

# Content-Aware Hierarchical Point-of-Interest Embedding Model for Successive POI Recommendation

Buru Chang<sup>1</sup>, Yonggyu Park<sup>1</sup>, Donghyeon Park<sup>1</sup>, Seongsoon Kim<sup>2</sup> and Jaewoo Kang<sup>1</sup>

<sup>1</sup> Korea University

<sup>2</sup> Naver Corporation

buru\_chang@korea.ac.kr, yonggyu@korea.ac.kr, parkdh@korea.ac.kr,  
seongsoon.kim@navercorp.com, kangj@korea.ac.kr

## Abstract

Recommending a point-of-interest (POI) a user will visit next based on temporal and spatial context information is an important task in mobile-based applications. Recently, several POI recommendation models based on conventional sequential-data modeling approaches have been proposed. However, such models focus on only a user’s check-in sequence information and the physical distance between POIs. Furthermore, they do not utilize the characteristics of POIs or the relationships between POIs. To address this problem, we propose CAPE, the first content-aware POI embedding model which utilizes text content that provides information about the characteristics of a POI. CAPE consists of a check-in context layer and a text content layer. The check-in context layer captures the geographical influence of POIs from the check-in sequence of a user, while the text content layer captures the characteristics of POIs from the text content. To validate the efficacy of CAPE, we constructed a large-scale POI dataset. In the experimental evaluation, we show that the performance of the existing POI recommendation models can be significantly improved by simply applying CAPE to the models.

## 1 Introduction

In recent years, many mobile-based social media platforms such as Twitter, Facebook, Instagram, and Foursquare have grown in popularity.

Users on these platforms generate a large amount of data which includes text content with temporal and spatial information. Such information is particularly useful for understanding user behavior and preferences for a point-of-interest (POI) which is a specific location that someone finds interesting. In mobile-based applications, it is necessary but challenging to recommend where a user will visit next based on temporal and spatial information.

Recently, several models based on conventional sequential-data modeling approaches and designed to recommend the

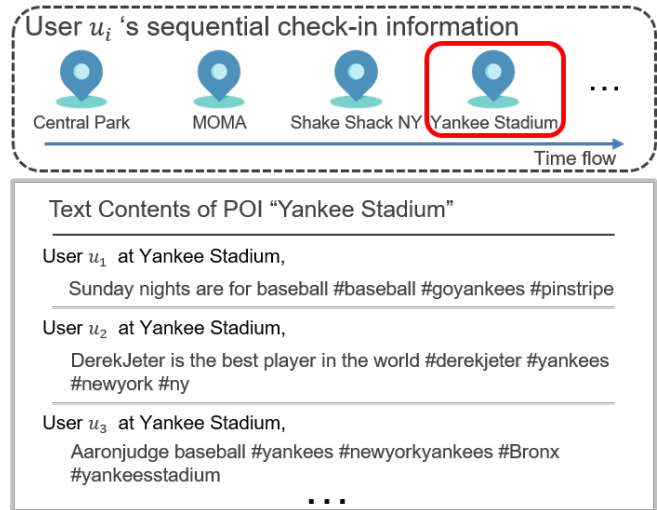


Figure 1: Example data from mobile-based social media. The user check-in information is represented as a sequence of POIs. Each POI contains text contents.

next POI have been proposed. Markov Chain based models [Cheng *et al.*, 2013; Chen *et al.*, 2014] exploit a user’s check-in sequence information while considering temporal and spatial constraints. Based on tensor factorization, STELLAR [Zhao *et al.*, 2016] models POI pairs visited successively. Also, ST-RNN [Liu *et al.*, 2016a] employs a recurrent neural network (RNN) and achieves state-of-the-art performance in modeling sequential data.

Although ST-RNN achieves state-of-the-art performance, it still has room for improvement. Since ST-RNN employs randomly initialized vectors as POI representations, it does not utilize the characteristics of POIs or the relationships between POIs. ST-RNN uses only the time interval between two check-ins and the distance between two POIs as factors to recommend the next POI.

Although POI embedding models [Liu *et al.*, 2016b; Feng *et al.*, 2017; Zhao *et al.*, 2017] have been recently proposed to capture relationships between POIs, they have not yet been employed in successive POI recommendation tasks. In addi-

tion, the existing POI embedding models use only user check-in context to capture the geographical influence of POIs, but they do not utilize text content which provides information about the characteristics of a POI. For example, if a POI’s text content includes the words “latte,” “coffee,” and “drinks,” the POI has characteristics associated with a cafe. Figure 1 shows the examples of text contents of the POI “Yankee Stadium” written by users.

In this paper, we first describe our newly constructed dataset that includes text contents collected from Instagram, and analyze the characteristics of the dataset used for our POI embedding study. Second, we propose **CAPE**, which is a **C**ontent-**A**ware hierarchical **P**OI **E**mbedding model for POI recommendation. From text contents, CAPE captures not only the geographical influence of POIs, but also the characteristics of POIs. CAPE consists of a check-in context layer and a text content layer. In the check-in context layer, embedding vectors of POIs in a user’s check-in sequence are trained to be close to each other in the vector space using Skip-gram [Mikolov *et al.*, 2013]. At the same time, in the text content layer, POI embedding vectors are trained to capture the characteristics of a POI from the words used in the POI’s text content. In the experimental section, we implement the successive POI recommendation models and apply our CAPE model to the POI recommendation models. The experimental evaluation on our dataset shows that CAPE contributes to improving the performance of successive POI recommendation models. The evaluation result also shows that CAPE captures the geographical influences of POIs and characteristics of POIs. CAPE outperforms previously proposed POI embedding models.

The contributions of this paper are summarized as follows.

- We construct a new large-scale POI dataset and analyze the dataset to investigate its applicability to POI recommendation studies. We show the importance of text content in POI embedding.
- We propose CAPE which is a new content-aware hierarchical POI embedding model that utilizes a user’s check-in sequence and text content about POIs. To the best of our knowledge, CAPE is the first content-aware POI embedding model.
- CAPE which captures the geographical influence and characteristics of a POI significantly improves the performance of successive POI recommendation models.
- We release our newly constructed large-scale dataset which includes text content written by mobile-based social network users. Our source code implemented in an open library is also publicly available.

The remainder of this paper is organized as follows. In Section 2, we summarize previously proposed successive POI recommendation models and POI embedding models. In Section 3, we introduce our newly constructed dataset which includes text content, and investigate the necessity of content-aware POI embedding. In Section 4, we describe our proposed model CAPE. In Section 5, we compare our proposed model with existing POI embedding models, and analyze the

experimental results. Last, in Section 6, we provide our concluding remarks of this paper.

## 2 Related Work

### 2.1 Successive POI Recommendation

There exist several successive POI recommendation models which are based on sequential-data modeling approaches. Zhao *et al.* [Zhao *et al.*, 2016] proposed a spatial-temporal latent ranking (STELLAR) method that employs a ranking-based pairwise tensor factorization framework to model consecutive POIs. To capture and exploit the temporal influence of check-in times on POI recommendation, STELLAR models not only POI-POI pairs but also time-POI pairs. However, STELLAR is unable to model the whole check-in sequence of a user because it focuses on pairs of consecutive POIs. Cheng *et al.* [Cheng *et al.*, 2013] and Chen *et al.* [Chen *et al.*, 2014] proposed recommendation models based on Factorizing Personalized Markov Chain (FPMC) [Rendle *et al.*, 2010] which extends the Markov chain model by factorizing the probability transition. Considering a user’s movement constraints, the model by Cheng *et al.* [Cheng *et al.*, 2013] embeds the personalized Markov chain. The model by Chen *et al.* [Chen *et al.*, 2014] combines the personalized Markov chain with the global Markov chain to capture global behavioral patterns. However, since the FPMC based models linearly combine all the embedded Markov chains and constraints, they require a strong independence assumption among different factors [Wang *et al.*, 2015]. With the success of RNN in sequential data modeling, RNN is also used to model a user’s check-in sequence for successive POI recommendation. Liu *et al.* [Liu *et al.*, 2016a] proposed ST-RNN which extends RNNs with time- and distance-specific transition matrices to model local temporal and spatial contexts, respectively.

Although the above models have achieved satisfactory results in some applications, the models are unable to utilize the characteristics of POIs and the relationships between POIs because they employ randomly initialized vectors as POI representations. In this paper, we focus on improving the performance of the existing POI recommendation models using our pre-trained POI embedding model CAPE, which captures the characteristics of POIs.

### 2.2 POI Embedding

As word embedding has proven to be successful in the NLP field, studies have proposed POI embedding models that capture the geographical influence of POIs from a user’s check-in sequence. Liu *et al.* [Liu *et al.*, 2016b] assume that a POI is influenced by other POIs in a user’s check-in sequence. They treated each POI as a “word” and each user’s sequential check-in information as a “sentence,” and used Skip-gram [Mikolov *et al.*, 2013] to train a POI’s latent representation vector. Furthermore, they proposed a time-aware POI recommendation model that uses a time latent representation vector. Feng *et al.* [Feng *et al.*, 2017] proposed POI2Vec, which is a POI latent representation model to incorporate geographical influence. They developed a hierarchical binary tree based on the physical distance between POIs to reflect the geographical

# of check-ins	2,216,631
# of POIs	13,187
# of users	78,233
# of words	958,386
Avg. # of check-ins per user	28.3
Avg. # of POIs per user	15.2
Avg. # of users per POI	90.0

Table 1: Statistics of our newly constructed dataset.

influence. Zhao *et al.* [Zhao *et al.*, 2017] considered the temporal characteristics of POIs. Because of the low correlation between POIs visited on different days, they treated a user’s check-ins of one day as a “sentence”. Then, they proposed a temporal POI embedding model that differentiates between weekdays and weekends using a specific temporal state.

There are few studies that use text content for POI embedding. However, our model utilizes a user’s check-in sequence information and text content for POI embedding.

### 3 Data Description and Analysis

In this section, we first describe our large-scale dataset which includes text content, and then analyze the properties of the dataset.

#### 3.1 Data Description

Since most of the existing POI recommendation studies do not use text content, there is no suitable dataset that contains text content. Although Yang *et al.* [Yang *et al.*, 2013] use text content to understand user sentiment and improve the POI recommendation performance, their dataset is unsuitable for learning POI representations. First, many POIs in their dataset do not have text contents written by users. Second, the dataset is not large enough for training POI representations. Therefore, we constructed a new dataset that contains text contents which refer to POIs. We collected data from Instagram, which is one of the most popular mobile-based social networks. Instagram data includes not only user POI check-in information, but also text content written by users. We collected Instagram data created in New York City and preprocessed the collected data utilizing the same method of Zhao *et al.* [Zhao *et al.*, 2017]. We removed the POIs with less than five checked-ins and the users who had less than ten posts. After preprocessing, our new dataset<sup>1</sup> includes 2,216,631 check-ins at 13,187 POIs of 78,233 users. The statistics of our dataset are summarized in Table 1.

#### 3.2 Empirical Analysis

Instagram generates a large amount of location-aware data each day. However, the Instagram dataset has been used in only one POI recommendation study [Wang *et al.*, 2017] that used images to understand user preference about POIs. Therefore, we conducted a data analysis to determine whether our newly constructed dataset is suitable for the successive

<sup>1</sup>Our newly constructed dataset is available at <https://dmis.korea.ac.kr/cape>

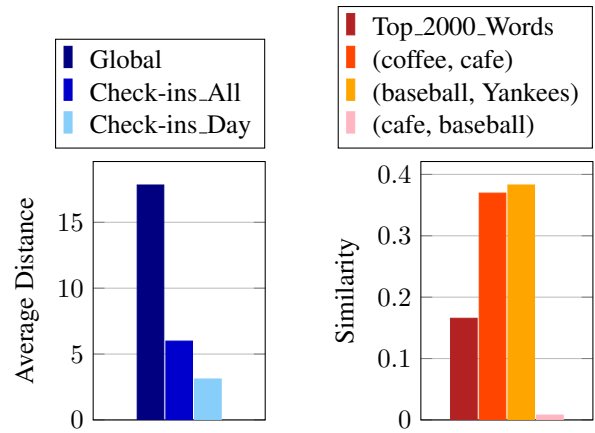


Figure 2: Average distance between POIs

Figure 3: Jaccard similarity between the POI sets

POI recommendation task. We also analyzed the relationships between words and POIs to capture the importance of text content in POI embedding.

#### Geographical influence

According to POI recommendation studies [Ye *et al.*, 2011; Liu *et al.*, 2013], location-based social network users tend to visit POIs located near the POIs that the users have already visited. We investigate whether the users in our dataset have a similar tendency. We measure the average distance between POIs in a user’s check-in information (*Check-ins*) and compare this distance with the average distance between all the POIs in the dataset (*Global*). We make two user check-in information sets using the methods of Liu *et al.* [Liu *et al.*, 2016b] and Zhao *et al.* [Zhao *et al.*, 2017], respectively. The first set which we refer to as the (*Check-ins\_All*) set consists of all the check-ins of each user. The second set which is called the (*Check-ins\_Day*) set contains all the sequential POI check-ins which are organized by day. As shown in Figure 2, we observe that the average distances between POIs in the (*Check-ins\_All*) set and the (*Check-ins\_Day*) set are about 2.98 and 5.74 times smaller, respectively, than the average distance between all the POI pairs in the (*Global*) set. These results demonstrate that POIs visited by each user are close in proximity and are geographically influenced by each other. They also show that Instagram users tend to visit POIs located near POIs the users have already visited.

#### Textual Influence on POIs

We qualitatively analyze the relationship between words and POIs in the dataset. We measure the correlation between the words by calculating the Jaccard similarity of POIs in their text content. We first remove stop-words and construct a set of POIs for each word. Only POIs whose text content mentions a word more than five times are included in the POI set of the word. For example, when word  $w_1$  is used in the text content of POIs  $l_1, l_2,$  and  $l_3$ , the POI set of word  $w_1$  is  $\{l_1, l_2, l_3\}$ . We make the POI sets for the top 2000 words sorted by frequency, and calculate the average Jaccard similarity value for all the pairs of words (*Top\_2000\_Words*). We also compute the similarity values of the following word

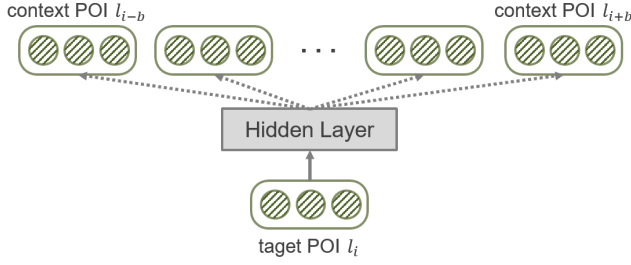


Figure 4: The illustration of the check-in context layer.

pairs: (*coffee, cafe*), (*baseball, Yankees*), and (*cafe, baseball*). The results are shown in Figure 3. As expected, (coffee, cafe) and (baseball, Yankees) have a much higher Jaccard similarity than (cafe, baseball) as the former pairs share similar semantic contexts and thus have larger POI overlaps. It is also interesting to see that the Jaccard similarity of (cafe, baseball) is substantially lower than the average Jaccard similarity of all the pairs in the *Top\_2000\_Words* set. This result demonstrates that the text content correlates with the POIs, which motivates us to use the text content in our POI embedding.

## 4 Method

In this section, we describe our content-aware hierarchical POI embedding model CAPE, which captures not only the geographical influence of POIs, but also the characteristics of POIs from text content. CAPE consists of a check-in context layer and a text content layer.

### 4.1 Check-in Context Layer

As explained in the data analysis section, POIs visited by each user, especially those which are consecutive in the user’s check-in context, are geographically influenced by each other. To capture this geographical influence of POIs, the check-in context layer employs the Skip-gram based POI embedding model proposed in the study by Liu *et al.* [Liu *et al.*, 2016b]. For a better understanding of the check-in context layer, we define some data concepts as follows.

**Definition 1 (Check-in Data)** Each check-in data  $d$  consists of user  $u$ , POI  $l$ , text content  $x$ , and check-in time  $t$ . We denote dataset  $D$  and check-in data  $d$  as  $D = (d_1, d_2, \dots)$  and  $d = (u, l, x, t)$ , respectively.

**Definition 2 (User Check-in Information)** The user check-in information is a list of POIs sorted by check-in time. The check-in information of the user  $u \in U$  is defined as  $q_u = (l_1, l_2, \dots)$  where  $U$  is the set of users in  $D$ .

**Definition 3 (Target and Context POI)** In the user check-in sequence, the  $b$  POI visited before and after one target POI  $l_i$  is defined as the context POI for the target POI  $l_i$ .  $b$  is the pre-defined context window size.

The context layer is described in Figure 4. The context layer considers a user’s check-in information as a “sentence” and each POI in a sequence as a “word” to use Word2Vec [Mikolov *et al.*, 2013] for POI embedding. According to Zhao *et al.* [Zhao *et al.*, 2017], the POIs that are visited by a user in the same day are more related to each other. Based on

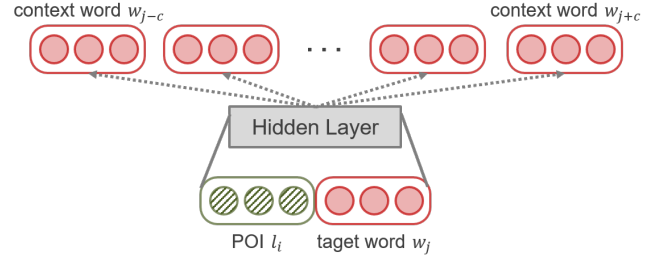


Figure 5: The illustration of the text content layer.

this, we treat a sequence of POIs visited by a user in the same day as a “sentence”. The check-in context layer trains the POI embedding vector by maximizing the objective function  $O_{ctx}$  in the following way,

$$O_{ctx} = \sum_{u \in U} \sum_{l_i \in q_u} \sum_{-b \leq j \leq b, j \neq 0} \log(P(l_{i+j}|l_i)). \quad (1)$$

Given the target POI  $l_i$ , probability  $P(l_{i+j}|l_i)$  of the context POI  $l_{i+j}$  is computed by using the soft-max function as follows,

$$P(l_{i+j}|l_i) = \frac{\exp(\mathbf{l}'_{i+j} \cdot \mathbf{l}_i)}{\sum_{k=1}^{|L|} \exp(\mathbf{l}'_k \cdot \mathbf{l}_i)} \quad (2)$$

where  $\mathbf{l}$  and  $\mathbf{l}'$  are input and output vectors of POI  $l$  respectively, and  $|L|$  is the number of POIs in the dataset. To speed up the learning process, we apply the negative sampling technique [Mikolov *et al.*, 2013], and the loss function for the check-in context layer  $loss_{ctx}$  is then defined as follows:

$$loss_{ctx} = - \sum_{u \in U} \frac{1}{|q_u|} \sum_{l_i \in q_u} \sum_{-b \leq j \leq b, j \neq 0} (\log(\sigma(\mathbf{l}'_{i+j} \cdot \mathbf{l}_i)) + \frac{1}{m} \sum_{\bar{l} \in L_{neg}} \log(\sigma(-\bar{\mathbf{l}}' \cdot \mathbf{l}_i))), \quad (3)$$

where  $\sigma(\cdot)$  is the sigmoid function,  $L_{neg}$  is the set of  $m$  negative POI samples, and  $\bar{\mathbf{l}}'$  is the output representation vector of negative sample  $\bar{l}$ .

### 4.2 Text Content Layer

Existing POI embedding models that utilize a user’s check-in information capture only the geographical influence of POIs, and thus fail to capture the characteristics of POIs. The text content of an Instagram post provides information on what activities people have done at each POI. To capture the characteristics of a POI from text content, we propose a text content layer. Before describing the layer, we define two more concepts as follows.

**Definition 4 (Text Content)** Each text content  $s_i$  consists of words, and is defined as  $s_i = \{w_1, w_2, \dots\}$  where  $w$  is a word. Each POI  $l_i$  contains multiple text contents written about the POI  $l_i$ . The set of multiple contents  $x_i$  about POI  $l_i$  is defined as  $x_i = \{s_1, s_2, \dots\}$ .

**Definition 5 (Target and Context Word)** In the text content  $s_i$ , each of the  $c$  words located before and after target word

$w_j$  is defined as the context of  $w_j$ .  $c$  is the pre-defined context window size for the text content.

The text content layer is described in Figure 5. We first treat a text content as a ‘‘sentence’’ and train the word embedding vector  $\mathbf{w}$  using Word2Vec [Mikolov *et al.*, 2013]. Then, the content layer trains the POI vector and word embedding vector together, maximizing the objective function  $O_{ctn}$ .

$$O_{ctn} = \sum_{l_i \in L} \sum_{s \in x_i} \sum_{w_j \in s} \sum_{-c \leq k \leq c, k \neq 0} \log(P(w_{j+k}|w_j, l_i)). \quad (4)$$

Given the POI  $l_i$  and target word  $w_j$  in the text content  $s$ , the objective function is computed using the sum of probability of the context word  $w_{j+k}$ . We formulate the probability  $P(w_{j+k}|w_j, l_i)$  using a soft-max function as follows:

$$\hat{\mathbf{w}}_j = \mathbf{l}_i \oplus \mathbf{w}_j, \quad (5)$$

$$P(w_{j+k}|w_j, l_i) = \frac{\exp(\mathbf{w}'_{j+k} \cdot \hat{\mathbf{w}}_j)}{\sum_{n=1}^{|V|} \exp(\mathbf{w}'_n \cdot \hat{\mathbf{w}}_j)},$$

where  $\oplus$  is a concatenate function,  $|V|$  is the number of words in the dataset.  $\mathbf{w}$  and  $\mathbf{w}'$  are input and output word representation vectors, respectively, in the soft-max layer. Similar to Equation 3, we formulate the loss function for the content layer  $loss_{ctn}$  using the negative sampling technique as follows:

$$loss_{ctn} = - \sum_{l_i \in L} \frac{1}{|l_i|} \sum_{s \in x_i} \sum_{w_j \in s} \sum_{-c \leq k \leq c, k \neq 0} ( \quad (6)$$

$$\log(\sigma(\mathbf{w}'_{j+k} \cdot \hat{\mathbf{w}}_j)) + \frac{1}{m} \sum_{\bar{w} \in V_{neg}} \log(\sigma(-\bar{\mathbf{w}}' \cdot \hat{\mathbf{w}}_j))),$$

where  $V_{neg}$  is the set of negatively sampled  $m$  words and  $\bar{\mathbf{w}}'$  is the output vector of negative sample word  $\bar{w}$ .

In this paper, we sample the words that are highly associated with a POI  $l_i$  as a text content  $s'_i$  to filter many meaningless words such as ‘‘selfie’’ and ‘‘follow’’. We compute the association weight between a word and a POI, which is similar to tf-idf, as follows:

$$association(l_i, w_j) = \frac{TF(w_j, l_i)}{DF(w_j)}, \quad (7)$$

where  $TF(w_j, l_i)$  is the frequency of  $w_j$  used in  $l_i$ 's text contents and  $DF(w_j)$  is the number of POIs whose text contents contain the word  $w_j$ . Then, we sample the top- $k$  words sorted by association weight for each POI. The loss function is modified as:

$$loss'_{ctn} = - \sum_{l_i \in L} \sum_{w_j \in s'_i} \sum_{w_k \in s'_i, k \neq j} (\log(\sigma(\mathbf{w}'_k \cdot \hat{\mathbf{w}}_j)) \quad (8)$$

$$+ \frac{1}{m} \sum_{\bar{w} \in V_{neg}} \log(\sigma(-\bar{\mathbf{w}}' \cdot \hat{\mathbf{w}}_j))),$$

where  $s'_i$  is the set of sampled words for POI  $l_i$ .

Finally, we combine the check-in context and text content layers. The final objective function of our content-aware POI embedding model CAPE is represented with hyperparameter  $\alpha$  as follows:

$$O_{emb} = O_{ctx} + \alpha \cdot O_{ctn} \quad (9)$$

CAPE simultaneously learns the geographical influence and semantical characteristics of POIs, maximizing objective function  $O_{emb}$ .

## 5 Experimental Evaluation

To qualitatively assess CAPE on the successive POI recommendation task, we conduct an experimental evaluation.

### 5.1 Experimental Setup

#### Successive POI Recommendation

Given a time-ordered user check-in sequence, recommendation models utilize the first (n-1) POIs as an input and predict the n-th POI as the recommended POI. We implement the following POI recommendation models that utilize user check-in sequences. (1) **STELLAR** [Zhao *et al.*, 2016], the tensor factorization based model which trains pairs of consecutive POIs to predict the next POI; (2) **LSTM**, a simple RNN based model using LSTM cells; (3) **GRU**, a simple RNN based model using GRU cells; (4) **ST-RNN** [Liu *et al.*, 2016a], the state-of-the-art POI recommendation model, which employs RNN and incorporates the time interval between two check-ins and the distance between two POIs as factors.

#### Baselines

We compare CAPE with the following four baseline POI embedding models. (1) **Random**, a simple baseline POI representation model, which uses a randomly initialized vector and is employed by most POI recommendation studies that use the check-in sequence data of users; (2) **SG** [Liu *et al.*, 2016b], a Skip-gram based POI embedding model which models user check-in sequences; (3) **POI2Vec** [Feng *et al.*, 2017], a hierarchical binary tree based model which utilizes spatial location information; (4) **Geo-Teaser** [Zhao *et al.*, 2017], a geo-temporal sequential embedding rank model, which uses both the geographical influence and temporal characteristics of each POI. We measure changes in performance when we apply each POI embedding model to the recommendation models.

#### Dataset Preparation

We evaluate our POI embedding model CAPE on our dataset described in the **Data Description and Analysis** section. To compare CAPE with the existing POI embedding baseline models, we divide the dataset into training, validation, and test sets using the same method employed in the study of Liu *et al.* [Liu *et al.*, 2016a]. The most recent check-ins which constitute 20% of the total check-ins of each user are used for the test set. The less recent check-ins which constitute 10% are used for the validation set, and the remaining check-ins which constitute 70% are used for the training set.

#### Metric

To compare the performance of the recommendation models, we employ two evaluation metrics: Recall@k and Mean Reciprocal Rank (MRR). **Recall@k** is a popular metric used for ranking tasks, and is computed when the correct successive POI is in the top-k recommended POI list. Since there is only one correct answer in our successive POI recommendation task, we do not employ F1-score@k (or Precision@k) used



Recommendation Models	Embedding Models	Recall@1	Recall@5	Recall@10	MRR
STELLAR	Random	0.1202	0.2199	0.2718	0.1785
	SG	0.1308	0.2251	0.2923	0.1857
	Geo-Teaser	0.1259	0.2227	0.3896	0.1812
	CAPE	0.1341	0.2384	0.2989	0.1899
LSTM	Random	0.1207	0.2225	0.2751	0.1805
	SG	0.1318	0.2344	0.2984	0.1875
	Geo-Teaser	0.1263	0.2298	0.2928	0.1821
	CAPE	0.1381	0.2412	0.3054	0.1939
GRU	Random	0.1197	0.2207	0.2726	0.1792
	SG	0.1356	0.2419	0.3040	0.1919
	Geo-Teaser	0.1291	0.2334	0.2980	0.1850
	CAPE	0.1390	0.2433	0.3079	0.1953
ST-RNN	Random	0.1054	0.2019	0.2426	0.1681
	SG	0.1185	0.2142	0.2529	0.1721
	Geo-Teaser	0.1145	0.2101	0.2489	0.1712
	CAPE	0.1245	0.2239	0.2601	0.1796

Table 2: Performance comparison between CAPE and baseline models in the preference-based setting. (\* We could not use POI2Vec [Feng et al., 2017] results because the authors did not make them available and our implementation could not reproduce the published results.)

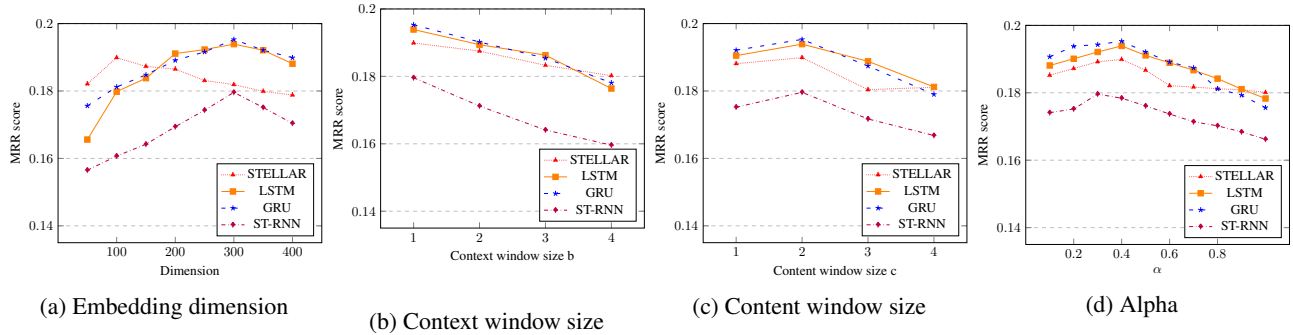


Figure 6: Effects of hyper-parameters.

in Liu *et al.* [Liu *et al.*, 2016a]. We report the Recall@k for k = 1, 5, and 10. **MRR** is used to measure the quality of ranked lists. When the next correct POI is highly ranked in the recommended list, the evaluation score is also high.

## 5.2 Evaluation Result

### Performance Comparison

Table 2 summarizes the experimental results evaluated using the metrics Recall@N and MRR. The results show that pre-trained POI embedding models improve recommendation performance. CAPE, SG, and Geo-Teaser outperform Random by 7.4%, 4.3%, and 1.8%, respectively, in terms of MRR, when applied to the recommendation models. Moreover, CAPE outperforms the POI embedding models in terms of the Recall@k and MRR metrics when applied to the recommendation models. In particular, CAPE outperforms Geo-Teaser even though CAPE does not use location information, which is used by Geo-Teaser. These results demonstrate that text contents provide useful information for understanding the characteristics of POIs. Unlike the results reported in the study by Liu *et al.* [Liu *et al.*, 2016a], the performance of ST-RNN, which utilizes consecutive check-in time intervals as temporal factors, was lower than that of LSTM. The average time interval between two consecutive check-ins in our Instagram dataset is about 10 times longer than that in the Gowalla dataset. Thus, the context of a target POI, which is the input to ST-RNN, cannot be properly constructed. When the input context is inappropriate, the state of ST-RNN is computed by only previous state. We report that CAPE, SG-CWARP, and

Geo-Teaser took about 3 hours 40 minutes, 3 hours, and 7 hours to be optimized, respectively, on a single GTX Titan X GPU machine.

### Hyper-parameter Optimization

We observe performance variance when we change the hyper-parameters of CAPE. The results are summarized in Figure 6. We first set the range of POI embedding dimension to 50-450. We then observe the performance changes from increasing the dimension in increments of 50. As shown in Figure 6a, the best performance is obtained when the dimension is set to 300, except in the case of STELLAR which achieves the best performance when the dimension is set to 100. Figure 6b and 6c show the variations in performance when we change the check-in context and text content window sizes, respectively. We set the ranges to 1-4, and observe the performance changes after increasing the window sizes by 1. When we set the context and content window sizes to 1 and 2, respectively, the best performance is obtained. CAPE optimizes the linearly combined  $loss_{ctx}$  and  $loss_{ctn}$  using a hyper-parameter  $\alpha$ . We also investigate the effect of hyper-parameter  $\alpha$ . The POI recommendation models, except ST-RNN, obtain the best performance using CAPE when the hyper-parameter  $\alpha$  is 0.4. ST-RNN achieves the best performance when the hyper-parameter is 0.3.

## 6 Conclusion

In this paper, we introduced our newly constructed large-scale dataset and investigated the necessity of using text content in

POI embedding. We also proposed CAPE which is a novel content-aware hierarchical POI embedding model for POI recommendation. CAPE captures the characteristics and geographical influence of POIs from the text content and check-in context of users. The experimental results showed that CAPE contributes to improving POI recommendation performance.

## Acknowledgements

This research was supported by the National Research Foundation of Korea (No. 2017R1A2A1A17069645, 2017M3C4A7065887).

## References

- [Chen *et al.*, 2014] Meng Chen, Yang Liu, and Xiaohui Yu. Nlpmm: A next location predictor with markov modeling. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 186–197. Springer, 2014.
- [Cheng *et al.*, 2013] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, volume 13, pages 2605–2611, 2013.
- [Feng *et al.*, 2017] Shanshan Feng, Gao Cong, Bo An, and Yeow Meng Chee. Poi2vec: Geographical latent representation for predicting future visitors. In *AAAI*, pages 102–108, 2017.
- [Liu *et al.*, 2013] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. Learning geographical preferences for point-of-interest recommendation. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1043–1051. ACM, 2013.
- [Liu *et al.*, 2016a] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*, pages 194–200, 2016.
- [Liu *et al.*, 2016b] Xin Liu, Yong Liu, and Xiaoli Li. Exploring the context of locations for personalized location recommendations. *IJCAI*, 2016.
- [Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [Rendle *et al.*, 2010] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820. ACM, 2010.
- [Wang *et al.*, 2015] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 403–412. ACM, 2015.
- [Wang *et al.*, 2017] Suhang Wang, Yilin Wang, Jiliang Tang, Kai Shu, Suhas Ranganath, and Huan Liu. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *Proceedings of the 26th International Conference on World Wide Web*, pages 391–400. International World Wide Web Conferences Steering Committee, 2017.
- [Yang *et al.*, 2013] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhu Wang. A sentiment-enhanced personalized location recommendation system. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, pages 119–128. ACM, 2013.
- [Ye *et al.*, 2011] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 325–334. ACM, 2011.
- [Zhao *et al.*, 2016] Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael R Lyu, and Irwin King. Stellar: Spatial-temporal latent ranking for successive point-of-interest recommendation. In *AAAI*, pages 315–322, 2016.
- [Zhao *et al.*, 2017] Shenglin Zhao, Tong Zhao, Irwin King, and Michael R Lyu. Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 153–162. International World Wide Web Conferences Steering Committee, 2017.