From Place2Vec to Multi-Scale Built-Environment Representation: A General-Purpose Distributional Embedding for Urban Data Analysis

Zhangyu Wang

University of California, Santa Barbara
Santa Barbara, California
zhangyuwang@ucsb.edu

Vahid Moosavi

Eidgenössische Technische Hochschule Zürich
Zurich, Switzerland
svm@arch.ethz.ch

ABSTRACT

Built environments like cities, roads, communities are rich sources of urban data. Many downstream applications require comprehensive analysis like geographic information retrieval, recommender systems, geographic knowledge graphs, and in general, understanding urban spaces [28]. Points of Interest (POI), as one of the most researched aspects of urban data, has been successfully modeled using concepts borrowed from Machine Learning (ML) and Natural Language Processing (NLP). In the work of Place2Vec [28], a Word2Vec-like statistical model is proposed to represent spatial adjacency with a continuous embedding space. This method successfully models the functional semantics of POIs with regard to several human-assessment based evaluations. However, though the Place2Vec model addresses the distributional heterogeneity within a given spatial context with ITDL augmentation, it does not address the spatial heterogeneity among different regions. To solve this problem, we propose to introduce a hierarchical, density-based, self-adjusting clustering mechanism. The boundary of relatedness and unrelatedness is learned from the given context, where denser areas have tighter bounds while sparser areas have looser ones. We train our model on both the baseline Yelp hierarchical dataset [28] and our OpenStreetMap dataset. We demonstrate that 1) our model significantly improves the performance on 2 of the 3 baseline tasks and the stability of training, and 2) our model generalizes excellently across 112 cities of radically different scales (minimum 1725 POIs, maximum 2694070 POIs), regions (North America, Europe, Asia, Africa) and types (commercial, touristy, industrial, etc.) without the need of adjusting or tuning any hyperparameters. We also demonstrate that our model can be used to discover interesting facts about cities like inter-city semantic analogy and intra-city connectivity, which can be very useful in urban planning, social computing and public policy making.

CCS CONCEPTS

• **Information systems** → **Information Retrieval**; • **Similarity measures**; • **Computer methodologies** → Machine learning;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

LocalRec'20, November 3, 2020, Seattle, WA, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-8160-4/20/11...\$15.00

<https://doi.org/10.1145/3423334.3431450>

KEYWORDS

Points of Interest, Similarity, Geo-Semantics, Machine Learning

ACM Reference Format:

Zhangyu Wang and Vahid Moosavi. 2020. From Place2Vec to Multi-Scale Built-Environment Representation: A General-Purpose Distributional Embedding for Urban Data Analysis. In *4th ACM SIGSPATIAL Workshop on Location-Based Recommendations, Geosocial Networks, and Geoadvertising (LocalRec'20)*, November 3, 2020, Seattle, WA, USA. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3423334.3431450>

1 INTRODUCTION

Understanding semantic similarity and relatedness of entities is a key component of many quantitative methods in geographical information retrieval, spatial data analysis and urban planning [10]. The approach of assigning measures, or intuitively, distances, to entities, falls into two major categories. One category relies on ontology trees, or, entity hierarchy, and defines distance with Minimum Spanning Tree (MST), shortest path, sub-tree clustering and other variants of tree structure similarity. The other category, however, is statistical and mostly data-driven. Introduced firstly to Natural Language Processing (NLP), distributional representation of entities in a large context becomes the indispensable founding block of all mainstream language models [21]. It is powerful in two ways. First, the distance space is continuous, which means you can compare any two given entities, even if they are not known relevant beforehand. This makes automated knowledge discovery on geographical data possible, because it does not assume any experts who have a prior knowledge (e.g., the ontology tree) but allows researchers to generate knowledge from raw data. Second, the distributional assumption is conditioned on the corpus, i.e., the data we collect for training. That means given different contexts, we can learn different measurements.

Word2Vec is a most widely used model in the second category. Its fundamental idea is that the meaning of a given entity (or, word) is defined by the co-occurring context. Analogously, we can define similarity and dissimilarity of geographical entities, or in practice, POI types, by spatial co-occurrence. We say "**Gas Station**" is similar to "**Car Wash**" because they both appear frequently along a freeway, while "**Gas Station**" is dissimilar to "**School**" since the neighborhoods they are present in are radically different. Place2Vec [28] is the first attempt to introduce Word2Vec to model the relatedness of POI types. Aside from a standard distributional algorithm, this work added a specially designed ITDL augmentation component to incorporate spatial distribution factors. By using default bin widths, the continuous neighborhood of a center POI is divided

into multiple discrete distance lags. POIs that fall into different lags contribute to the semantics of the center POI differently.

But there are two major problems. Firstly, the shape of neighborhood in Place2Vec is fixed to be circular, which is unreasonable. Secondly, though distance lags and importance decay estimation addresses the multi-scale property of POI types, it does not address the multi-scale property of built environments. The bin width of **Residence** in New York City should be at 10-meter level, while that in the Grand Canyon of Colorado should be at mile level. Not only in high-density and low-density areas, but also in places with different patterns, like American cities and Chinese cities, the reasonable mapping from spatial distance to functional relatedness are too different to be set at default values. If we want to apply Place2Vec to cities around the world, we will either have to manually tune hyperparameters for each dataset, or sacrifice model performance and robustness in exchange of computational simplicity.

Therefore we propose an unsupervised method that tackles these underlying problems. We employ MeanShift [5] to perform a density-based clustering of the raw POIs, so that more concentrated areas (CBDs, downtowns) have smaller cluster radii and more scattered areas (suburbs, rural areas) have larger ones. Conditioned on the clustering results, we learn embedding of POIs that is aware of the multi-scalability of both POIs and built-environments, called **Urban2Vec**.

The contributions of this paper are as follows:

- We propose Urban2Vec, a hierarchical density-based distributional embedding model.
- We demonstrate that Place2Vec model is not robust given different datasets, i.e., data from places that are different in spatial scales, POI distribution patterns, cultural and demographic backgrounds. On the contrary, Urban2Vec not only outperforms Place2Vec[28] in 2 of the 3 baseline evaluations (Yelp Hierarchy, Yelp Binary HIT, Yelp Ranking HIT) and excels in training stability, but also generalizes easily to 112 cities worldwide, without the need of tuning or adjusting any hyperparameters.
- We introduce 2 new sets of experiments: POI Semantics Experiments and Community Connectivity Experiments. In POI Semantics Experiments, we both qualitatively find examples of POI semantic analogy (e.g. $Sim_{Poland}(\text{Church, School}) \approx Sim_{Iraq}(\text{Mosque, School})$) and quantitatively analyze the structure of the embedding space. Results show that our method successfully preserves POI semantics. In Community Connectivity Experiments, we use vector similarity to model the connectivity of POIs. Results show that POI embedding can be used to model the formation of communities and detect city substructures, which is very important in urban planning, social computing and public policy making.
- We contribute 1) a new baseline dataset based on OpenStreetMap, 2) codes for training Urban2Vec embedding and visualizing results, 3) pretrained Urban2Vec vectors.

2 RELATED WORK

2.1 Distributional Word Embedding

Research on POI embeddings borrows concepts of neural network language models and distributional word embeddings from Natural

Language Processing (NLP) [4]. The basic idea of applying NLP models to geographical information problems is that they conform to similar distributional assumptions. The neural probabilistic language model assumes that given a target blank, the preceding word sequence S_p and the succeeding word sequence S_s , one can decide the probability of a word w_i appearing in the blank, $P(w_i|S_p, S_s)$. Similarly, if we consider each POI type as a "word", and the POIs surrounding it as the preceding and succeeding sequences, we can define a POI language model in the same way.

Due to the curse of dimensionality, most earlier language models set the length of S_p as 1, 2, or 3, and simply ignored the influence of S_s . This is usually called n -gram models, where n is the maximum number of preceding words we consider relevant to the conditional probability $P(w_i|S_p)$. Later, Word2Vec, the most successful and widely used distributed word representation, replaced the n -gram assumption with Skip-Gram and Continuous-Bag-of-Words (CBOW) [21]. In this revised model, we no longer consider a sequential conditional probability $P(w_i|S_p, S_s)$; instead, we consider the words appearing in a given window size to be the "context", and the context words contribute independently to the target word semantics. For example, if window size is 5, that means the context is the 5 words that precede the target word and the 5 words that succeed it. In Skip-Gram we give the model the target word and train the model to predict its surrounding words, while in CBOW the other way around. Theoretical analysis shows Word2Vec achieves similar optima as PMI (Pairwise Mutual Information) Matrix Factorization [15], which means Word2Vec is in principle modeling the co-occurrence of words in lower dimension spaces. This is very important for justifying the introduction of Word2Vec model to POIs, because the most intuitive measurement of POI relatedness is spatial co-occurrence.

2.2 Indirect Modeling of POIs

While there are many previous works that use Word2Vec in modeling geographical information, not all of them use it in the spatial co-occurrence way. Cocos and Callison-Burch [7] approach this topic in a reverse way: they augment texts with geolocation tags from Google Places and OpenStreetMap. Another similar work [31] also conditions texts on locations. Liu et al. [17] take travel routes instead of POIs as contexts. All of them are not modeling geographic entities but directly.

Other researchers do pay attention to POIs directly. Yao et al. [29] focus on urban land use and urban dynamics. Feng et al. [8] models user activities given a POI context. Zhao et al. [32] models temporal check-in trajectories. Zhang et al. [30] train a by far most sophisticated multi-modal representation model of human activities in urban spaces: CrossMap. It captures the spatial co-occurrence, the temporal consequence and the text co-reference simultaneously. For example, a tweet saying "**We went to the stadium this evening for Kobe Bryant**" contains an event that associates spatially to **Stadium**, temporally to **Evening** and textually to **Kobe Bryant-Basketball Player**. Then one can compute the relatedness between "**We went to the live house tonight for a rock band**" and the aforementioned event. Interestingly, in this paper, though they factually put text at the core of the model, they also employ MeanShift when estimating local hotspots. While they

use the same density estimation strategy as in our paper, they are not following a Word2Vec schema for learning latent representation. Instead, they design 2 objectives, ReconEmbed and GraphEmbed, mainly CRFs, to supervise the model. One problem of this complicated probabilistic model is that spatial, temporal and textual correlations are all embedded into a single joint vector space. More experimental evaluation and theoretical analysis is needed to demonstrate how well it performs if we only consider spatial distribution. In the experiment part of our paper we will briefly show that the much simpler Word2Vec model plus MeanShift density estimation is already good enough for POI representation.

2.3 Direct Modeling of POIs

Place2Vec is most closely related to our work. It focuses solely on POI spatial distribution, not involving temporal sequences or location-tagged texts. This is a good setting for studying POI distribution, because in most cases we either have no access to the full spatial-temporal-contextual triples that CrossMap requires for training, or the data are highly biased and imbalanced. For example, in China, the majority of social media contents are still produced by younger and wealthier users. If we assign too much weight on texts, we might overestimate the relatedness between **Shopping Mall** and **University**, while in real life the most important contributor of customer flow is the middle-aged local residents. Therefore we argue in order to learn a general-purpose, unbiased POI representation, we should use models that only capture spatial distribution patterns, like Word2Vec.

What Place2Vec adds to simple Word2Vec is the ITDL augmentation. It comes from a very intuitive motivation: for different POI types, the function that maps the absolute distance between two POIs to their relatedness should be different. For example, for small shops, 10-meter is a good step-size for relatedness decay; but for malls, even if they are miles away, it is still very probable that the relatedness is strong. Place2Vec uses different default bin width values for different POI types, which in their paper is referred to as "multiple spatial contexts". This is an effective way of tackling distributional disparity among different POI types, but it requires prior knowledge of optimal bin width values. Furthermore, even for the same POI types, when spatial contexts vary, the optimal bin width values should also vary. For example, suppose the bin width of **Restaurant** along Main Street, Flushing is optimal at 20 meters, it is not very reasonable to assume that value remains optimal somewhere near JFK International Airport. If we choose two cities of different states and of different scales, for example, New York City and a small town in Nevada, the discrepancy of optimal bin width values will be even larger. If we consider cities from different countries and of different cultural, demographic backgrounds, we should expect the bin width values usable for one to be completely unusable for the other. A quick example: in small towns of the US, residential blocks are usually quite far—beyond comfortable walking distance—to shopping malls, because the community layout is planned for driving; in small towns of China, however, residential buildings are closely intertwined with shopping facilities—sometimes they even overlap—because the majority of the residents do not own a private car and have to rely on walking distance shops for household purchase. If we think even one step further, we will see that such

difference of optimal hyperparameters is actually the knowledge we wish to discover from geographic data. If we already know the optimal hyperparameters, there is no need of "learning" things from data at all. A good model should not require us to know the data patterns a priori; it should detect such patterns and give us insights on the knowledge underlying the data on its own.

Therefore we propose the Urban2Vec model, which combines the MeanShift density estimation with the ITDL spatial augmentation. We focus on modeling only POI spatial distribution using Word2Vec, while the hyperparameters for ITDL augmentation are estimated with global density information.

3 DATASETS

We have two major sources of data, the Yelp Challenge Dataset and OpenStreetMap Dataset. They have very different structures and contents. The Yelp Challenge Dataset is the baseline dataset in Place2Vec [28]. It covers 11 cities from 4 countries (UK, Germany, Canada, US), 1030 POI types and 22 root categories. The most valuable feature of this dataset is that it provides a human-labeled, hierarchical POI type tree. We will use this data mainly for fairly comparing our model's performance with Place2Vec on the 3 baseline evaluations.

The OpenStreetMap dataset is collected and processed using our pipeline. It covers 114 cities, 38 countries and districts from all continents, 171 POI types and 15 root categories. This dataset is very diversified in POI distributional patterns, as it includes cities of very different economic development levels (New York vs. Kyiv), cultural, religious and demographic backgrounds (Warsaw vs. Baghdad vs. Tokyo), and scales (Los Angeles 2.9M POI entries vs. Albany 3K POI entries). See Section ?? for details. We will apply our extended semantic evaluations on this dataset.

4 METHODS

4.1 Latent Representation

Data representation is the founding block of all nowadays AI systems [3]. Latent representation, parametric or non-parametric, has long been incorporated in geospatial data science. Chopra et al. [6] use a simple feedforward regression model to learn from real estate transaction records the conditional distribution of house prices upon house locations. In general, one can consider latent representation learning as the process of decomposing the joint distribution $P(X)$ of random variable X into the combination of the marginal or conditional distributions of a set of unseen (thus the name "latent") variables $x_1, x_2, \dots, x_n$. The parametric version, usually referred to as *probabilistic graphical model*, relies on prior knowledge of latent variable types and correlations, but performs excessively well when data size is limited and the prior knowledge is accurate. The non-parametric version, on the contrary, sometimes referred to interchangeably as *embedding*, does not assume a strong prior structure. Instead, it assumes that the higher-dimensional ground-truth can be reconstructed by the combination of lower-dimensional features. This is related to PCA and Matrix Decomposition.

AutoEncoders [1] are one variant of non-parametric latent representation learning models. In most cases, the Encoder part of the AutoEncoder model takes in a higher-dimensional input, for example, a 784-dimensional (28x28) MNIST handwritten digit image, and

compresses this information into a 100-dimensional vector, which is called the "hidden state" or "latent vector". Then in reverse, the Decoder part takes in the 100-dimensional hidden state and tries to reconstruct the original 784-dimensional image as accurately as possible, measured usually by MSELoss. When the training is over, we discard the Decoder part and only keep the Encoder part. As the training process ensures that the original information can be restored from the lower-dimensional vector, we can use the hidden state directly as the representation of the original data. This process of compressing original data into hidden states is called *embedding*.

Word2Vec [20, 21] follows the same non-parametric schema as AutoEncoders, while it does not optimize the MSELoss, but the CrossEntropyLoss over all possible tokens. Unlike the AutoEncoder model, which is intrinsically adapted for continuous ground-truth such as pixels, Word2Vec is intrinsically discrete, which satisfies the distributional characteristics of natural language and POIs. In practice, we use the Word2Vec model with Skip-Gram, where the optimization objective is the cross entropy between the predicted context POI type distribution conditioned on the center POI type and the ground truth context POI type. Mathematically it is defined as follows:

$$L(\hat{P}_x, y) = - \sum_{c=1}^C [c = y] \log \hat{P}_x(c) \quad (1)$$

Here $\hat{P}_x$ is the learned conditional probability distribution over all possible POI types given center POI type x , y is the ground truth POI type, C is the number of POI types, and $[\cdot]$ is the indicator function that

$$[X] = \begin{cases} 1, & X \text{ is True} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

For computational simplicity, each POI type x is assigned a unique index i . We randomly initialize an $C \times H$ embedding matrix M_{center} , where H is the dimension of our learned latent vector, a hyperparameter that is usually set at 100 to 200 in language modeling. By simply choosing the i^{th} row of M_{center} , we obtain a $1 \times H$ input vector u_i for x . Then u_i is fed through a simple 1-layer linear feedforward neural network, which is mathematically equivalent to the right multiplication with the transpose of a $C \times H$ matrix $M_{context}$, yielding a $1 \times C$ score vector s_i , which is then passed through a Softmax layer to be normalized as a probability distribution. Then we can apply the aforementioned cross entropy loss. In some sense, this is also a "reconstruction" process, where we know the center POI type as the input information, and try to rebuild the true probability distribution of its surrounding POI types.

A standard SGD (Stochastic Gradient Descent) algorithm is used to optimize this parameters. When training is over, just like in AutoEncoders, we no longer need this learned model anymore. Instead, we take M_{center} and $M_{context}$ as the learned representation for POI types at the center and in the context respectively. Some research shows that by setting M_{center} and $M_{context}$ the same matrix, i.e., we do not distinguish the semantic roles of a center POI and a context POI, helps the model performance [22].

4.2 MeanShift as Density Estimation

MeanShift is a non-parametric clustering algorithm that finds regions containing a density peak. Compared with other clustering algorithms, for example KMeans and GMM (Gaussian Mixture Model), the most important property of MeanShift is that it does not require a preset cluster number. This is exactly the motivation of using clustering algorithm at all: we wish the model to detect the neighborhood boundaries on its own, without any prior knowledge.

Mathematically MeanShift is defined by a kernel density estimate and an iterative process. Firstly we need a probabilistic estimate of $\mathbf{x}$ being a cluster center given N observations $\{\mathbf{x}_n\}_{n=1}^N$ [26].

$$p(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N K\left(\frac{\|\mathbf{x} - \mathbf{x}_n\|}{\sigma}\right), \mathbf{x} \in \mathbb{R}^D \quad (3)$$

σ is the bandwidth and $K(\cdot)$ is the kernel function. Usually we use Gaussian kernels for computational efficiency. One form of Gaussian kernels is defined as follows:

$$K(t) = e^{-t^2/2} \quad (4)$$

After we have a kernel density estimate, we need an iterative process that under some mild assumptions finally converges to the mode. We define an iterative function f as follows:

$$f(\mathbf{x}) = \sum_{n=1}^N \frac{K'(\frac{\|\mathbf{x} - \mathbf{x}_n\|}{\sigma})}{\sum_{n=1}^N K'(\frac{\|\mathbf{x} - \mathbf{x}_n\|}{\sigma})} \mathbf{x}_n \quad (5)$$

$$K'(t) = \frac{dK}{dt} \quad (6)$$

If $K(\cdot)$ is a Gaussian kernel, after very simple evaluation we can rewrite $f(\mathbf{x})$ as follows:

$$f(\mathbf{x}) = \sum_{n=1}^N p(n|\mathbf{x}) \mathbf{x}_n \quad (7)$$

$$p(n|\mathbf{x}) = \frac{\exp(-\frac{1}{2} \|\frac{\mathbf{x} - \mathbf{x}_n}{\sigma}\|^2)}{\sum_{n'=1}^N \exp(-\frac{1}{2} \|\frac{\mathbf{x} - \mathbf{x}_{n'}}{\sigma}\|^2)} \quad (8)$$

Each iteration, we let $\mathbf{x}^{\tau+1} = f(\mathbf{x}^\tau)$. $f(\mathbf{x}) - \mathbf{x}$ is called *mean shift*, because it measures the averaged mode shift between two successive iterations. When $f(\mathbf{x}) - \mathbf{x} < \epsilon$, where ϵ is the preset tolerance, we say the iterative process converges. We propose to take the estimated clusters as context boundaries, that is, only POIs that fall into a same cluster are considered relevant. In this way we no longer have to set thresholds manually. This is based on the observation that cities are organized in "clusters"—POIs with stronger functional correlation usually form a denser area that can be separated from other POIs.

4.3 Hierarchically Estimated Spatial Importance Decay (HESID)

In Place2Vec [28] it is shown that using POI distance as simple thresholds (i.e., deciding whether a surrounding POI should be considered as the context) is not very helpful. The rationale is intuitive: *everything is related to everything else, but near things are*

more related than distant things. [30] Within a given radius, though all POIs are considered related to the center POI, obviously the POI that is only 5 meters away from the center implies stronger semantic correlation than that is 100 meters away. A naive revision is assigning each POI an importance factor, or, weight, that decays with distance. A simple formulation is as follows:

$$L(\hat{P}_x, y) = -w(d_{x,y}) \sum_{c=1}^C [c = y] \log \hat{P}_x(c) \quad (9)$$

$d_{x,y}$ is the Euclidean distance between the center POI x and the ground truth context POI y . The weight function $w(\cdot)$ can be any function that monotonically declines. Some reasonable instances could be:

$$w(t) = \frac{1}{t} \quad (10)$$

$$w(t) = \frac{1}{t^2} \quad (11)$$

$$w(t) = e^{-\frac{1}{T-t}} \quad (12)$$

where T is the context threshold.

It is also possible to use different distances, i.e., non-Euclidean distances. Shortest route distance, average traveling time and graph network based methods [2] may all apply to this setting. In this paper we will stick to Euclidean distances.

The naive importance decay is problematic in the way that it assumes for any pair of POI types, the speed the importance factor declines is identical. This assumption does not hold in most cases. Some POI types, for example, **Residence**, are very likely to be extremely close to each other and their influence on other POI types decays steeply, because they are usually planned in semi-closed blocks. Some POI types, for example, **Police Office**, in the other way, are distributed very sparsely, while their influence almost never decays. This motivation leads to the ITDL augmentation proposed in Place2Vec [28]. Though the ITDL augmentation takes a relatively complicated form, it is fundamental idea remains simple. In our paper, we will not discuss the specific version of ITDL as appeared in [28], but re-state it as a more generalized form.

In general, our purpose is to obtain an augmented weight function

$$w(d_{x,y}, x, y, S) \quad (13)$$

that takes the center POI type x , the context POI type y and the scope S as extra information to decide the appropriate importance decay of this specific POI pair (x, y) . Two major factors need to be included: the correlation between x and y and the information of y . Take a simple example: if on average, a **Residence** is 50 meters from a **Shop**, then it is reasonable to have a function that assigns 1 weights to (**Residence**, **Shop**) pairs with distance around 50 meters and larger than 1 weights to closer pairs and smaller than 1 weights to farther pairs. And a POI type that is dominant in a certain neighborhood by its own nature, for example, **ACCOMMO**, which is 10 times larger in number than the second most frequent POI type in OpenStreetMap dataset, does not provide much information and should be assigned lower weights. This is similar to stop word

removal and word frequency smoothing in language models. The scope S is the set of data we use to estimate these two factors.

Following the above analysis, we propose a simple yet effective augmented weight function:

$$w(x, y, C) = k \exp\left(\frac{\mu_{x,y} - d_{x,y}}{\sigma_{x,y}}\right) I_{y,S} \quad (14)$$

$$I_{y,S} = 1 + \sum_{i \in S} p_{y,i} \log p_{y,i} \quad (15)$$

In this function, k is a scaling factor, $\mu_{x,y}$ is the average distance between any two adjacent POIs of type x and type y and $\sigma_{x,y}$ the standard variance. By adjacent we mean there is no closer POI of type y to the given POI of type x . $I_{y,S}$ is the measurement of how much information y carries in the scope S using information entropy. I is factually modeling how independently y appears in S . If $I_{y,S}$ is very small, that means y appears mostly for its own sake, thus not carrying much information, and should be assigned lower weights.

Notice that this weight function is parametric on some prior statistical features, that is, $\mu_{x,y}$, $\sigma_{x,y}$ and $I_{y,S}$, estimated on S . If we assume S to be the entire dataset, our weight decay strategy will become very similar to the ITDL augmentation except that we use multiplication to associate the exponential term and the entropy term, while ITDL uses convex combination. But such assumption is dangerous. It is equivalent to the assumption that these globally estimated features fit everywhere in a local context. Unfortunately this doesn't hold. Even within the same city, for example, the POI type **Shop** will have smaller μ , smaller σ and smaller I in commercial districts and have larger μ , larger σ and larger I in residential districts. In other words, we need to ensure S is homogeneous.

To solve this problem, we propose the **Hierarchically Estimated Spatial Importance Decay (HESID)** algorithm. In brief, the MeanShift clustering is done in iterations; every iteration we cluster again with finer granularity on the clusters from the last iteration. In this way we obtain a cluster hierarchy, where sibling clusters are likely to have common statistical features that can be estimated on the parent cluster.

Algorithm 1: Hierarchically Estimated Spatial Importance Decay

Result: A weight dictionary W
 A set of bandwidths $\{b_1, b_2, \dots, b_m\}$;
 Dataset D , Clusters $\{D\}$;
 An empty weight dictionary W ;
while $i \leq m$ **do**
 $T \leftarrow \emptyset$ **while** C in Clusters **do**
 $\mu, \sigma, I \leftarrow \text{Estimate}(C)$;
 For every data point d in C ,
 $W[d] \leftarrow W[d] + (\mu, \sigma, I)$;
 end
 $T \leftarrow T + \text{MeanShift}(C, b_i)$;
end
 Clusters $\leftarrow T$

When the algorithm is finished, we obtain m sets of (μ, σ, I) estimates for each data point y . We use these estimates to compute m different weights and use a linear combination of them as the final weight. In this way, we incorporate all, hierarchically from global to local information to define an appropriate importance decay function.

5 EXPERIMENT AND RESULT

5.1 MeanShift Clustering and HESID

We apply the MeanShift algorithm to the OpenStreetMap dataset to validate its effectiveness of adjusting radii when contexts change. A qualified clustering should satisfy: 1) clusters at dense areas should have smaller radii and sparse areas larger ones, and 2) cluster sizes are of similar order of magnitude. Quantitatively we can count the mean and standard deviation of radii and sizes of the clusters that fall into different types of regions. The difficult part is, however, that we do not have good annotations of urban segmentation. Actually, even the simple task of defining city boundaries is still under research. Therefore, we qualitatively examine some examples of different scales and morphology. Figure 1 shows how our observation supports the assumption that the MeanShift algorithm be very effective and stable in adjusting to varying contexts.

We face the same difficulty of lacking reliable ground truth when quantitatively evaluating the effectiveness of hierarchical clustering. Similarly, we use Berlin as an example to illustrate how our HESID mechanism works. Figure 2 demonstrates that hierarchical clustering is especially effective in tackling complex, embedded structures: it divides the map into center blocks and outskirt blocks, while each center block and outskirt block are divided again into sub-centers and sub-outskirts. This is unable to be achieved by setting any uniform hyperparameter for a single clustering algorithm.

5.2 Yelp Hierarchy Evaluation

Yelp Dataset comes with an annotated POI hierarchy tree. Using this tree, we can define 6 different measurements of POI semantic similarity: Wu & Palmer Similarity (**WuPalmer**) [27], Leacock & Chodorow Similarity (**Leacock**) [13], Lin Similarity with Sanchez Information Content (**SimLin Sanchez**) [16, 23], Lin Similarity with Seco Information Content (**SimLin Seco**) [16, 24], Jiang & Conrath Distance with Sanchez Information Content (**Disjc Sanchez**) [11, 23] and Jiang & Conrath Distance with Seco Information Content (**Disjc Seco**) [11, 24]. See [28] for more details of evaluation. Figure 3 shows how embedding dimension and maximum training iteration affect the performance of our method. From the experiment results we see our method has worse, yet much stabler performance.

An observation can partly explain why our method is beaten in the Yelp Hierarchy Evaluation. [28] suggests that this evaluation is potentially faulty. Firstly, the hierarchy is created by a small set of people which leads to a bias. Secondly, the hierarchy tree is very shallow (average path length 1.73, maximum depth 5) which leads to a large number of draws in the ranking, thus corrupts the final MRR. Finally, this hierarchy tree is based on ontological taxonomy instead of spatial semantics. These mean we may be evaluating our embedding against a wrong ground truth.

Nevertheless, the stability of our method is obvious. Unlike the wavy performance of the Place2Vec baseline, our method (See Figure 4) shows a steadily upgoing curve when embedding dimension goes up until some threshold of overfitting (around 270) is reached, and such trend remains consistent given different training iterations and measurements.

5.3 Yelp Binary HIT Evaluation and Yelp Ranking HIT Evaluation

As is discussed in the last section, the Yelp Hierarchy Evaluation may be flawed. [28] proposed two Human Intelligence Tasks (HITs) using Amazon Mechanical Turk as a complement.

The Binary HIT Evaluation is composed of 80 triplets with each element in the triplet being a place type, and a "most dissimilar" type is marked out by human judges [28]. Our task is to find this most dissimilar type using pairwise vector similarity. This is a much simpler yet more robust task than the Yelp Hierarchy Evaluation. Given the triplet, the dispute over the choice of the "most dissimilar" will be largely resolved. There will also be no draws and the human judges will consider more about functional relations.

Analogously, the Ranking HIT Evaluation requires the human judges to rank the candidate POIs from the most similar to the most dissimilar. Yan et. al. [28] selected 10 place types and for each place type 7 candidate place types for ranking, altogether 70 POI type pairs. They asked Amazon Mechanical Turk to rate the pairs on a similarity scale of 1-7. We construct a pairwise vector similarity rank and compare it with the human rank. Average Spearman's Rank Correlation Coefficient (Spearman's ρ) is selected as the measurement.

As is depicted in Figure 5, our method significantly outperforms the baseline both in Spearman's ρ (on average **0.1/12%** better in Binary HIT Evaluation and **0.3/67%** better in Ranking HIT Evaluation) and in training stability.

5.4 OpenStreetMap POI Semantics Experiments

Like word analogy *king - man = queen - woman* in the field of natural language processing [21], POI embedding may also have similar linear compositionality. Levy et. al. [14] find that the underlying cause of word analogy is not based on the non-linear optimization

$$\operatorname{argmax}_{x \in V - \{a, b, c\}} (\cos(x, a) - \cos(x, c) + \cos(x, b)) \quad (16)$$

We can likewise define POI analogy as in Eq. 16. One possible example is *FoodStore - Food = Bar - Alcohol*. However, such analogy is very rare in POI data. Instead, we notice meaningful analogy across cities. For example, **Izakaya** in Japan is a good semantic analogy to **Bar** in Ireland. We define Inter-City POI analogy as

$$\operatorname{argmin}_{x \in V_A} (|\operatorname{Rank}_A(x, u) - \operatorname{Rank}_B(y, v)|) \quad (17)$$

where A, B are cities, u is a POI vector in city A , v and y are POI vectors in city B . Our objective is to find a POI vector x in city A that minimizes the difference between the similarity ranks of (x, u) and (y, v) . We wish to see that such $(x, u) - (y, v)$ semantic quadruple forms an assertion that "What x is to u in City A is like what y is to v in City B ".

Figure 1: Examples of Meanshift Clustering, OpenStreetMap POI Data

City A	POI Type x	POI Type u	City B	POI Type y	POI Type v
Warsaw	POW_Christian	EDU_School	Baghdad	POW_Islamic	EDU_School
Rome	TOURIST_Ruins	POI_Peak	Shanghai	POW_Buddhist	POI_Peak
Paris	TOURIST_Castle	ACCOMMO_Camping	Alexandria	TOURIST_Memorial	ACCOMMO_Camping
Athens	TOURIST_Art	TOURIST_Memorial	New York	ACCOMMO_Hostel	TOURIST_Memorial
New York	ACCOMMO_Hostel	TOURIST_Memorial	Alexandria	ACCOMMO_Camping	TOURIST_Memorial
Boston	FOOD_Cafe	SHOP_Alcohol	Alexandria	TRANSPORT_Airport	SHOP_Alcohol

Table 1: Examples of POI analogy with OpenStreetMap data

As we do not have gold standard annotations to do quantitative evaluations, we will report some most interesting findings as a qualitative proof. Table 1 gives a small portion of the POI analogies that reveals very inspiring facts. Examples show that our embedding captures very rich information underlying spatial distribution.

Apart from POI analogy, we also use t-SNE[18] to visualize the high-dimensional (dimension = 15 in this evaluation) embedding space on a 2D plot. We firstly divide the total 171 POI types into 3 groups: **Commercial** (blue), **Public** (red) and **Residence** (green). See Section ?? for details. Figure 7 shows that the general embedding space (trained on the data of all 112 cities) is well separated into 2 subspaces: **Commercial**(blue) in the left, **Public**(red) in the right and **Residence** mixed. This is a compelling evidence of our method’s effectiveness in preserving POI semantics.

Potential drawback may be that both OpenStreetMap Dataset and Yelp Dataset are biased. OpenStreetMap is dominated by tourist spots as it is labeled by volunteering travellers. Yelp Dataset, otherwise, is highly biased towards commercial facilities, especially

restaurants. Both datasets are not ideal for our proposed semantic preservation task.

5.5 OpenStreetMap Community Connectivity Experiments

Besides POI semantics, we also care about the connectivity among communities. Other than graph-based connectivity such as in community detection[12, 25], we propose a new form of connectivity: Markovian connectivity. Intuitively, it is the "easiness" of one community transforming into another. Formally, for transition between single POI types, it can be described by a transition matrix[9, 19]; for communities, it can be described by the cumulative transition probability along all possible paths. Figure 8 is an example.

From Community A (**Restaurant, House**) to Community B (**Bar, Hotel**) there are 4 possible paths in total: ((T1), (T4, T5)), ((T1), (T6)), ((T2, T3), (T4, T5)), ((T2, T3), (T6)), and the cumulative transition probability is $0.3 \times (0.2 \times 0.6) + 0.3 \times 0.1 + (0.4 \times 0.8) \times (0.2 \times 0.6) + (0.4 \times 0.8) \times 0.1 = 0.1364$. This is the Markovian connectivity from Community A to Community B. Notice it is directed.

Figure 2: Hierarchical Meanshift Clustering with OpenStreetMap POI Data, Berlin

With this concept of Markovian connectivity, we can deal with problems that graph-based connectivity models fail at. Algorithm 2 is an example of iterative algorithms to determine community boundaries. It expands the borders of the initial community each iteration, measures how possible the expansion is by the Markovian connectivity, and stops the process when the Markovian connectivity for the current expansion is lower than the given threshold or the community size reaches the upperbound.

An intrinsic problem with Algorithm 2 is its complexity. When the size of C , constant h and constant k increase, the number of possible paths explode exponentially. The simplest solution is to constrain h and k to a small enough number. In this paper, for the sake of computability, we set $h = 1$ and $k = 1$. That is, we only consider 1 change and 1 hop at a time. This can be easily described by a 1st order Markovian matrix[9].

We obtain the Markovian matrices for all 112 cities in OpenStreetMap Dataset by computing pairwise vector cosine similarity scores for all POI types and normalize them into probabilities. In order to discover more high-level connectivity patterns, we aggregate POIs of the same category (See Table ??). Plotting the category-level

Figure 3: Yelp Hierarchy-based Evaluation against baseline (dimension from 10 to 100)

Figure 4: Yelp Hierarchy-based Evaluation (dimension from 10 to 300)

connectivity matrix with a chord diagram (see Figure 9) we discover 2 types of cities with distinct patterns. **Type A**, mostly giant commercial metropolises, demonstrates strong tendency towards intra-category connectivity. The big brown arch in each diagram on the upper row implies huge, continuous mono-functional commercial complexes. **Type B**, demonstrating weaker intra-category connectivity but stronger inter-category connectivity, implies multi-functional and self-sustained communities. This is just the case with

Figure 5: Yelp HIT Evaluation against baseline (dimension from 10 to 100). Left: Binary HIT; Right: Ranking HIT

Figure 6: Yelp HIT Evaluation (dimension from 10 to 300). Left: Binary HIT; Right: Ranking HIT

Figure 7: T-SNE visualization of the general embedding space, OpenStreetMap data, dimension = 15

many medium-sized cities and cities with a strong tradition of fused communities.

6 CONCLUSION AND FUTURE WORK

This paper proves both theoretically and empirically that by incorporating density estimations, we can learn POI embedding stably

Figure 8: An example of community transition

Algorithm 2: Iteratively Expanding Community Borders

Result: A set of POIs C
 Full set of POIs S ;
 Initial set of POIs $C \subset S$;
 Add-on set of POIs $U \leftarrow \emptyset$;
 Constant h, k, p, N ;
 Markovian connectivity $mc \leftarrow 1$;
 Threshold t ;
 An algorithm $MC(A, B, h)$ that computes the Markovian connectivity from A to B in at most h hops;
while $mc \geq t$ or $\text{Size}(C) \leq N$ or $S - C \neq \emptyset$ **do**
 $C \leftarrow C + U$;
 $D \leftarrow C + \{\text{The closest } k \text{ POIs to } C \text{ in } (S - C)\}$;
 $U \leftarrow \text{Randomly draw } k \text{ samples (probability greater than } p) \text{ from } C$;
 $mc \leftarrow MC(C + U, D)$;
end
 return C ;

with better semantic preservation. We also explore the potential and ways of applying POI embedding to other fields of research, such as community detection, urban planning and social studies.

The biggest weak point of this paper is that we lack ground-truth for quantitative evaluations. In the future we need to 1) construct a more balanced dataset for training and, 2) annotate more data for evaluation and design more comprehensive tests and metrics.

REFERENCES

- [1] Pierre Baldi. 2012. Autoencoders, Unsupervised Learning, and Deep Architectures. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning (Proceedings of Machine Learning Research)*, Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor, and Daniel Silver (Eds.), Vol. 27. PMLR, Bellevue, Washington, USA, 37–49. <http://proceedings.mlr.press/v27/baldi12a.html>
- [2] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çağlar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. 2018. Relational inductive biases, deep learning, and graph networks. *CoRR* abs/1806.01261 (2018). arXiv:1806.01261 <http://arxiv.org/abs/1806.01261>
- [3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*

Figure 9: City Patterns

- 35, 8 (Aug. 2013), 1798–1828. <https://doi.org/10.1109/TPAMI.2013.50>
- [4] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.* 3 (March 2003), 1137–1155. <http://dl.acm.org/citation.cfm?id=944919.944966>
- [5] Miguel Á. Carreira-Perpiñán. 2015. A review of mean-shift algorithms for clustering. *CoRR abs/1503.00687* (2015). arXiv:1503.00687 <http://arxiv.org/abs/1503.00687>
- [6] Sumit Chopra, Trivikraman Thampy, John Leahy, Andrew Caplin, and Yann LeCun. 2007. Discovering the Hidden Structure of House Prices with a Non-parametric Latent Manifold Model. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*. ACM, New York, NY, USA, 173–182. <https://doi.org/10.1145/1281192.1281214>
- [7] Anne Cocos and Chris Callison-Burch. 2017. The Language of Place: Semantic Value from Geospatial Context. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, Valencia, Spain, 99–104. <https://www.aclweb.org/anthology/E17-2016>
- [8] Shanshan Feng, Gao Cong, Bo An, and Yeow Meng Chee. 2017. POI2Vec: Geographical Latent Representation for Predicting Future Visitors. <https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14902>
- [9] Paul A Gagnieu. 2017. *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons.
- [10] Krzysztof Janowicz, Martin Raubal, and Werner Kuhn. 2011. The semantics of similarity in geographic information retrieval. *J. Spatial Information Science* 2 (2011), 29–57.
- [11] Jay J. Jiang and David W. Conrath. 1997. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *Proceedings of the 10th Research on Computational Linguistics International Conference*. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP), Taipei, Taiwan, 19–33. <https://www.aclweb.org/anthology/O97-1002>
- [12] Andrea Lancichinetti and Santo Fortunato. 2009. Community detection algorithms: A comparative analysis. *Physical Review E* 80, 5 (Nov 2009). <https://doi.org/10.1103/physreve.80.056117>
- [13] Claudia Leacock and Martin Chodorow. 1998. Combining local context and wordnet similarity for word sense identification.
- [14] Omer Levy and Yoav Goldberg. 2014. Linguistic Regularities in Sparse and Explicit Word Representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Ann Arbor, Michigan, 171–180. <https://doi.org/10.3115/v1/W14-1618>
- [15] Omer Levy and Yoav Goldberg. 2014. Neural Word Embedding As Implicit Matrix Factorization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'14)*. MIT Press, Cambridge, MA, USA, 2177–2185. <http://dl.acm.org/citation.cfm?id=2969033.2969070>
- [16] Dekang Lin. 1998. An Information-Theoretic Definition of Similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML '98)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 296–304.
- [17] Kang Liu, Song Gao, Peiyuan Qiu, Xiliang Liu, Bo Yan, and Feng Lu. 2017. Road2Vec: Measuring Traffic Interactions in Urban Road System from Massive Travel Routes. *International Journal of Geo-Information* 6 (10 2017), 321. <https://doi.org/10.3390/ijgi6110321>
- [18] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.
- [19] Christopher D Manning, Christopher D Manning, and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT press.
- [20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:cs.CL/1301.3781
- [21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *CoRR abs/1310.4546* (2013). arXiv:1310.4546 <http://arxiv.org/abs/1310.4546>
- [22] Ofir Press and Lior Wolf. 2016. Using the Output Embedding to Improve Language Models. *CoRR abs/1608.05859* (2016). arXiv:1608.05859 <http://arxiv.org/abs/1608.05859>
- [23] David Sánchez, Montserrat Batet, and David Isern. 2011. Ontology-Based Information Content Computation. *Know.-Based Syst.* 24, 2 (March 2011), 297–303. <https://doi.org/10.1016/j.knsys.2010.10.001>
- [24] Nuno Seco, Tony Veale, and Jer Hayes. 2004. An Intrinsic Information Content Metric for Semantic Similarity in WordNet. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI'04)*. IOS Press, NLD, 1089–1090.
- [25] Stanislav Sobolevsky, Riccardo Campari, Alexander Belyi, and Carlo Ratti. 2013. A General Optimization Technique for High Quality Community Detection in Complex Networks. *CoRR abs/1308.3508* (2013). arXiv:1308.3508 <http://arxiv.org/abs/1308.3508>
- [26] M.P. Wand and M.C. Jones. 1994. *Kernel Smoothing*. Taylor & Francis. <https://books.google.com/books?id=GTOOI5yE008C>
- [27] Zhibiao Wu and Martha Palmer. 1994. Verbs Semantics and Lexical Selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics (ACL '94)*. Association for Computational Linguistics, USA, 133–138. <https://doi.org/10.3115/981732.981751>
- [28] Bo Yan, Krzysztof Janowicz, Gengchen Mai, and Song Gao. 2017. From ITDL to Place2Vec: Reasoning About Place Type Similarity and Relatedness by Learning Embeddings From Augmented Spatial Contexts. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '17)*. ACM, New York, NY, USA, Article 35, 10 pages. <https://doi.org/10.1145/3139958.3140054>
- [29] Yao Yao, Xia Li, Xiaoping Liu, Penghua Liu, Zhaotang Liang, Jinbao Zhang, and Ke Mai. 2017. Sensing spatial distribution of urban land use by integrating points-of-interest and Google Word2Vec model. *International Journal of Geographical Information Science* 31, 4 (2017), 825–848. <https://doi.org/10.1080/13658816.2016.1244608>
- [30] Chao Zhang, Keyang Zhang, Quan Yuan, Haoruo Peng, Yu Zheng, Tim Hanratty, Shaowen Wang, and Jiawei Han. 2017. Regions, Periods, Activities: Uncovering Urban Dynamics via Cross-Modal Representation Learning. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 361–370. <https://doi.org/10.1145/3038912.3052601>
- [31] Yating Zhang, Adam Jatowt, and Katsumi Tanaka. 2017. Is Tofu the Cheese of Asia?: Searching for Corresponding Objects across Geographical Areas. <https://doi.org/10.1145/3041021.3055132>
- [32] Shenglin Zhao, Michael Lyu, and Irwin King. 2018. *Geo-Teaser: Geo-Temporal Sequential Embedding Rank for POI Recommendation*. 57–78. https://doi.org/10.1007/978-981-13-1349-3_4

A DATASET STATISTICS

City	State/Country	POI Amount
los angeles	california	2694070
dallas	texas	1259599
amsterdam	netherlands	750857
birmingham	england	732321
rotterdam	netherlands	720743
tokyo	japan	655948
london	england	484007
dc baltimore	maryland	373173
prague	czech republic	353783
new york	new york	301112
berlin	germany	282898
pittsburgh	pennsylvania	272141
atlanta	georgia	261334
moscow	ruissia	230378
lyon	france	219138
oslo	norway	208825
minneapolis saint paul	minnesota	185943
hamburg	germany	178966
stockholm	sweden	175816
brussels	belgium	168712
miami	florida	167049
paris	france	163307
manchester	england	161069
budapest	hungary	159956
madrid	spain	152145
seattle	washington	149076
milan	italy	141819
edinburgh	scotland	139725
toronto	canada	137868
dublin	ireland	134567
antwerp	belgium	133983
liverpool	england	131819
seoul	south korea	123072
ottawa	canada	119829
chicago	illinois	113187
melbourne	australia	103808
buenos aires	argentina	99898
philadelphia	pennsylvania	98514
montreal	canada	96764
hong kong	china	96182
helsinki	finland	92326
detroit	michigan	92004
saint petersburg	ruissia	86700
glasgow	scotland	84822
barcelona	spain	84791
warsaw	poland	82566
genoa	italy	77786
shanghai	china	77010
salt lake city	utah	72884
istanbul	turkey	72764
san francisco	california	72286
vienna	austria	70092
copenhagen	denmark	68238
houston	texas	65676
portland	oregon	64727
dubai	abu dhabi	61771
singapore	singapore	60545
lisbon	portugal	59941
vancouver	canada	58396
cologne	germany	57433
taipei	taiwan	55452
sydney	australia	54845
cleveland	ohio	51582
zurich	switzerland	50525
sao paulo	brazil	48924
duesseldorf	germany	48675
kyiv	ukraine	48392
rome	italy	47978
munich	germany	47102
athens	greece	45045
guangzhou	china	42170
mexico city	mexico	41617
southampton	england	40832
minsk	belarus	40548
bangkok	thailand	40333

sofia	bulgaria	40286
frankfurt	germany	38318
dresden	germany	35100
new hampshire	united states	34917
san jose	california	34203
tehran	iran	33351
turin	italy	31260
rio de janeiro	brazil	28103
venice	italy	26647
beijing	china	24945
marseille	france	23329
brasilia	brazil	22521
strasbourg	france	21781
austin	texas	20694
florence	italy	19125
bordeaux	france	18975
las vegas	nevada	18302
boston	massachusetts	17481
san diego	california	15679
baghdad	iraq	15414
shenzhen	china	15209
valencia	spain	14005
bologna	italy	13770
naples	italy	11333
cairo	egypt	10790
new orleans	louisiana	10667
tianjin	china	9580
chongqing	china	9506
louisville	kentucky	8252
doha	qatar	7894
wuhan	china	7230
jerusalem	israel	6889
belfast	ireland	5971
chengdu	china	5481
hangzhou	china	4084
albany	new york	3111
alexandria	egypt	1725

Table 2: OpenStreetMap City Statistics

ID	Type	Category	Group	Amount
1	ALPINEHUT	ACCOMMO	Res	723
2	CAMPING	ACCOMMO	Res	5072
3	CARAVAN	ACCOMMO	Res	1469
4	CHALET	ACCOMMO	Res	1622
5	HOSTEL	ACCOMMO	Res	4442
6	HOTEL	ACCOMMO	Res	51706
7	MOTEL	ACCOMMO	Res	3988
8	COURT	AMENITY	Pub	2512
9	FIRESTATION	AMENITY	Pub	13647
10	GOVERNMENT	AMENITY	Pub	546
11	LIBRARY	AMENITY	Pub	15826
12	PLAYGROUND	AMENITY	Pub	14914
13	POLICE	AMENITY	Pub	16434
14	POSTOFFICE	AMENITY	Pub	23095
15	PRISON	AMENITY	Pub	972
16	PUBLICBUILDING	AMENITY	Pub	28112
17	TOWNHALL	AMENITY	Pub	10811
18	BLOCKS	BARRIER	Pub	22792
19	COLLEGE	EDUCATION	Pub	6991
20	NURSERY	EDUCATION	Pub	38635
21	SCHOOL	EDUCATION	Pub	158307
22	UNIVERSITY	EDUCATION	Pub	22494
23	BAR	FOOD	Com	34921
24	BIERGARTEN	FOOD	Com	865
25	CAFE	FOOD	Com	93353
26	FASTFOOD	FOOD	Com	107439
27	ICECREAM	FOOD	Com	6018
28	PUB	FOOD	Com	40764
29	RESTAURANT	FOOD	Com	246174
30	DENTIST	HEALTH	Pub	18935
31	DOCTORS	HEALTH	Pub	20482
32	HOSPITALEMERGENCY	HEALTH	Pub	1067

33	HOSPITAL	HEALTH	Pub	32673
34	PHARMACY	HEALTH	Pub	52158
35	VETERINARY	HEALTH	Pub	7105
36	ALLOTMENTS	LANDUSE	Pub	9263
37	CONIFEROUSDECIDUOUS	LANDUSE	Pub	16455
38	CONIFEROUS	LANDUSE	Pub	37
39	DECIDUOUS	LANDUSE	Pub	137
40	GRASS	LANDUSE	Pub	145040
41	HILLS	LANDUSE	Pub	851
42	MILITARY	LANDUSE	Pub	2742
43	QUARY	LANDUSE	Pub	3379
44	SCRUB	LANDUSE	Pub	1344
45	SWAMP	LANDUSE	Pub	2737
46	BANK	MONEY	Com	64511
47	EXCHANGE	MONEY	Com	1887
48	BAHAI	POW	Pub	30
49	BUDDHIST	POW	Pub	6076
50	CHRISTIAN	POW	Pub	93480
51	HINDU	POW	Pub	322
52	ISLAMIC	POW	Pub	5965
53	JAIN	POW	Pub	9
54	JEWISH	POW	Pub	1759
55	SHINTO	POW	Pub	3711
56	SIKH	POW	Pub	175
57	APARTMENT	ACCOMMO	Res	1164557
58	UNKNOWN	POW	Pub	8691
59	CAVE	POI	Pub	1608
60	CRANE	POI	Pub	24844
61	EMBASSY	POI	Pub	4176
62	BUNKER	POI	Pub	1040
63	MINE	POI	Pub	128
64	PEAK1	POI	Pub	80
65	PEAK	POI	Pub	21728
66	CITY	POI	Pub	1171
67	HAMLET	POI	Pub	126600
68	SUBURB	POI	Pub	16462
69	TOWN	POI	Pub	9088
70	VILLAGE	POI	Pub	45771
71	TOWERCOMMUNICATION	POI	Pub	815
72	TOWERLOOKOUT	POI	Pub	4181
73	ALCOHOL	SHOP	Com	15279
74	BAKERY	SHOP	Com	30026
75	BICYCLE	SHOP	Com	24653
76	BOOK	SHOP	Com	10243
77	BUTCHER	SHOP	Com	10055
78	CARREPAIR	SHOP	Com	33964
79	CAR	SHOP	Com	19728
80	CLOTHES	SHOP	Com	72230
81	COMPUTER	SHOP	Com	4534
82	CONFECTIONERY	SHOP	Com	10692
83	CONVENIENCE	SHOP	Com	88780
84	COPYSHOP	SHOP	Com	9069
85	DEPARTMENTSTORE	SHOP	Com	157706
86	DIY	SHOP	Com	23016
87	FISH	SHOP	Com	1905
88	FLORIST	SHOP	Com	18949
89	GARDENCENTRE	SHOP	Com	3247
90	GIFT	SHOP	Com	8386
91	GREENGROCER	SHOP	Com	5909
92	HAIRDRESSER	SHOP	Com	72567
93	HEARINGAIDS	SHOP	Com	1209
94	HIFI	SHOP	Com	11077
95	JEWELRY	SHOP	Com	11695
96	KIOSK	SHOP	Com	7287
97	LAUNDRETTE	SHOP	Com	17201
98	MARKETPLACE	SHOP	Com	5330
99	PHONE	SHOP	Com	14333
100	MOTORCYCLE	SHOP	Com	10959
101	MUSIC	SHOP	Com	1643
102	NEWSPAPER	SHOP	Com	4228
103	PET	SHOP	Com	6967
104	SHOES	SHOP	Com	13501
105	SUPERMARKET	SHOP	Com	89951
106	TOBACCO	SHOP	Com	2994
107	TOYS	SHOP	Com	5416
108	VENDINGMACHINE	SHOP	Com	11433

109	VIDEORENTAL	SHOP	Com	888
110	ARCHERY	SPORT	Pub	297
111	BASEBALL	SPORT	Pub	3305
112	BASKETBALL	SPORT	Pub	2769
113	BOWLING	SPORT	Pub	1256
114	CANOE	SPORT	Pub	530
115	CRICKET	SPORT	Pub	894
116	DIVING	SPORT	Pub	23
117	FOOTBALL	SPORT	Pub	1281
118	GOLF	SPORT	Pub	12559
119	GYM	SPORT	Pub	937
120	GYMNASIUM	SPORT	Pub	680
121	CLIMBING	SPORT	Pub	1555
122	HORSE	SPORT	Pub	1643
123	ICESKATING	SPORT	Pub	981
124	LEISURECENTER	SPORT	Pub	19179
125	MINIATURGOLF	SPORT	Pub	399
126	MOTORRACING	SPORT	Pub	531
127	SHOOTING	SPORT	Pub	786
128	SKATING	SPORT	Pub	1148
129	SKIINGDOWNHILL	SPORT	Pub	206
130	SNOOKER	SPORT	Pub	624
131	SOCCER	SPORT	Pub	9011
132	STADIUM	SPORT	Pub	8513
133	SWIMMING	SPORT	Pub	5452
134	TENNIS	SPORT	Pub	5594
135	WATERSKI	SPORT	Pub	34
136	SURFING	SPORT	Pub	39
137	ARCHAEOLOGICAL	TOURIST	Pub	3825
138	ART	TOURIST	Pub	23703
139	ATTRACTION	TOURIST	Pub	19193
140	BATTLEFIELD	TOURIST	Pub	111
141	BEACH	TOURIST	Pub	6928
142	CASTLE	TOURIST	Pub	363
143	CASTLE2	TOURIST	Pub	3879
144	CINEMA	TOURIST	Pub	5410
145	FOUNTAIN	TOURIST	Pub	11737
146	INFORMATION	TOURIST	Pub	1967
147	MEMORIAL	TOURIST	Pub	31826
148	MONUMENT	TOURIST	Pub	6173
149	MUSEUM	TOURIST	Pub	12504
150	NIGHTCLUB	TOURIST	Pub	5735
151	RUINS	TOURIST	Pub	4208
152	THEATRE	TOURIST	Pub	24443
153	THEMEPARK	TOURIST	Pub	1032
154	RESIDENCE	ACCOMMO	Res	2506101
155	HOUSE	ACCOMMO	Res	8593475
156	WINDMILL	TOURIST	Pub	831
157	WRECK	TOURIST	Pub	100
158	ZOO	TOURIST	Pub	924
159	TERMINAL	TRANSPORT	Pub	732
160	AIRPORT	TRANSPORT	Pub	2480
161	BUSSTOP	TRANSPORT	Pub	741215
162	FUEL	TRANSPORT	Pub	46268
163	LIGHTHOUSE	TRANSPORT	Pub	639
164	MARINA	TRANSPORT	Pub	3114
165	RENTALCAR	TRANSPORT	Pub	10196
166	SUBWAY	TRANSPORT	Pub	21667
167	STATION	TRANSPORT	Pub	29628
168	TRAMSTOP	TRANSPORT	Pub	19329
169	DAM	WATER	Pub	4607
170	TOWER	WATER	Pub	976
171	WEIR	WATER	Pub	5598

Table 3: OpenStreetMap POI Statistics. Res=Residence, Pub=Public, Com=Commercial

For Yelp Challenge Dataset, see **Yelp Challenge Dataset** for details. We use the standard dataset without any modification.