Unsupervised Translation via Hierarchical Anchoring: Functional Mapping of Places across Cities

Takahiro Yabe
Lyles School of Civil Engineering
Purdue University, USA
tyabe@purdue.edu

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

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

Yoshihide Sekimoto
Institute of Industrial Science
University of Tokyo, Japan
sekimoto@iis.u-tokyo.ac.jp

Satish V. Ukkusuri
Lyles School of Civil Engineering
Purdue University, USA
sukkusur@purdue.edu

ABSTRACT

Unsupervised translation has become a popular task in natural language processing (NLP) due to difficulties in collecting large scale parallel datasets. In the urban computing field, place embeddings generated using human mobility patterns via recurrent neural networks are used to understand the functionality of urban areas. Translating place embeddings across cities allow us to transfer knowledge across cities, which may be used for various downstream tasks such as planning new store locations. Despite such advances, current methods fail to translate place embeddings across domains with different scales (e.g. Tokyo to Niigata), due to the straightforward adoption of neural machine translation (NMT) methods from NLP, where vocabulary sizes are similar across languages. We refer to this issue as the *domain imbalance problem* in unsupervised translation tasks. We address this problem by proposing an unsupervised translation method that translates embeddings by exploiting common hierarchical structures that exist across imbalanced domains. The effectiveness of our method is tested using place embeddings generated from mobile phone data in 6 Japanese cities of heterogeneous sizes. Validation using landuse data clarify that using hierarchical anchors improves the translation accuracy across imbalanced domains. Our method is agnostic to input data type, thus could be applied to unsupervised translation tasks in various fields in addition to linguistics and urban computing.

CCS CONCEPTS

• **Computing methodologies → Spatial and physical reasoning**; **Machine translation**;

KEYWORDS

neural machine translation, embeddings, hierarchical structures, human mobility, mobile phone data

ACM Reference Format:
Takahiro Yabe, Kota Tsubouchi, Toru Shimizu, Yoshihide Sekimoto, and Satish V. Ukkusuri. 2020. Unsupervised Translation via Hierarchical Anchoring: Functional Mapping of Places across Cities. In *The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 4–8, 2020, San Diego, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330697

1 INTRODUCTION

Unsupervised machine translation has become a popular task in the natural language processing field due to the high cost of collecting large scale parallel data (e.g. [3, 11, 23]). In the urban computing field, large mobility datasets collected from mobile devices such as GPS trajectory data have allowed us to observe the dynamics of cities at an unprecedented spatio-temporal resolution and scale [7, 21]. Combined with recurrent neural network (RNN) models, recent studies have made significant progress in quantifying the functions of places in an analogical manner to word embeddings in the natural language processing field (e.g. [12, 16, 24, 27, 32]). Such high dimensional embeddings (or representations) of places have been shown to effectively capture the complex functions of places within cities [31], and have been applied in various downstream tasks in urban planning, such as identifying spatial clusters with respect to functionality [28], choosing sites for opening new stores [25], and predicting where users will go to in future timesteps [10].

A recent study adopted unsupervised language translation methods into the urban computing field to share knowledge and insights among different cities [26]. Several unsupervised neural machine translation methods developed in the natural language processing field were tested to perform translation of places across cities. However, due to the rather straightforward adoption of the translation methods, further validation showed that the translation method perform poorly across cities with different scales (e.g. Tokyo with 30M residents and Niigata with 0.8M). One possible reason of this failure was the significant imbalance in the scales of the source and target domains, since the scales of cities have much larger variance than that of vocabulary sizes across languages [19]. This *domain imbalance problem* is a key issue that needs to be solved to translate place embeddings across cities (Figure 1). Solving this issue could also potentially benefit unsupervised translation tasks in various fields of research in addition to languages and cities, where the source and target domains could have significantly different scales.

KDD '20, August 4–8, 2020, San Diego, CA, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330697

Analysis of mobility patterns within cities around the globe using novel mobility datasets have revealed various interesting properties of urban structures [17], including fractal properties [6], scaling laws [9], and hierarchical organization [13]. In particular, a recent study revealed positive connections between the hierarchical properties of cities and key urban indicators including higher use of public transport, higher levels of walkability, lower pollutant emissions per capita and better health indicators [5].

In this study, we attempt to overcome the aforementioned *domain imbalance problem* that exist in unsupervised translation tasks with an innovative method that utilizes the hierarchical structures that are common across domains of different sizes. We demonstrate our approach and its effectiveness through the example of unsupervised translation of place embeddings across cities with varying scales. We propose a translation model that aligns the vector spaces of the source and target domains using the hierarchical structure common across both domains. The model performances are tested using real world mobility data collected from mobile phones in 6 cities of varying scales in Japan, and are validated using landuse data. Results show that our methods are able to accurately translate place embeddings across cities, especially under the domain imbalance problem setting, where the urban scales are significantly different.

The key contributions of this paper are as follows:

- To the best of our knowledge, this study is the first to address the *domain imbalance problem* in unsupervised embedding translation tasks, and to present a method to overcome the problem.
- We propose a novel unsupervised translation method that leverages the common hierarchical structures across domains to generate effective anchor points.
- We verify that our method can successfully improve the unsupervised translation accuracy of place embeddings across cities with varying sizes, using real world mobility data from 6 heterogeneous cities.

2 PRELIMINARIES

*Definition 1 (**Human Mobility Data**).* Sequences of users' staypoint locations with timestamps are extracted from mobility data using methods explained in Section 3.1. The usual human mobility patterns of a city c is the set of all staypoint sequences of individuals whose home location belongs to city c.

*Definition 2 (**Place Embeddings**).* A city c is divided into disjoint cells by grid sizes of $r = 500$ meters. We will call each cell as a place i, and denote its representation as x_c^i, which is a d-dimensional vector. Place embeddings x_c^i are learned from the human mobility patterns observed in city c, using methods explained in Section 3.1. Embeddings of all places are stacked as a $(d \times n_c)$ matrix X_c, where n_c is the number of places in city c.

*Definition 3 (**Domain Imbalance Problem**).* The problem where there is significant imbalance in the sizes of the source and target domains in unsupervised translation tasks. Although this issue is rare in language translation tasks due to similar vocabulary sizes across languages, it is a critical problem when translating place embeddings across cities, due to the scale-free nature of city sizes.

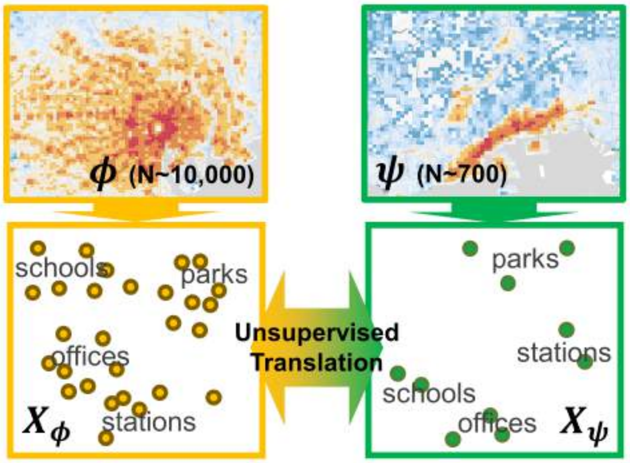

Figure 1: Illustration of the *domain imbalance problem* setting, where our objective is to translate embeddings across domains with significant imbalance in vocabulary sizes in an unsupervised manner.

*Problem Definition (**Unsupervised Translation of Place Embeddings**).* Place embeddings X_c are learned independently for each city c from the observed mobility patterns. Thus, for different cities, the vector spaces are not shared. Translating place representations from city ϕ to city ψ is equivalent to finding a mapping function f that aligns the two vector spaces, i.e., $X_\psi \approx \tilde{X_\phi} = f(X_\phi)$ in an unsupervised manner. Methods used to translate place representations are explained in Section 3.2 and 3.3.

3 METHODOLOGY

3.1 Generating Place Embeddings from Human Mobility Trajectories

We first extract human mobility patterns in each city from the location data observed from mobile phones. Each observation of the location data contains the user ID, timestamp, longitude and latitude. More details of the mobile phone data that we use in this study are explained in Section 4.1.1. Our goal is to extract users' sequences of staypoint locations from the observations. We achieve this by setting two threshold parameters; one spatial threshold and one temporal threshold. To cope with noisy location observations (e.g. spatial errors in GPS data), we perform mean shift clustering to estimate the true location for each observation, as described in previous studies (e.g. [4]). For each user, we read their location data in time order, and search for locations where the user has stayed within the distance defined by the spatial threshold parameter for a duration longer than the time defined by the temporal threshold parameter. We use 1000 meters as the spatial threshold, and 30 minutes as the temporal threshold in this study. As a result, we are able to obtain sequences of staypoint locations for each user, which will be used to generate place embeddings using methods explained in the following section.

To obtain the embeddings of places in a city, we solve a self-supervised task in which an Long Short-Term Memory (LSTM)

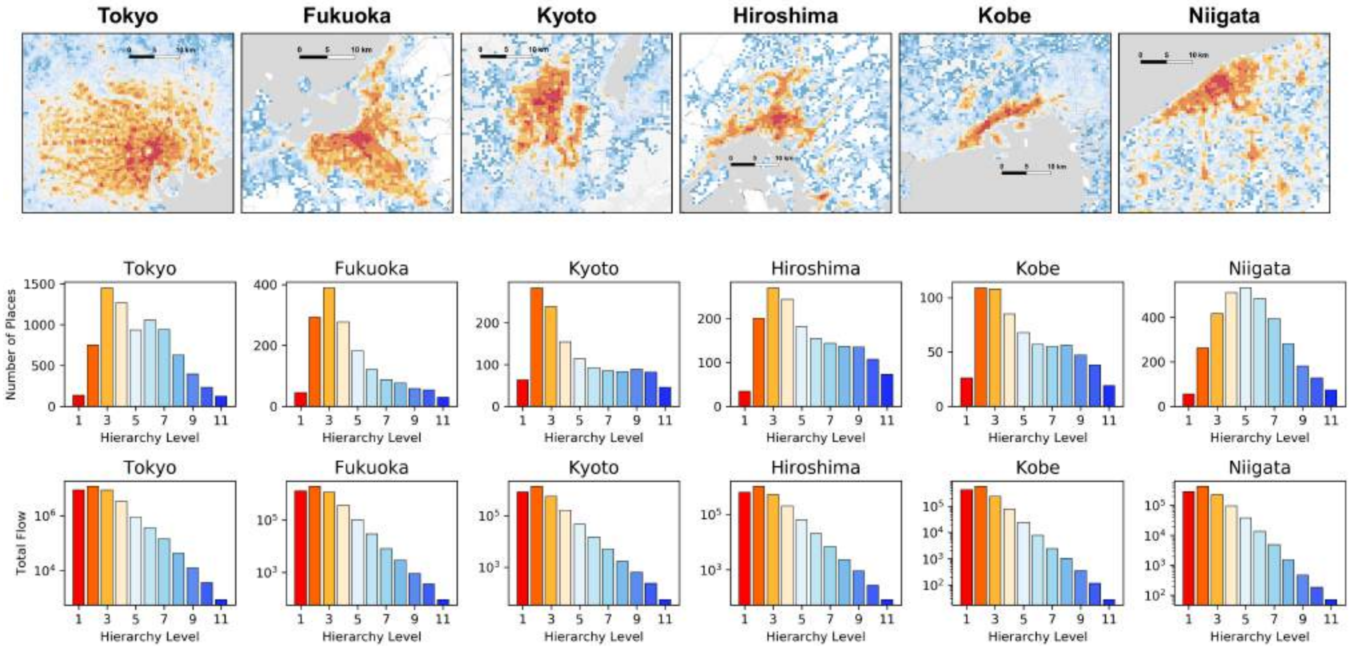

Figure 2: Universal hierarchical structure of cities used in our study.

RNN model is trained to predict the next staypoint of a user using mobility data, which is analogous to language models which are trained to predict the next word in a sentence. After training an LSTM RNN model using staypoint sequences of a city c, we extract and stack the embedding layer's parameters of the size $n_c \times d$, and define it as the matrix of place embeddings X_c. We refer to this place embedding learning model as "`IndivLSTM`" in the following sections. Specific model hyperparameter settings are explained in Section 4.2.1.

3.2 Analysis of Hierarchical Structure

One popular method of determining the hierarchical structures in cities is to iteratively apply the Loubar method proposed in [17], which uses the Lorentz curve of the number of visits to each location [5]. The Lorentz curve, which is a standard notion in the economics domain, is a cumulative distribution function of a distribution of datapoints. Given the average daily visit count values for all places in a city, we first sort the datapoints by ascending order and denote them as $(n_1 < \cdots < n_i < \cdots < n_{N_c})$, where n_k is the daily visit count in the k-th popular place in city c, and N_c is the total number of places in city c. The Lorentz curve is constructed by plotting the proportion of the places $F = \frac{i}{N_c}$ on the horizontal axis and the cumulative proportion of the covered visit counts L, which is calculated by the following equation. An example of the Lorentz curve is shown in the Supplementary Material (Figure A1)

$$f(i) = \frac{\sum_{j=1}^{i} n_j}{\sum_{j=1}^{N_c} n_j} \quad (1)$$

If the visit counts of all places were equal, the Lorentz curve would be a linear diagonal function.

The minimum threshold value of the first hierarchical level is computed by taking the intersection between the tangent of $f(F)$ at point $F = 1$ (i.e. the maximum value of the Lorentz curve) and the horizontal axis ($f(F) = 0$). In Louail et al. [17], the computed minimum threshold value was used to classify places in a city into "hotspots" and other places. Bassolas et al. [5] extended this method in an iterative manner to find multiple minimum thresholds for different hierarchical layers. After extracting the places in hierarchical level l, those places are excluded from the data distribution, and the minimum threshold value is recalculated using the new distribution to extract the places in hierarchical level $l + 1$. This procedure is iterated until all of the places in the city are assigned to a hierarchical level. For a more detailed explanation on the methods of urban hierarchical structure analysis, please refer to Bassolas et al. [5].

Figure 2 shows the estimated urban hierarchical structures in each city used in this study. The first row shows the colored maps of each city, where the colors indicate the hierarchical level each place belongs to (red: hierarchical level 1, blue: hierarchical level 11). The second row shows the histogram of the number of places belonging to each of the hierarchical levels in each city. The third row shows the total number of visits observed in the places belonging to each layer, which is calculated as $\sum_{j:l(j)=L} n_j$ for hierarchical level L where $l(j)$ denotes the hierarchical level of place j. While the second row highlights the different distributions of the number of places in each hierarchy (i.e. Tokyo and Fukuoka are more shifted to the left with more high-level places compared to Niigata), the distribution of the total number of visits in each hierarchical

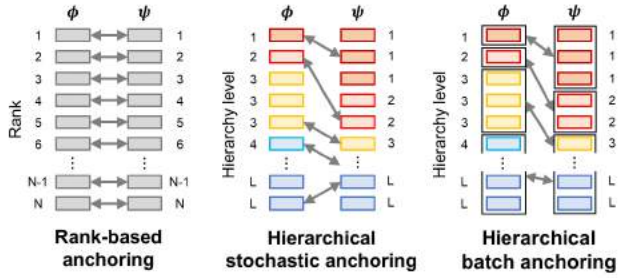
ϕ
ψ
1
2
3
4
5
6
N-1
N
Rank
Hierarchy level
L
Rank-based anchoring
Hierarchical stochastic anchoring
Hierarchical batch anchoring

Figure 3: Illustrative explanation of how the embeddings across the pair of cities are aligned in different methods.

level are strikingly similar across all cities, where the majority of the visits are concentrated in the first couple of hierarchical levels in all cities. This common hierarchical characteristic across cities motivates us to exploit the urban hierarchical structures. In the next section, we explain how we take advantage of this common hierarchical structure in our method for translating embeddings across imbalanced domains.

3.3 Translation via Hierarchial Anchoring

In previous studies in the natural language processing field, various methods have been proposed to obtain the best rotation matrix $R \in \mathbb{R}^{d\times d}$ that maps 2 embedding matrices $X_\psi, X_\phi \in \mathbb{R}^{d\times N}$ in an unsupervised manner. A previous study on unsupervised translation of place embeddings showed that a rank-based Procrustes alignment performed best out of the various methods [26]. Although this method was shown to be successful in cases where the source and target domains were of similar scales, a straightforward application to domain imbalanced settings could be problematic (and we show in the experiments that this is indeed the case). To overcome the difficulty in translation of embeddings under domain size imbalance (e.g. cities with different sizes), we propose a hierarchical alignment strategy to map the two domains. The main idea is to generate anchoring points based on hierarchical levels.

To perform translation, we first create anchoring embedding matrices, which serve as reference points to compute the optimal alignment operators. Figure 3 illustrates the different methods to create anchoring embedding matrices across cities. The left panel shows the **rank-based anchoring** method, which generates a one-to-one matching based on the sorted rank of places to generate anchoring pairs to align the embeddings. The center panel shows the **hierarchical stochastic anchoring** approach, where anchoring pairs of embeddings are selected within each hierarchical level in a stochastic manner with a predefined probability p, and are stacked together to obtain the anchoring embedding matrices for the two domains. The right panel shows the **hierarchical batch anchoring** approach, which instead of randomly selecting the embedding pairs, the mean vectors of the embeddings in each hierarchical level are computed and stacked to generate the anchoring embedding matrices, which are used to find the best alignment operator.

To find the optimal alignment operator using the anchoring embedding matrices, we test the Orthogonal Procrustes alignment and Affine alignment methods. Given the anchoring embedding matrices X^*_ϕ and X^*_ϕ for cities ϕ and ψ, respectively, Orthogonal Procrustes alignment computes the rotational matrix that optimizes the following equation:

$$R^* = \underset{R^T R = I}{\text{argmin}} \left\| RX^*_\phi - X^*_\psi \right\|_F \tag{2}$$

where, $R \in \mathbb{R}^{d\times d}$ is the optimal rotational matrix, and $\|\cdot\|_F$ is the Frobenius norm. Affine alignment introduces an extra transformation vector that increases the model complexity, and solves the following problem:

$$A^*, b^* = \underset{A,b}{\text{argmin}} \left\| (AX^*_\phi + b) - X^*_\psi \right\|_F \tag{3}$$

where, $A^* \in \mathbb{R}^{d\times d}$ is the optimal rotational matrix and $b^* \in \mathbb{R}^d$ is the optimal transformation vector. The optimization can be performed using standard solvers using least squares method. In the experiments, we test the effectiveness of different combinations of anchor embedding matrix generation methods (rank-based, hierarchical stochastic, and hierarchical batch), and alignment methods (Orthogonal Procrustes and Affine alignment).

Table 1: Statistics showing the varying scales of the cities

Scale	City	# Users	# Steps.	# Places (Urban)
Large	Tokyo	308,140	43,498,760	8020 (589)
Medium	Fukuoka	41,111	7,288,330	1636 (84)
	Kyoto	29,920	4,867,294	1363 (30)
	Hiroshima	21,868	3,876,699	1741 (25)
	Kobe	17,172	2,704,310	676 (63)
Small	Niigata	10,619	2,156,353	3312 (8)

4 EXPERIMENTAL VALIDATION

4.1 Data

4.1.1 Mobile Phone Location Data. Yahoo Japan Corporation[1] collects location information of mobile phone app users in order to send relevant notifications and information to the users. The users in this study have accepted to provide their location information. The data are anonymized so that individuals cannot be specified, and personal information such as gender, age and occupation are unknown. Each GPS record consists of a user's unique ID (random character string), timestamp, longitude, and latitude. The data acquisition frequency of GPS locations changes according to the movement speed of the user to minimize the burden on the user's smartphone battery. The data has a sample rate of approximately 2% of the population, and past studies suggest that this sample rate is enough to grasp the macroscopic urban dynamics. Table 1 shows the statistics of the dataset collected for 6 cities that we focus on in this study. We observe that the cities are significantly imbalanced in terms of the number of mobile phone users, total step sizes, and the number of places classified as urban areas.

[1] https://about.yahoo.co.jp/info/en/company/

4.1.2 Land Use Data. To validate whether the translated place embeddings correctly capture the functionality of the places, we use the Urban Area Land Use Mesh Data[2] in the National Land Numerical Information Database[3] provided by the Ministry of Infrastructure, Land, and Transport and Tourism of Japan. The dataset divides all urban areas of the country into $100m \times 100m$ grid cells, and assigns one category to each grid cell out of 17 options. The 17 options include farmland, residential area, business district, parks, forests, factories, public facilities, water body, open spaces, roads, railways, golf courses, etc. Because the categories are very detailed, we categorize these landuse categories into 7 label types: high-rise buildings, low-rise dense residential areas, low-rise sparse residential areas, industrial areas, agricultural areas, public facilities and parks, and water bodies. We aggregate these data into our spatial scale ($500m \times 500m$), and label each place with the landuse label which has the majority number of pixels in that $500m \times 500m$ place.

4.2 Experiment Settings

4.2.1 Model Hyperparameters. To learn the place embeddings described in Section 3.1, we setup the model and input data with the following procedure. The model consists of the embedding layer, LSTM RNN block, readout layer, and the output layer. While the main input of the model is a sequence of staypoints representing a user’s movement, we added two supplementary values, which are the timestamp of when the user had entered that place and the duration time of the stay, to incorporate time-dependency of the users’ behavior. The embeddings of staypoints were set to 64-dimensional vectors. The timestamp and stay duration were converted to 8-dimensional and 4-dimensional vectors respectively, and the three vectors at each step were concatenated into a 86-dimensional vector. The LSTM RNN block scanning over the embedding sequence consists of two layers of the size 128, and the hidden vectors of both layers were fed into the readout layer of the size 64, which were then read by the output layer producing the probability distribution over staypoints for the next place prediction. The parameter matrix of the staypoint embedding was reused as the output layer’s matrix to reduce the total number of parameters and make the training data usage more efficient. We applied dropout with the keep probability 0.8 to three points of the model: the embedding layer, readout layer, and output layer. We continued the training for 20 epochs, evaluated performance on the validation data at the end of each epoch, and used the embedding matrix of the best model for subsequent processing.

4.2.2 Comparative Methods. We compare the translation performances of the methods described in Section 3.3, as well as a state-of-the-art method used in language translation tasks. The combinations of the anchor embeddings matrix generation and alignment methods are as follows: rank based + Procrustes (**RP**), rank based + Affine (**RA**), hierarchical stochastic anchoring + Procrustes (**HSP**), hierarchical stochastic anchoring + Affine (**HSA**), hierarchical batch anchoring + Procrustes (**HBP**), and hierarchical batch anchoring + Affine (**HBA**). For the stochastic anchoring methods, results using $p = 0.5$ are reported since this probability had the best performance out of all 0.1 incremental values of p. In addition, we test **`JointLSTM`**, which applies the `IndivLSTM` model to all of the 6 cities together on the self-supervised next staypoint prediction task. We merge the mobility datasets of the cities into one and train the model over the merged data to obtain the place embeddings of all cities. To allow the model to treat places of the two cities as equally as possible, we mask the candidates of ψ at the output when the next staypoint belongs to ϕ and vice versa, releasing the model from the burden of distinguishing between two cities. The rationale behind this approach is that, places with similar functions will be visited in a similar manner (e.g. time of day, day of week, after and before certain places) regardless of the city, and that the mobility patterns of people are common across different cities. A previous study shows that this approach is effective in translating word embeddings of one language to another in an unsupervised manner [23].

Table 2: Quality of generated place embeddings measured by land use classification accuracy.

Method	Cities					
	Tokyo	F’oka	Kyoto	Hiro.	Kobe	Niig.
`IndivLSTM`	0.691	0.809	0.794	0.827	0.675	0.748
`JointLSTM`	0.679	0.780	0.724	0.710	0.639	0.735
Random	0.341	0.383	0.353	0.437	0.315	0.507

4.2.3 Evaluation Metrics. To evaluate the performance of the translation methods, we test the prediction accuracy of landuse classification using the translated place embeddings. Embeddings and landuse labels from the source city are used as training data, and the embeddings and landuse data from the target city are used as test data. We denote the place embeddings of the source (ϕ) and target (ψ) cities as X_{ϕ} and X_{ψ}. Similarly, we denote the landuse labels of each place in the source and target cities as y_{ϕ} and y_{ψ}. We also denote the translated place embeddings of city ϕ as $f(X_{\phi}) := \tilde{X_{\phi}}$ using the translation function $f(\cdot)$. We first train the landuse label classifier using the labels (y_{ϕ}) and translated place embeddings from the source city ($\tilde{X_{\phi}}$). Then, we test the predictive accuracy of landuse labels (y_{ψ}) using the trained classifier and the place embeddings from the target city (X_{ψ}). If the embeddings are perfectly translated and mapped into the target city, the classifier would be able to classify the landuse labels using the test data similarly as the training data. We use logistic regression as the classifier, and since the problem is a multi-class classification task, we use accuracy and F1-score as the evaluation metrics. The default hyper-parameter of the logistic regression model was set to $C = 1$, but we clarify that the ranking of the performances of the various translation methods do not depend on the choice of the classifier or the hyper-parameter.

4.3 Results

4.3.1 Quality of Generated Place Embeddings. Before performing any translation task, we check that the place embeddings produced by the two LSTM models described in Section 3.1 are of high quality. Table 2 shows the land use classification accuracy using the produced place embeddings in each city. As previously explained, logistic regression was used to classify the land use labels using only the place embeddings as features. Training and test data were

[2] http://nlftp.mlit.go.jp/ksj/gml/datalist/KsjTmplt-L03-b-u.html

[3] http://nlftp.mlit.go.jp/ksj/

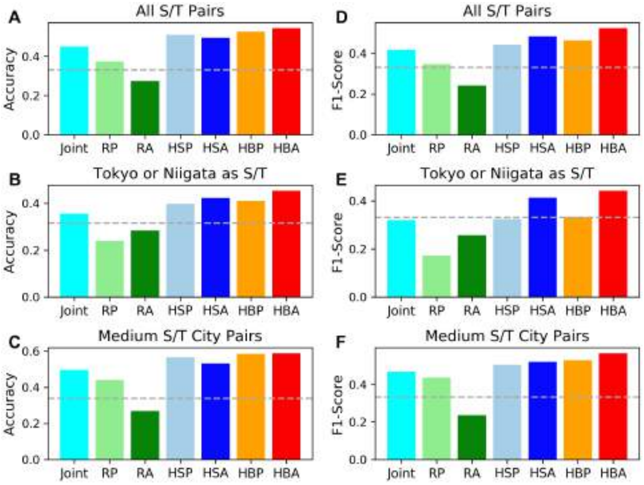

Figure 4: Translation performance of methods, measured by mean accuracy (A-C) and mean F1-score (D-F) of landuse prediction. (A) and (D) show results across all city pairs. (B) and (E) show results of translation when source or target city is either Tokyo or Niigata. (C) and (F) show performances when source and target cities are both medium sized.

randomly shuffled and split into 80% and 20% of the data, and the reported accuracy results are the mean values of 10 trials. Details of the experiment settings are noted in Section 2.6 of the Supplementary Material. We observe that for all cities, despite some differences across cities, both the place embeddings generated by `IndivLSTM` and `JointLSTM` are able to encode landuse information well, as previously shown by various studies (e.g. Zhang et al. [31]). The results also show that the quality of place embeddings drop using the `JointLSTM` model compared to the `IndivLSTM` model, since the `JointLSTM` model shares model parameters across all cities. We note that the unsupervised translation methods are agnostic of the place embedding generation methods. In the unsupervised translation experiments, we use the place embeddings produced by these LSTM-based models.

4.3.2 Translation Accuracy. In this study, we quantitatively evaluate the translation accuracy of each method using the predictive performance of the landuse labels, which is a multi-class classification task. Figure 4 shows the translation performances of the proposed method (red) and the comparative methods (Table 1). In all panels, the horizontal dashed gray line shows the accuracy when we use randomized labels. The left column presents the performances using accuracy, and the right column uses the weighted F1-score. The top row shows the mean performance metrics of place embedding translation across all city pairs, whereas the panels in the center row and the bottom row show the performances when Tokyo or Niigata (large or small cities) are either or both the source or the target city, and when both the source and target cities are medium sized cities (Fukuoka, Kyoto, Hiroshima, or Kobe), respectively. Most importantly, we observe that our proposed translation method that performs Affine alignment using hierarchical batch anchoring (“HBA”) performs best in all of the cases. Using the hierarchical structures for anchoring performs better than using rank based anchoring (“RP” and “RA”) in all of its variants (“HBP”, “HSP” and “HSA”). The joint learning approach (“Joint”) performs better than the random baseline in most cases, however its performances are limited compared to the hierarchical approaches. The rank-based Procrustes approach (“RP”), which was shown to perform well in a previous study across cities with similar sizes [26], performs well across medium source and target city pairs in this study as well (panels C and F), however is inferior to the hierarchical anchoring approaches under domain imbalance.

To obtain a more detailed understanding of the translation accuracy across the cities with different scales, we plot the pairwise translation performances of the three main methods (JointLSTM, rank-based anchoring + Procrustes alignment, and hierarhical-batch anchoring + Affine) in Figure 5. The matrices show the predictive F1-scores from the source city (vertical axis) to the target city (horizontal axis), where warmer colors (red, orange) show higher predictive performances. The diagonal elements are colored white because there are no translation operations involved in predicting landuse labels of the same city. The matrices are divided into sections with black border lines, showing the boundaries between large, medium, and small cities. We can immediately observe a significant difference in predicting the target landuse labels in Tokyo (large city), where the RP (Rank-based Orthogonal Procrustes) method performs particularly poorly. Translation from medium cities to Niigata (small city) works better using our proposed method compared to the two other methods. One exception was the predictive accuracy from Tokyo to Niigata, where the RP method performed better compared to the hierarchical batch anchored Affine mapping. This phenomenon can be explained by looking at the sensitivity analysis conducted in the next subsection, where we point out the effects of selecting which hierarchical layers to use for translation on the performances.

4.3.3 Which Hierarchical Levels should we use? So far, we have clarified the effectiveness of our unsupervised translation approach that uses hierarchical anchoring. Here, we further conduct sensitivity analysis on the number of hierarchical layers used in our translation method. Figure 6A shows the relationship between the number of hierarchy levels used and the prediction accuracy in landuse classification task using HBA method for each source-target city pair. The red, blue, black, and green plots show the translation tasks with Tokyo as the source (“From Tokyo”), Tokyo as the target (“To Tokyo”), from Tokyo to Niigata, and tasks with other cities as source and target, respectively. We observe intuitive trends in translation across the source-target groups. The increasing trend in the red plots (Tokyo as source city) indicate that increasing the number of hierarchical levels and including more rural areas increases the translation accuracy to cities smaller than itself, whereas the decreasing trend in the blue plots (Tokyo as target city) indicate that increasing the number of hierarchical levels and including information from more rural areas in the more rural source city decreases the translation accuracy. In contrast to these dependencies of translation accuracy from and to Tokyo on the number of hierarchical levels, the translation accuracy stays consistent with respect to the number of hierarchical levels across source and target cities of similar scales.

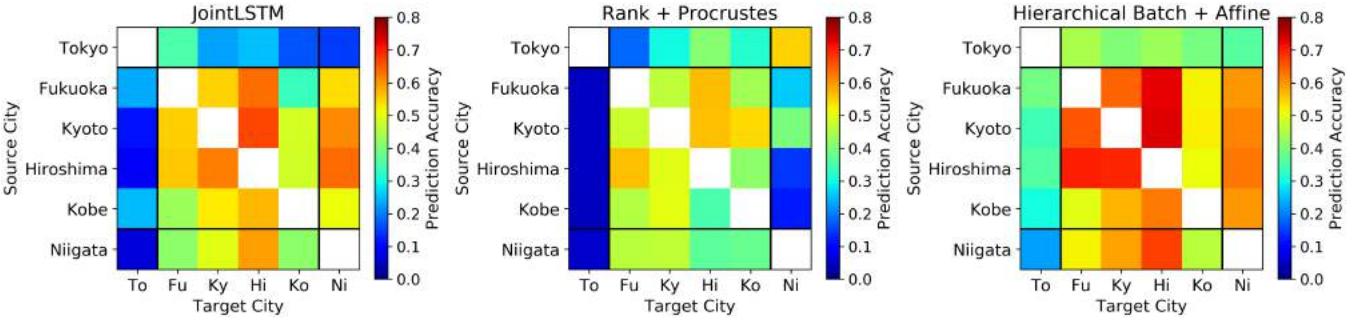

Figure 5: Mean pairwise translation accuracy across all source and target cities with different scales. Left: Accuracy using the `JointLSTM` method. Center: Accuracy using rank-based anchoring and Procrustes alignment method [26]. Right: Accuracy using hierarchical batch anchoring Affine alignment method (Proposed method).

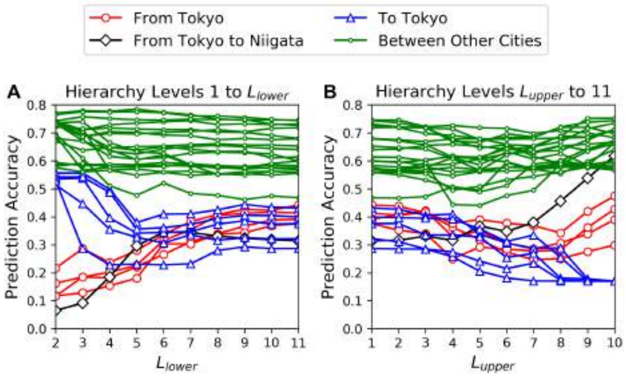

Figure 6: Sensitivity of translation accuracy with respect to the hierarchical levels used for translation using HBA. (A) Using hierarchical levels from the highest level $L = 1$ to L_{lower}. (B) Using hierarchical levels starting from L_{upper} to the lowest level $L = 11$. $L_{lower} = 11$ in panel A and $L_{upper} = 1$ in panel B correspond to the same case, where all hierarchical levels from $L = 1$ to $L = 11$ are used.

Figure 6B shows the inverse setting of Figure 6A, where we select only a subset of the lower hierarchical layers for translation. We observe that once again, the translation accuracy stays consistent with respect to the number of hierarchical levels across medium source and target cities. However, we observe that when using Tokyo as the source, the accuracy increases when we limit the hierarchical layers to lower layers (e.g. $L = 8, 9, 10$). In fact, although we observed a low translation accuracy for Tokyo → Niigata in Figure 5, we clarify that this was because we used the upper-level hierarchical information from Tokyo which was less relevant to Niigata. When we use Tokyo as the target, the translation accuracy drops as we throw away upper-level hierarchical information from the medium and smaller sized cities.

4.4 Case Study: Translating *Tsukiji Fish Market*

Finally, we qualitatively assess the translation performance through a case study of translating a point-of-interest (POI). Figure 7 shows the translation results of "Tsukiji Fish Market" from Tokyo to Hiroshima, Kyoto, and Niigata. Tsukiji Fish Market[4], one of the largest fish markets in Tokyo, is a very popular tourist spot for visitors and also for local residents. Each of the panels in Figure 7 show the similarity of each place to the translated Tsukiji Fish Market embedding $\tilde{x}_{Tsukiji}$. Given the norm distance of place i, denoted as $d(x_i) = \left\|x_i - \tilde{x}_{Tsukiji}\right\|_2$, the similarity is computed by normalizing the norm distances with respect to all the places in the city. Normalized similarity is computed as $S(i) = \frac{\max d(x_i) - d(x_i)}{\max d(x_i) - \min d(x_i)}$. Places colored in bold red color indicate high proximity close to $S(i) = 1$ with minimum norm distance, and the POIs inside those places are annotated in the maps. We can observe that for all the cities, we are able to detect large scale shopping malls (e.g. Aeon Malls in all cities) and even the Nishiki market[5] in Kyoto and Niigata Fish Market, which are popular markets for purchasing local products, via translation of places.

5 DISCUSSIONS

In this study, we proposed a novel unsupervised translation method that exploits the hierarchical structure that exist across different domains to enable translation of embeddings across domains of imbalanced sizes. The effectiveness of our method was shown through experiments using real data collected from 6 Japanese cities with varying sizes. It was interesting to observe that hierarchical batch anchoring worked better than hierarchical stochastic anchoring in all experiment settings. This implies that the hierarchical anchoring works best with fewer but less noisy anchor points. Although the joint learning method is considered to be one of the state-of-the-art methods in unsupervised translation tasks in the language domain, our analysis showed that the method using hierarchical structures worked better under domain imbalance settings. The key assumption of the joint learning method is that the mobility patterns (or sentences in the language domain) have similar structures across different cities. However, as we can see from Table 1, the average length of staypoints per user differed significantly across cities of different sizes (e.g. Tokyo: 141 steps/user, Niigata: 203 steps/user), indicating that such assumptions do not hold in cities.

[4] https://en.wikipedia.org/wiki/Tsukiji_fish_market

[5] https://en.wikipedia.org/wiki/Nishiki_Market

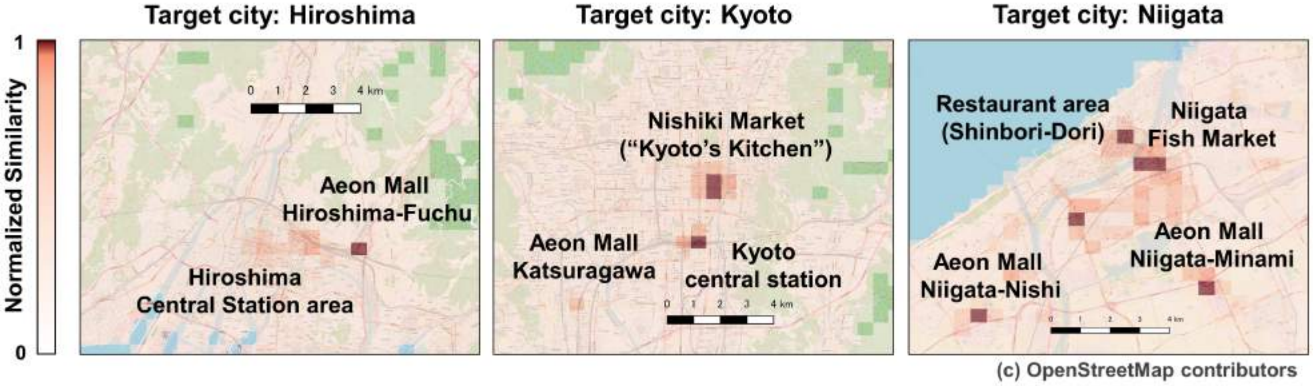

Figure 7: Case study showing the results of translating Tsukiji Fish Market (Tokyo) to Hiroshima, Kyoto, and Niigata via hierarchical batch anchoring and Affine alignment. We were able to translate the fish market in Tokyo into large scale shopping malls and local markets across cities of different scales.

In addition to the improvement in landuse label prediction tasks using the translated place embeddings, further analysis on using different combinations of hierarchical levels for translation in Section 4.3.3 provided interesting insights and possible reasoning on the translation performances of the proposed method. Figure 6 shows the strong dependence of translation accuracy on the hierarchical layers we use for translation. In general, it was found that when translating from a large city (e.g. Tokyo) to smaller cities, using the full set of hierarchical levels is optimal. In the extreme domain imbalance case (from Tokyo to Niigata), it was found that limiting information to only the bottom 2 hierarchical levels produced best translation accuracy. On the other hand, when we translate from smaller cities to larger cities, using information from only the higher hierarchical levels was often sufficient and better than using information from all of the layers in the smaller cities. Although these findings match our intuition, further investigation needs to be done in finding rules and methods in choosing the optimal ranges of hierarchical levels that we should use for translation, given the sizes of the source and target domains.

We believe this study leads to many research questions worthy of investigation. Representation (or embedding) learning has become a large branch of machine learning in recent years [8], and its techniques have been applied to various data types, including graphs [15] and images [18]. A natural extension of this study would be to apply our method to unsupervised translation tasks using embeddings generated from other types of data, such as language translation where vocabulary sizes significant vary across the languages. Since hierarchical structure analysis is agnostic to data, it can be easily extended to other problem settings. For example in the language setting, vocabulary can be grouped into hierarchical levels based on their appearance frequencies. Applying the translated place embeddings to solve various downstream urban problems would be another broad research direction. For example, selection of appropriate locations to open new stores has been a popular problem in urban computing [25]. Applying the translation results of place embeddings, such as the example shown in Figure 7 on the Tsukiji Fish Market, may assist planning of new store locations.

6 RELATED WORKS

6.1 Place Embedding Generation

Learning the place embeddings have been a popular research topic in the field of urban computing [33], often as a subproblem for larger tasks such as POI recommendation [10] and site selection problems [25]. Recent developments in the natural language processing field (e.g. Mikolov et al. [20]) has inspired many studies on place representation learning. Models such as `SkipGram` and `POI2vec` have applied ideas similar to `word2vec` on social media check-in data, where sequences of POIs are treated as sentences in the `word2vec` model [12, 16]. `CAPE` used both location and text data for POI embedding [10]. `Geo-Teaser` used the users' check-in and the geographical proximity of POIs for embedding generation [32]. `Place2vec` uses the physical proximity between POIs and the number of visit counts to perform POI embedding [27].

Recent studies such as `DeepMove` have extended such methods to large scale mobility data [24, 34]. `ZE-Mob` proposes a origin-destination coupled embedding model, where the assumptions are that origin and destinations of the same trip should have similar representations [28]. Zhang et al. [30] developed an unsupervised collective graph-regularized dual adversarial learning framework for multi-view graph representation learning. More recently, an LSTM based method that utilizes spatial hierarchy to produce fine grained place embeddings was proposed [22]. Our translation method is agnostic of the place embedding generation methods, thus we use a standard but effective LSTM model to generate place embeddings.

6.2 Unsupervised Neural Machine Translation

Unsupervised machine translation has become a popular topic in the representation learning literature, due to the difficulty of collecting large scale cross-lingual training data [2, 3]. This unsupervised setting applies to our problem setting, since we are not given any dictionary training data across cities. Studies take different approaches to unsupervised word translation tasks; Zhang et al. [29] uses an adversarial training approach, Conneau et al. [11] learns a linear transformation matrix to map words in one language to another, Artetxe et al. [1] applies better initial estimates using probability density functions of distances to other word embeddings, and Wada and Iwata [23] apply a shared LSTM model to jointly embed two languages. Cross-comparative studies of these methods have shown that the LSTM models works best under low resources. Hamilton et al. [14] apply a similar idea with Conneau et al. [11] to align word embeddings computed from corpus from different years, to understand the transition of the semantics of words over the years. A recent paper applied the methods developed in the language domain to translate place embeddings across different cities and tested their performances using real world data [26].

7 CONCLUSION

Despite the rising interest in unsupervised translation tasks, how to overcome the *domain imbalance problem* has been understudied. Using place embeddings and cities as an example problem setting, we propose and test a novel unsupervised translation method that exploits the hierarchical structures that are common across different domains despite scale differences. Experiments using data collected from 6 Japanese cities of different sizes clarified that our hierarchical anchoring approach improves the translation performance compared to previously proposed methods. Our method is agnostic to the type of input data, thus could be applied to unsupervised translation tasks in various fields in addition to linguistics and urban computing.

ACKNOWLEDGEMENTS

T.Y. is partly supported by the Doctoral Fellowship provided by the Purdue Systems Collaboratory. The work of T.Y. and S.V.U. is partly funded by NSF Grant No. 1638311.

REFERENCES

[1] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. *arXiv preprint arXiv:1805.06297* (2018).

[2] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Unsupervised statistical machine translation. *arXiv preprint arXiv:1809.01272* (2018).

[3] Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041* (2017).

[4] Daniel Ashbrook and Thad Starner. 2003. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous computing* 7, 5 (2003), 275–286.

[5] Aleix Bassolas, Hugo Barbosa-Filho, Brian Dickinson, Xerxes Dotiwalla, Paul Eastham, Riccardo Gallotti, Gourab Ghoshal, Bryant Gipson, Surendra A Hazarie, Henry Kautz, et al. 2019. Hierarchical organization of urban mobility and its connection with city livability. *Nature communications* 10, 1 (2019), 1–10.

[6] Michael Batty. 2007. *Cities and complexity: understanding cities with cellular automata, agent-based models, and fractals.* The MIT press.

[7] Michael Batty, Kay W Axhausen, Fosca Giannotti, Alexei Pozdnoukhov, Armando Bazzani, Monica Wachowicz, Georgios Ouzounis, and Yuval Portugali. 2012. Smart cities of the future. *The European Physical Journal Special Topics* 214, 1 (2012), 481–518.

[8] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.

[9] Luís MA Bettencourt, José Lobo, Dirk Helbing, Christian Kühnert, and Geoffrey B West. 2007. Growth, innovation, scaling, and the pace of life in cities. *Proceedings of the national academy of sciences* 104, 17 (2007), 7301–7306.

[10] 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 *IJCAI*. 3301–3307.

[11] Alexis Conneau, Guillaume Lample, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087* (2017).

[12] Shanshan Feng, Gao Cong, Bo An, and Yeow Meng Chee. 2017. Poi2vec: Geographical latent representation for predicting future visitors. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[13] Sebastian Grauwin, Michael Szell, Stanislav Sobolevsky, Philipp Hövel, Filippo Simini, Maarten Vanhoof, Zbigniew Smoreda, Albert-László Barabási, and Carlo Ratti. 2017. Identifying and modeling the structural discontinuities of human interactions. *Scientific reports* 7 (2017), 46677.

[14] William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. *arXiv preprint arXiv:1605.09096* (2016).

[15] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017).

[16] Xin Liu, Yong Liu, and Xiaoli Li. 2016. Exploring the Context of Locations for Personalized Location Recommendations.. In *IJCAI*. 1188–1194.

[17] Thomas Louail, Maxime Lenormand, Oliva G Cantu Ros, Miguel Picornell, Ricardo Herranz, Enrique Frias-Martinez, José J Ramasco, and Marc Barthelemy. 2014. From mobile phone data to the spatial structure of cities. *Scientific reports* 4 (2014), 5276.

[18] Xiaoqiang Lu, Xiangtao Zheng, and Yuan Yuan. 2017. Remote sensing scene classification by unsupervised representation learning. *IEEE Transactions on Geoscience and Remote Sensing* 55, 9 (2017), 5148–5157.

[19] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168* (2013).

[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[21] Carlo Ratti, Dennis Frenchman, Riccardo Maria Pulselli, and Sarah Williams. 2006. Mobile landscapes: using location data from cell phones for urban analysis. *Environment and Planning B: Planning and Design* 33, 5 (2006), 727–748.

[22] Toru Shimizu, Takahiro Yabe, and Kota Tsubouchi. 2020. Learning Fine Grained Place Embeddings with Spatial Hierarchy from Human Mobility Trajectories. arXiv:cs.LG/2002.02058

[23] Takashi Wada and Tomoharu Iwata. 2018. Unsupervised cross-lingual word embedding by multilingual neural language models. *arXiv preprint arXiv:1809.02306* (2018).

[24] Hongjian Wang and Zhenhui Li. 2017. Region representation learning via mobility flow. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 237–246.

[25] Mengwen Xu, Tianyi Wang, Zhengwei Wu, Jingbo Zhou, Jian Li, and Haishan Wu. 2016. Demand Driven Store Site Selection via Multiple Spatial-temporal Data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPACIAL ’16)*. ACM, New York, NY, USA, Article 40, 10 pages. https://doi.org/10.1145/2996913.2996996

[26] Takahiro Yabe, Kota Tsubouchi, Toru Shimizu, Yoshihide Sekimoto, and Satish V Ukkusuri. 2019. City2City: Translating Place Representations across Cities. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 412–415.

[27] 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*. ACM, 35.

[28] Zijun Yao, Yanjie Fu, Bin Liu, Wangsu Hu, and Hui Xiong. 2018. Representing Urban Functions through Zone Embedding with Human Mobility Patterns.. In *IJCAI*. 3919–3925.

[29] Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1959–1970.

[30] Yunchao Zhang, Yanjie Fu, Pengyang Wang, Xiaolin Li, and Yu Zheng. 2019. Unifying Inter-region Autocorrelation and Intra-region Structures for Spatial Embedding via Collective Adversarial Learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1700–1708.

[31] Yatao Zhang, Qingquan Li, Wei Tu, Ke Mai, Yao Yao, and Yiyong Chen. 2019. Functional urban land use recognition integrating multi-source geospatial data and cross-correlations. *Computers, Environment and Urban Systems* 78 (2019), 101374.

[32] 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*. International World Wide Web Conferences Steering Committee, 153–162.

[33] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 3 (2014), 38.

[34] Yang Zhou and Yan Huang. 2018. DeepMove: Learning Place Representations through Large Scale Movement Data. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2403–2412.

Supplementary Material

1 DATASET

1.1 Mobile Phone Location Data

In this study, we utilized location information collected by Yahoo Japan Corporation[1]. The users in this study have accepted to provide their location information. The data are anonymized so that individuals cannot be specified, and personal information such as gender, age and occupation are unknown. Each GPS record consists of a user's unique ID (random character string), timestamp, longitude, and latitude. The data has a sample rate of approximately 2% of the population, and past studies suggest that this sample rate is enough to grasp the macroscopic urban dynamics. For data requests, please contact Dr. Kota Tsubouchi of Yahoo Japan Corporation (ktsubouc@yahoo-corp.jp).

1.2 Landuse Label Data

We use the Urban Area Land Use Mesh Data[2] in the National Land Numerical Information Database[3] provided by the Ministry of Infrastructure, Land, and Transport and Tourism of Japan for validation. The dataset divides all urban areas of the entire country into $100m \times 100m$ grid cells, and assigns one category to each grid cell out of 17 options. We categorize these landuse categories into 7 label types: high-rise buildings, low-rise dense residential areas, low-rise sparse residential areas, industrial areas, agricultural areas, public facilities and parks, and water bodies. We aggregate these data into our spatial scale ($500m \times 500m$), and label each place with the landuse label which has the majority number of pixels in that $500m \times 500m$ place. The total counts and percentages of the landuse labels in each city are shown in Table A1.

2 METHODS

2.1 Determining City Boundaries

To extract the mobile phone users we collect location data from, we need to determine the city boundaries of each city. Table A2 shows the minimum and maximum coordinates used in this study.

2.2 Home Location Estimation

Home locations of all users were estimated using the collected mobile phone location dataset. Previous studies have shown that home locations of individuals can be detected with high accuracy by clustering the individual's stay point locations during the night [1]. We assume that each individual has one main home location in this study. The home location of each individual user was detected by applying the mean-shift clustering algorithm [2] to the nighttime stay points (observed between 8PM and 6AM), weighted by the duration of stays in each location.

[1] https://about.yahoo.co.jp/info/en/
[2] http://nlftp.mlit.go.jp/ksj/gml/datalist/KsjTmplt-L03-b-u.html
[3] http://nlftp.mlit.go.jp/ksj/

KDD '20, August 4–8, 2020, San Diego, CA, USA
2020. ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330697

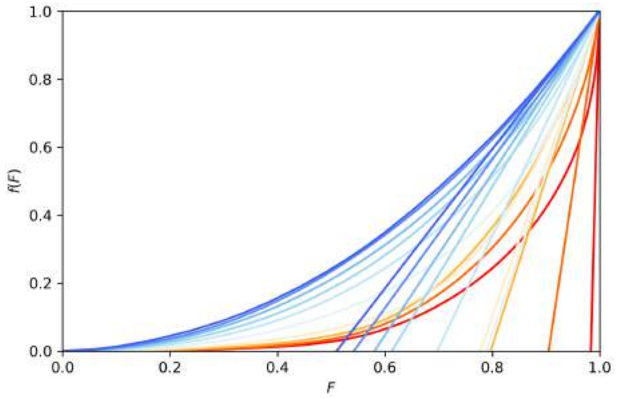

Figure A1: Example of how thresholds of hierarchical levels are determined. This shows the result for Tokyo. For each iteration, a Lorentz curve of the data distribution is drawn, and the threshold is computed by taking the intersection of the tangent line at $F = 1$ and the x-axis. Colors correspond to the hierarchy levels in Figure 2.

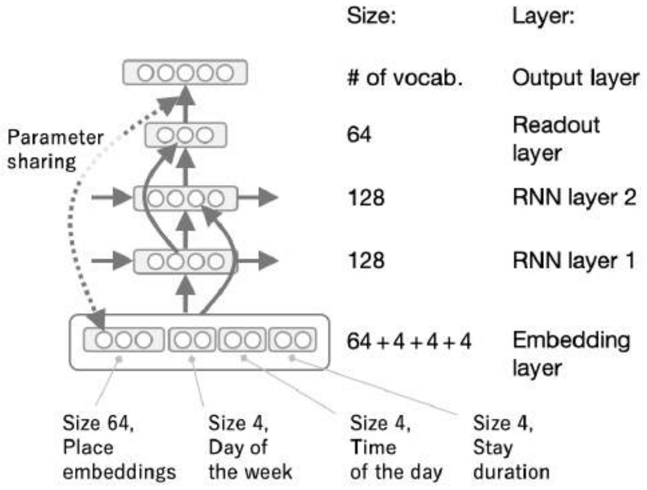

Figure A2: The overview of the LSTM RNN model used in representation learning of places.

2.3 Example of Estimating Hierarchical Levels

The places within a city is classified into each hierarchical level by using the Loubar method [3]. The methods are explained in Section 3.2 of the manuscript; we show an example plot (Tokyo) of how the thresholds are determined in Figure A1.

2.4 LSTM Model Architectures

2.4.1 LSTM for Individual Learning. To conduct the representation learning of places, we setup the model and input data as follows.

Table A1: Statistics showing the different landuse types in each city

City	**Total**	High-rise bldgs	Low-dense bldgs	Low-sparse bldgs	Industrial	Public/Parks	Agriculture	Water
Tokyo	8020	589 (7.3%)	1003 (12.5%)	4326 (53.9%)	73 (0.9%)	270 (3.3%)	1367 (17.0%)	392 (4.8%)
Fukuoka	1636	84 (5.1%)	40 (2.4%)	830 (50.7%)	29 (1.7%)	52 (3.1%)	574 (35.0%)	27 (1.6%)
Kyoto	1363	30 (2.2%)	147 (10.7%)	464 (34.0%)	14 (1.0%)	16 (1.1%)	639 (46.8%)	53 (3.8%)
Hiroshima	1741	25 (1.4%)	54 (3.1%)	578 (33.1%)	32 (1.8%)	17 (0.9%)	988 (56.7%)	47 (2.6%)
Kobe	676	63 (9.3%)	74 (10.9%)	134 (19.8%)	19 (2.8%)	41 (6.0%)	328 (48.5%)	17 (2.5%)
Niigata	3312	8 (0.2%)	51 (1.5%)	660 (19.9%)	32 (0.9%)	54 (1.6%)	2250 (67.9%)	257 (7.7%)

Table A2: Boundary Coordinates of the cities

City	Minimum (Lon, Lat)	Maximum (Lon, Lat)
Tokyo	(139.28, 35.59)	(139.92, 35.93)
Fukuoka	(130.30, 33.48)	(130.55, 33.74)
Kyoto	(135.65, 34.89)	(135.85, 35.11)
Hiroshima	(132.26, 34.27)	(132.60, 34.49)
Kobe	(135.13, 34.62)	(135.31, 34.77)
Niigata	(138.73, 37.72)	(139.27, 38.02)

The model consists of the embedding layer, LSTM RNN block, readout layer, and the output layer as shown in Fig A2. While the main input of the model is a sequence of staypoints representing a user’s movement, we added two supplementary values, which are the timestamp of when the user had entered that place and the duration time of the stay, to incorporate time-dependency of the users’ behavior. In the actual implementation, we treated the timestamp further decomposing it into two values: day of the week and time of the day. The embeddings of staypoints were set to 64-dimensional vectors. Day of the week, time of the day, and stay duration were discretized and converted to 4-dimensional vectors respectively, and the four vectors at each step were concatenated into a 86-dimensional vector. The LSTM RNN block scanning over the embedding sequence consists of two layers of the size 128, and the hidden vectors of both layers were fed into the readout layer of the size 64, which were then read by the output layer producing the probability distribution over staypoints for the next place prediction. The parameter matrix of the staypoint embedding was reused as the output layer’s matrix to reduce the total number of parameters and make the training data usage more efficient. We applied dropout with the keep probability 0.8 to three points of the model: the embedding layer, readout layer, and output layer. All models were implemented on Tensorflow[4].

2.4.2 LSTM for Joint Learning. We merge the mobility datasets of all the 6 cities into one, train the model over the merged data, and use the embedding layer matrix of the size $\left(\sum_c n_c\right) \times d$ as the representation matrix. The rationale behind this approach is that, representations of places with similar functions will be visited in a similar manner (e.g. time of day, day of week, after and before certain places) regardless of the city the places belong to. To let the model treat places of different cities as equally as possible, we mask the output logits of places in cities other than c which the next staypoint belongs to, releasing place embeddings at the output layer from the burden of distinguishing between cities. A previous study shows that this approach is effective in translating embeddings of one language to another in an unsupervised manner [4].

2.5 Training Procedure of LSTM Models

We trained the model of each configuration for 20 epochs, evaluated performance on the validation data at the end of each epoch, and used the embedding matrix of the best model for subsequent processing. To shorten the lead time, we parallelize the training process in the synchronous mode, using 4 workers in a 4-GPU environment. The mini-batch size was 16, and each time after processing 15 mini-batches, the parameter deltas are gathered to the parameter server and applied to the master model, and then the system moves forward to the next cycle, deploying the updated model and the next 15 mini-batches from the parameter server to each worker. Adam was used for the optimizer. We set the learning rate to 0.03 at the beginning and annealed it to 0.003 over the course of training.

2.6 Prediction of Land Use Labels

To evaluate the performance of the translation methods, we test the prediction accuracy of landuse classification using the translated place embeddings, as described in Section 4.2.3 in the manuscript. We use logistic regression as the classifier, and since the problem is a multi-class classification task, we use accuracy and F1-score as the evaluation metrics. The default hyper-parameter of the logistic regression model was set to $C = 1$. The algorithm was implemented using the scikit-learn package[5].

REFERENCES

[1] Francesco Calabrese, Giusy Di Lorenzo, Liang Liu, and Carlo Ratti. 2011. Estimating Origin-Destination flows using opportunistically collected mobile phone location data from one million users in Boston Metropolitan Area. *IEEE Pervasive Computing* 10, 4 (2011), 36–44.

[2] Yizong Cheng. 1995. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence* 17, 8 (1995), 790–799.

[3] Thomas Louail, Maxime Lenormand, Oliva G Cantu Ros, Miguel Picornell, Ricardo Herranz, Enrique Frias-Martinez, José J Ramasco, and Marc Barthelemy. 2014. From mobile phone data to the spatial structure of cities. *Scientific reports* 4 (2014), 5276.

[4] Takashi Wada and Tomoharu Iwata. 2018. Unsupervised cross-lingual word embedding by multilingual neural language models. *arXiv preprint arXiv:1809.02306* (2018).

[4] www.tensorflow.org

[5] https://scikit-learn.org/stable/