Wide Web (WWW 2010)*. ACM, 1171–1172. https://doi.org/10.1145/1772690.1772859

[15] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, 45–50. http://is.muni.cz/publication/884893/en.

[16] Yangyang Shi, Kaisheng Yao, Le Tian, and Daxin Jiang. 2016. Deep LSTM based Feature Mapping for Query Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*. ACL, 1501–1511. http://www.aclweb.org/anthology/N16-1176

[17] Toru Shimizu, Hayato Kobayashi, and Nobuyuki Shimizu. 2018. Pretraining Sentiment Classifiers with Unlabeled Dialog Data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*. ACL, 764–770. https://www.aclweb.org/anthology/P18-2121

[18] Vincent Spruyt. 2018. Loc2Vec: Learning location embeddings with triplet-loss networks. (2018). https://www.sentiance.com/2018/05/03/venue-mapping/

[19] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research* 9(Nov) (2008), 2579–2605. http://www.jmlr.org/papers/v9/vandermaaten08a.html

[20] Meng-xiang Wang, Wang-Chien Lee, Tao-yang Fu, and Ge Yu. 2019. Learning Embeddings of Intersections on Road Networks. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL 2019)*. ACM, 309–318. https://doi.org/10.1145/3347146.3359075

[21] Yu Wang and Eugene Agichtein. 2010. Query Ambiguity Revisited: Clickthrough Measures for Distinguishing Informational and Ambiguous Queries. In *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2010)*. ACL, 361–364. http://www.aclweb.org/anthology/N10-1055

[22] Hong Wei, Janit Anjaria, and Hanan Samet. 2019. Learning Embeddings of Spatial, Textual and Temporal Entities in Geotagged Tweets. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL 2019)*. ACM, 484–487. https://doi.org/10.1145/3347146.3359108

[23] Bo Yan, Krzysztof Janowicz, Gengchen Mai, and Song Gao. 2017. From ITDL to Place2Vec: Reasoning About Place Type Similarity and Relatedness by Learning Embeddings From Augmented Spatial Contexts. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL 2017)*. ACM, 35:1–35:10. https://doi.org/10.1145/3139958.3140054

[24] Yifang Yin, Zhenguang Liu, Ying Zhang, Sheng Wang, Rajiv Ratn Shah, and Roger Zimmermann. 2019. GPS2Vec: Towards Generating Worldwide GPS Embeddings. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL 2019)*. ACM, 416–419. https://doi.org/10.1145/3347146.3359067

[25] Shenglin Zhao, Tong Zhao, Irwin King, and Michael R. Lyu. 2017. Geo-Teaser: Geo-Temporal Sequential Embedding Rank for Point-of-interest Recommendation. In *Proceedings of the 26th International Conference on World Wide Web Companion (WWW 2017 Companion)*. IW3C2, 153–162. https://doi.org/10.1145/3041021.3054138

[26] Sanduo Zhou, Kefei Cheng, and Lijun Men. 2017. The survey of large-scale query classification. *AIP Conference Proceedings* 1834, 1 (2017), 040045. https://doi.org/10.1063/1.4981641