Journal of Information and Data Management, 2024, 15:1, doi: 10.5753/jidm.2024.3209

Sentence-ITDL: Generating POI type Embeddings based on Variable Length Sentences

Salatiel Dantas Silva [**Universidade Federal de Campina Grande** | ***salatiel@copin.ufcg.edu.br***]
Claudio E. C. Campelo [**Universidade Federal de Campina Grande** | ***campelo@dsc.ufcg.edu.br***]
Maxwell Guimarães de Oliveira [**Universidade Federal de Campina Grande** | ***maxwell@computacao.ufcg.edu.br***]

Systems and Computing Department, Federal University of Campina Grande (UFCG), Av. Aprigio Veloso 882, Campina Grande, PB, 58.429-140, Brazil.

Received: 2 March 2023 • **Published:** 9 July 2024

Abstract Point of Interest (POI) types are one of the most researched aspects of urban data. Developing new methods capable of capturing the semantics and similarity of POI types enables the creation of computational mechanisms that may assist in many tasks, such as POI recommendation and Urban Planning. Several works have successfully modeled POI types considering POI co-occurrences in different spatial regions along with statistical models based on the Word2Vec technique from Natural Language Processing (NLP). In the state-of-the-art, binary relations between each POI in a region indicate the co-occurrences. The relations are used to generate a set of two-word sentences using the POI types. Such sentences feed a Word2Vec model that produces POI type embeddings. Although these works have presented good results, they do not consider the spatial distance among related POIs as a feature to represent POI types. In this context, we present the Sentence-ITDL, an approach based on Word2Vec variable length sentences that include such a distance to generate POI type embeddings, providing an improved POI type representation. Our approach uses the distance to generate Word2Vec variable-length sentences. We define ranges of distances mapped to word positions in a sentence. From the mapping, nearby will have their types mapped to close positions in the sentences.Word2Vec's architecture uses the word position in a sentence to adjust the training weights of each POI type. In this manner, POI type embeddings can incorporate the distance. Experiments based on similarity assessments between POI types revealed that our representation provides values close to human judgment.

Keywords: Points of Interest, Machine Learning, Similarity, Geo-Semantics, Vector Embeddings

1 Introduction

Points of Interest (POI) are specific geographic locations considered useful or interesting. For an entity to be classified as a POI, it must have the following attributes: i) a name; ii) a location indicated by geographic coordinates; iii) at least one type, which indicates the nature or service offered by the POI; iv) an identifier; and v) some contact information [Laplante, 2015]. The importance of Points of Interest (POIs) for human activities has given rise to several kinds of researchs concerning reasoning and representing their types (such as *restaurants* and *parks*). Developing new methods capable of capturing the semantics and similarity of POI types enables the creation of computational mechanisms that may assist in many tasks. In Urban Planning, POIs can be exploited to help in decision-making, through a better understanding of cities' structure and the function of their different regions (such as residential and commercial areas) [Zhai *et al.*, 2019; Hu and Han, 2019; Mou *et al.*, 2020]. In Recommender Systems, representations that capture POI type semantics and similarity can help predict potential POIs that users might be interested in [Liu *et al.*, 2022; Ding *et al.*, 2019; Zhang *et al.*, 2022]. In Geographic Information Retrieval (GIR), in the absence of a desired answer to a place query, similar POI types can satisfy the user's needs to some degree. Moreover, representations that capture the semantics and similarity of types beyond nomenclature can help design relevant solutions to a variety of GIR problems [Gao and Yan, 2018; Huang *et al.*, 2022].

Recent researches [Harispe *et al.*, 2015; Yan *et al.*, 2017; Yao *et al.*, 2017] proposed to capture the semantics and similarity of POI types considering spatial information that implicitly relates the types, such as POI co-occurrences in different spatial regions. For instance, *Gas Stations* and *Car Wash* can be considered similar because they frequently have similar contexts, i.e. they usually have similar vicinity (Figure 1.a and Figure 1.b illustrates this scenario). In contrast, *Educational institutions* and *Gas Stations* can be considered different, because they do not share any or only a few instances of POI neighborhood. By analyzing the co-occurrence pattern, one can depict the types of POIs that are considered similar if they appear in similar contexts.

In this regard, Place2Vec [Yan *et al.*, 2017] is one of the pioneers representing POI types using the co-occurrence of POIs in a region (vicinity) along with the Word2Vec model [Mikolov *et al.*, 2013]. In Word2Vec, words in a text are represented as real values' vectors (named word embeddings, vector embeddings, or embeddings) based on the co-occurrence of the words [Bengio *et al.*, 2003; Mikolov *et al.*, 2013]. Concerning POIs, Place2Vec [Yan *et al.*, 2017] applied this technique to extract and measure the relation of POI types in a region according to the relation between its

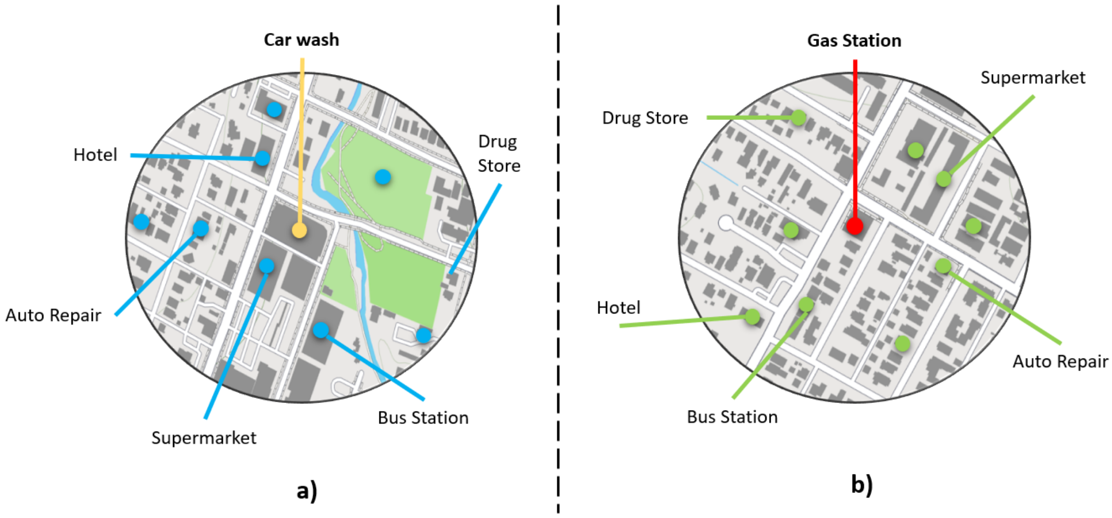

Figure 1. Context of POI co-occurrence in the vicinity of: a) Car Wash; b) Gas Station

vectors (called POI type embeddings). Thus, POI types with similar semantics will be closer in a vector space. These vectors can be used as input on clustering or recommendation algorithms, allowing tasks such as POI recommendation and POI type automated classification that may contribute to improving POI-based urban planning, business analysis, and so on.

Even though Place2Vec has presented good results, we must consider some limitations. To capture the semantics and similarity of POI types, two pieces of information were used to complement the co-occurrence: type uniqueness, based on the probability of a given type existing in a context, and popularity, inferred from the number of social media check-ins recorded in each POI. The intuition behind this method lies in the fact that different POI types characterize a context differently. For example, if there is a *shopping* and a *café* in a context, the *shopping* has more influence on the other POIs due to its popularity (it has more check-ins). Also, *shopping* is unique, occurring a few times in a neighborhood, while *café* may appear more frequently and not be as popular as *shopping*. Thus, considering popularity and uniqueness, increasing or decreasing the co-occurrence ratio of POI types in the Word2Vec training set is possible, allowing this method to generate more accurate vector embeddings for each POI type. However, such a method does not consider the distance among POIs as a feature to capture their similarity. The distance was only considered to check which POIs were in the vicinity. However, according to Tobler's first law of geography: "Everything is related to everything else, but near things are more related than distant things" [Tobler, 1970]. Thus, we agree that the closer the POIs are, the stronger the relationship among them. Another aspect worth mentioning is that the POI types have different levels arranged in a hierarchy[1]. Nonetheless, in Place2Vec, all POI types are related, with no distinction between different levels. Hence, types of a certain level can be directly related to types of other levels. This approach does not allow models to capture the semantics of types for each level separately, preventing comparisons across levels more accurately.

On this basis, this article presents the Sentence Information Theoretic Distance Lagged (Sentence-ITDL), an extended version of [Silva *et al.*, 2022], presented in XXIII Brazilian Symposium on GeoInformatics (GEOINFO 2022). Our method is an alternative to generate POI type embeddings that considers the distance between POIs in the vicinity to capture their similarity. Our approach uses the distance to generate Word2Vec variable-length sentences. Word2Vec's architecture uses the word position in a sentence to adjust the training weights of each POI type. In this manner, POI type embeddings can incorporate the distance. Furthermore, we generated POI type embeddings for the various levels of the hierarchy to understand how they impact the POI type semantics. The main contributions of this article are:

- A new method to represent POI types integrating the distance between POIs as a weight adjustment factor in the training of models. This enable Artificial Intelligence (AI) models to capture POI type semantics and produce more accurate representations;
- The vector embeddings of POI types that incorporate distance among the POIs. These embeddings can be used in several applications, such as recommendation algorithms, spatial searches, urban planning, geographic information retrieval, among others[2].

We organized the subsequent sections in this article as follows. Section 2 reports a range of related work developed in research focusing on the computational representation of POI types. After that, Section 3 presents our approach, exposing how we generated sentences and applied POI distance to update weights in the training of POI types. Following this, Section 4 provides an overview of the data used in the models' training. Afterward, Section 5 addresses the evaluation setup and experimental results, as well as comparisons between our models' and baseline results. Section 6 summarizes the results found, indicating future research possibilities.

[1] https://www.yelp.com/developers/documentation/v3/category_list

[2] Source code: https://github.com/SalatielDantas/S-ITDL.git

2 Related Work

This section discusses how related work represents POI types using computational mechanisms. The first subsection describes existing approaches that use POI vicinity to generate vector embeddings. The second subsection gives a discussion on other researches that use state-of-the-art NLP methods, such as transformers, to enrich the representation of POIs in general-purpose language models.

2.1 POI Type Embeddings Based on POI Vicinity

As an alternative to a computational representation of POI types, Yan *et al*. [2017] proposed the Information Theoretic, Distance Lagged (ITDL) algorithm to capture the POI types' semantics and, as a consequence, understand their relatedness and similarity. This method divides the continuous vicinity of a POI into multiple discrete bins (which are stripes of space around a POI). Figure 2 illustrates this configuration. Each bin represents a region used to learn a latent representation and encodes the semantic distribution between a center POI and its neighbors (context POIs). Two main pieces of information are considered to model the similarity and relatedness between the center POI Type and context POI Type: human activity (represented by the check-ins count of POIs in a bin) and the uniqueness of a place (represented by the probability of a POI appearing in a bin). They are combined to generate a number that augments the relation ⟨`center POI type, context POI type`⟩ in the train set.

Figure 2. An example of vicinity view from the discrete bins.

The algorithm uses binary relations between the center POI and the context POIs in each bin generating a list of tuples in the format ⟨`center POI type, context POI type`⟩. This set of tuples can be interpreted as a two-word statement set. Word2Vec estimates the probability of a context POI type occurring given the center POI type. Thus, the POI type co-occurrences with other POI types in a bin are used to represent them.

Similarly, Liu *et al*. [2019] applied Word2Vec to extract and represent a place's "niche" patterns. The developed models generate two main results: representations for the place's niche by type in a latent vectorial space, by which nearby vectors represent similar niches, and the occurrence probability of each POI type in the vicinity of a central POI. Liu *et al*. [2020] proposed a framework based on ITDL to envision and explore POIs, using dimensionality reduction techniques, such as t-SNE van der Maaten and Hinton [2008]. In this framework, the POI vectors are mapped to a two-dimensional space. Then, the configuration of POIs' regions is rendered by a thematic map that enables comparisons between different locations. Zhai *et al*. [2019] also proposed a framework based on ITDL to extract and identify the functions of urban regions. In this approach, the authors add the vectors of POIs in the vicinity to represent a single entity. Next, they applied the K-Means method to group similar entities. Furthermore, they enriched each entity by adopting vehicle and people traffic information to assist in tasks such as urban planning. Hu *et al*. [2020] also applied the idea of POIs' co-occurrence in a vicinity, using vector embedding multiprototype to encode the POI and topics that characterize it at once. After encoding, the vectors were clustered using the HDBSCAN algorithm McInnes *et al*. [2017] to generate POIs' regions of similar types and characteristics.

Wang *et al*. [2020] proposed Urban2Vec, a method similar to ITDL, but employing the MeanShift [Carreira-Perpinán, 2015] algorithm to define the edge of the neighborhood of POIs through clustering based on POI density. Thus, more concentrated areas will have smaller neighborhoods, and more dispersed areas (suburbs, rural areas) will have larger neighborhoods. As a result of the grouping, the vector embeddings of the types incorporate the multi-scalability of POIs in different environments. This method applies the co-occurrence and generates the vector embeddings from Word2Vec.

Chen *et al*. [2021] proposed the Hier-CEM (Category Embedding Model), which generates POI type vector embeddings incorporating the hierarchical structure of the types. Hier-CEM consists of two components: embeddings from a sequence of types in a neighborhood and embeddings from the hierarchy of types in the model. For each POI type in the sequence, it establishes connections using the types from the hierarchy. The constructed relationships generate vector embeddings via Word2Vec.

2.2 POI Embeddings Based on Transformers

Liu *et al*. [2021] proposed a pre-training model, called GeoBERT, to integrate semantics and geographic information into pre-trained representations of POIs. First, a graph simulates the distribution of POIs in the real world, where nodes represent POIs in a neighborhood connected based on their latitude and longitude. Were created some nodes to indicate the administrative level (street, neighborhood, city) and to connect POIs under the same properties. The graph is used as an input to train a graph-based model to integrate textual data with the vector embeddings of the POIs and trained on a BERT model [Devlin *et al*., 2018]. Li *et al*. [2022] proposed a language model that addresses spatial aspects to provide a general-purpose representation of geographic entities (POI) based on neighboring entities. The so-called SPABERT extends BERT [Devlin *et al*., 2018] to capture the linearized spatial context while preserving the spatial relationships of entities in two-dimensional space. This model is pre-trained as a masked language task and masked entity prediction task to learn spatial dependencies. The training used pseudo sentences from geographical databases derived from Open-

StreetMaps[3].

Based on the related work, we can say that the neighborhood information and check-in data are frequently applied strategies to represent POI types. This information defines the similarity of types from co-occurrences in the neighborhood context. In this case, check-ins capture geospatial semantics by replicating co-occurrences between types in a training set. The most used model is Word2Vec, which focuses on predicting context POI types based on center POI types. Recent methods based on transformers provide language models incorporating geographic information. These methods use the neighborhood of POIs to build pseudo-sentences to train a BERT model. Furthermore, the POIs were combined with textual data so that the language model captured the spatial information within the language. In this way, the resulting models present a general purpose aspect, being able to be used as a basis in several other activities.

Methods focused on generating POI type embeddings do not consider the distance between POIs in the context to capture their similarity. This article proposes a method to incorporate distance to generate variable sentences-length. The distance defines the position of each POI type in a sentence. The training weights of Word2Vec's architecture are different for each position in a sentence. Thus, we can incorporate the distance in the POI type embeddings using variable-length sentences.

3 Generating POI Type Embeddings

As mentioned in Subsection 2.1, The ITDL algorithm creates a set of tuples in the format ⟨`center POI type, context POI type`⟩ that can be used as a two-word phrase in Word2Vec's training. Figure 3 illustrates an example of training a set of tuples in Word2Vec's Skip-gram architecture. The first word in each tuple represents the center POI type. The second word represents the context POI type. The tuples processing is on the right side of Figure 3. In each instant $t_1, t_2, ..., t_n$, the weights (w') of each relation ⟨`center POI type, context POI type`⟩ are adjusted to estimate the probability of occurrence of the context POI type $(w(k+1))$, given the center POI type $(w(k))$.

Our approach follows a similar idea but includes the distance to generate variable-length sentences. The penalization of the POI types relationship is applied using Word2Vec's structure. As seen in Figure 6, it is possible to train the center word $(w(k))$ considering the words strictly next to it $(w(k+1), w(k-1))$ and the ones further away $(w(k+2)$, $(w(k-2))$. In Skip-Gram, an intrinsic penalty is applied on the weights as the words move away from the center word [Mikolov *et al.*, 2013]. The more distant the context word, the more penalized it will be. Based on this structure, we formulated the Sentence-ITDL algorithm that converts tuples into sentences with variable sizes, using the distance between POIs to define the types' position in the generated sentences (Section 3.1). Then, when we train the model, the position of each POI type in the sentence, now defined by the distance to the center POI, will affect the estimation of its types relation weight (w') (Section 3.2).

[3]www.https://openstreetmap.org

To make the conversion of tuples into sentences, we have made a change in the ITDL algorithm. The tuples generated are saved in the database along with the distance between the center POI and the context POI, resulting in a triple ⟨`center POI type, context POI type, distance`⟩. Once the build of the dataset is ready, it goes through another process: converting it into sentences with a pre-defined size. Then, we use the new dataset of sentences in the training process. Figure 4 illustrates the process followed by our approach.

The first step consists of executing the ITDL algorithm along with some POI dataset. The second step is the execution of Sentence-ITDL (Section 3.1), which transforms the triples into sentences. At the end of the process, the sentences are used to train a Word2Vec model using (Section 3.2).

3.1 Sentence-ITDL Algorithm

The Sentence-ITDL algorithm, presented in Algorithm 1, receives as input a set of triples (T), consisting of a center POI type $(tp1)$, a context POI type $(tp2)$, the distance (d) among these POIs, the distance from the center POI to the discrete bin (b) and the size of the sentences (s). In the algorithm, the triples set (T) is given to the *group_by_type function* (line 2) that groups the POIs types concerning the center POI types, ordered from the spatially closest to the farthest. To insert the POI types in a sentence, we fixed the first position to put the center POI type while the other positions of a sentence contain the context POI types. POIs are distributed in a continuous geographic space, and it is not possible to insert them directly into sentences that have a fixed size. For this, we define that each sentence index will be mapped to space intervals to cover the POIs. For that, we define a compartment size (c) that map each index to a range of spaces to the sentences index.

Algorithm 1: Sentence-ITDL Algorithm

```
   Input: T = (tp1, tp2, d), b, s
   Output: List of sentences
1  sentences ← empty_list()
2  G ← group_by_type(T)
3  c ← b/s
4  s ← s + 1
5  foreach g ∈ G do
6      i ← 0
7      while i < b do
8          K ← types_in_range(g, i, c)
9          foreach k ∈ K do
10             subsentence ← list_with_Sentinel()
11             subsentence[0] ← k['center_POI_type']
12             subsentence[i/c] ←
                 k['context_POI_type']
13             sentences.append(subsentence)
14         i ← i + 1
15 return sentences
```

To fill the sentences, we traverse distance intervals starting from the location of center POI and ending at b meters (the last bin distance), with the addition of c meters between each interval. The *types_in_range* function captures the POI types

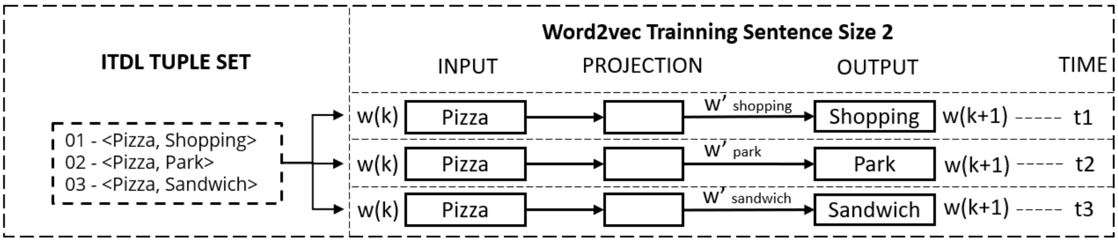

Figure 3. ITDL training example using Word2Vec's Skip-Gram architecture

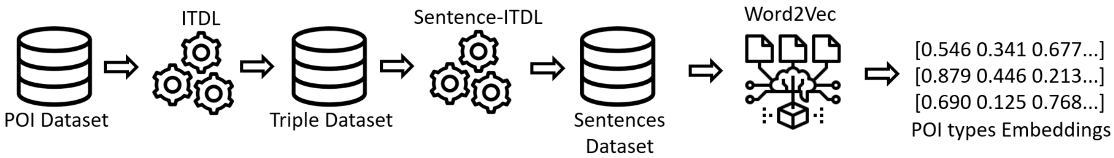

Figure 4. Process to generate embeddings with the ITDL and Sentence-ITDL algorithms

in the interval of distances greater than i and smaller than $i+c$. In other words, the function searches for POIs that fit in a range mapped to a specific sentence index. When identifying all POIs in the interval, the algorithm loops through one by one, creating sub-sentences. This sub-sentence is filled with a 'Sentinel' word, indicating that certain positions do not contain POI types. Then the center POI type is placed at the beginning of the sentence (index 0), and the context POI type at index (i/c). Finally, the new subsentence is added to the list of sentences. It is important to highlight that each subsentence consists only of a center POI type and a context POI type, aiming to maintain the binary relation generated by the ITDL. For instance, inserting a third POI type would modify such a relationship.

As an example, the left part of Figure 5 illustrates a situation in which the distance of the discrete bin (b) used is $100m$. The sentence size (s) is 2, indicating that the sentence will have two context words and a center word (index 0). The compartment size(c) is $50m$ (b/s). The center POI type will fill the first sentence index. The context POI types in $[0, 50]$ meters range will fill the second index. The context POI types in $[51, 100]$ meters range will fill the last index. Firstly, the triples 01 and 02 are converted for being in the interval $[0, 50]$. The last index is filled with 'Sentinel' words for these two sentences. We do not fill with another POI type in the empty indexes because we aim to keep the binary relation between the center POI and context POI individually. Then, we updated the range to $[51, 100]$ meters and converted the triple 03.

The right side of Figure 5 shows how training occurs in Word2Vec. The POIs closer to the center are less penalized than those further away. In the example, the *shopping* and *sandwich* types are trained at position $w(k+1)$, while the *park* type, which is more distant, is trained at position $w(k+2)$. Therefore, the weight of relationship $\langle \texttt{Pizza, Park} \rangle$ is more penalized than the other weights relationships.

During the training step, we set the Word2Vec to ignore the 'Sentinel' word, as its function only serves as a word that fills an empty space in the sentence. With this new approach, we can incorporate distance information among POIs to generate POI type embeddings capable of capturing the distance aspects.

3.2 Latent Representation

The Word2Vec, proposed by Mikolov *et al*. [2013], is a technique developed for NLP capable of generating predictive models from raw text. Word2Vec contains two architectures, as illustrated in Figure 6. In the Continuous Bag of Words (CBOW), given the preceding word sequence $w(k-1)$, $w(k-2)$, and the succeeding word sequence $w(k+1)$, $w(k+2)$, the model can calculate the probability of a word w_k appear. In the Skip-Gram architecture, the model must predict the context words (preceding and succeeding) from the center word (w_k). Word2Vec model uses a window with variable size to define the "context words". For example, if the window size is five, the context is the five words that precede a target word and the five that succeed.

In its architecture, Word2Vec has a hidden layer (projection) that learns each word's embedding, yielding a d-dimensional embedding space. Thus, one can use each word's hidden layer weights (vector) and apply vector space operations, such as cosine similarity [Wu *et al*., 2021]. This way, the similarity between different words can be calculated since words with similar co-occurring contexts will have similar embeddings.

In the POI scenario, the idea is similar. The model predicts a center POI type with a set of context POI types (from the vicinity) or vice-versa. For that, we can use the cross entropy to measure the difference between the learned probability and the true probability as:

$$D(\hat{y}, y) = -y_t log(\hat{y_t}) \tag{1}$$

In Equation 1, $\hat{y}$ represents learned probability distribution and y represents true probability distribution. $\hat{y}$ is the predicted probability of a context POI Type occurring given a center POI type (denoted by t), and y_t can is the actual probability of a context POI Type occurring given the center POI type. $\hat{y_t}$ can be defined as:

$$\hat{y_t} = P(t_1, t_2, t_3, ..., t_m \,|\, t_c) \tag{2}$$

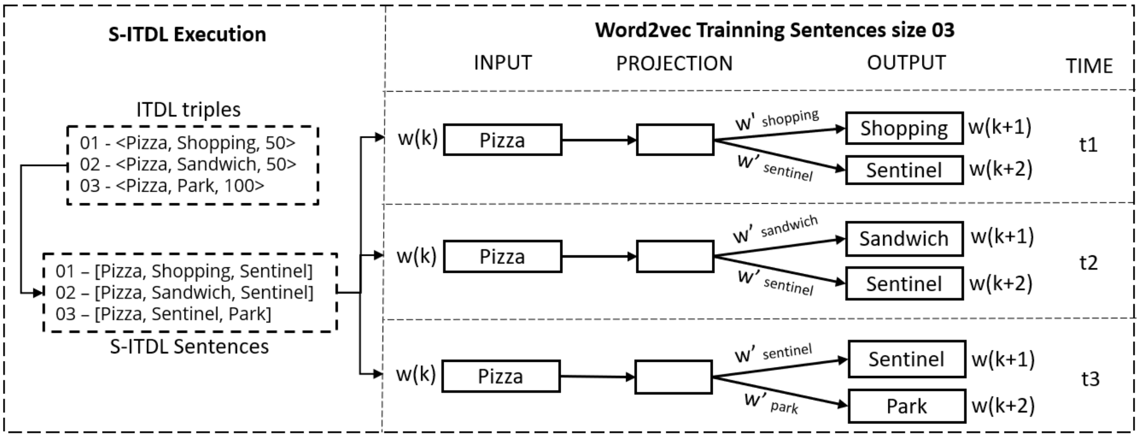

Figure 5. Example of Sentence-ITDL sentences training in the Word2Vec's Skip-Gram architecture

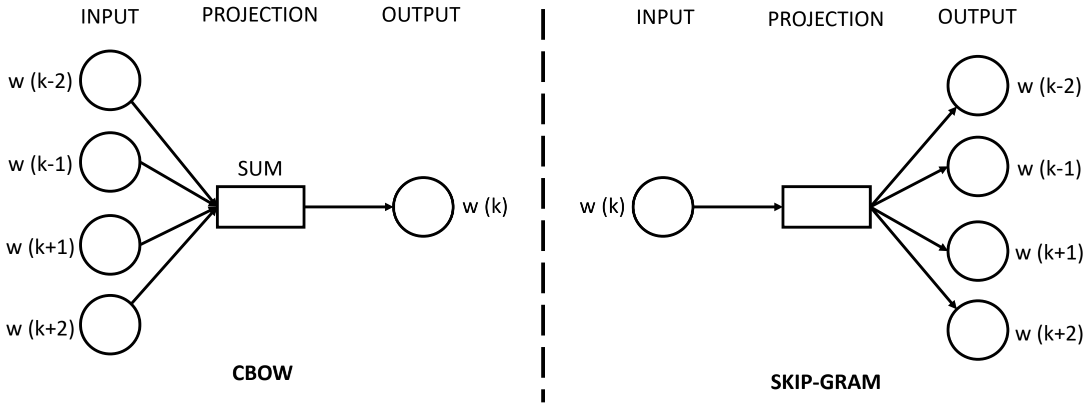

Figure 6. Word2Vec's Neural Network Architectures

where, $t_1, t_2, t_3, ..., t_m$ represent context POI types, and t_c represents the type of a POI that centers the context. In the output layer of the neural network, to transform the outputs into probabilities and replace the POI types with vector representations, the softmax function was used. Thus, the objective function is defined as follows:

$$Minimize\, J = -log \prod_{t=1}^{m} \frac{exp(u_t^T v_c)}{\sum_{k=1}^{|T|} exp(u_t^T v_c)} \quad (3)$$

where u_t and v_c are the context POI type vectors, and center POI type vectors, respectively; $|T|$ is the cardinality of a POI type. Thus, we can define a language model for POI types and use its vectors to calculate POI type similarity.

4 Dataset

The POIs used in this article come from the Yelp Challenge[4] database (February 2021 version). It comprises $160,585$ POIs, spread across 836 cities in the USA and Canada. Our experiments used the Austin region because it covers all POIs present in the evaluation set. Each POI is composed by 13 attributes, as shown in Table 1. In our experiments, we used: business_id, a unique identifier; latitude and longitude, indicating the exact location of the POI; categories, which indicate the POI types; and the check-in count that each POI presents, as this information is needed to run the Sentence-ITDL algorithms. Figure 7 illustrates the POI structure in the Yelp dataset.

Yelp has a taxonomy for the types of places, grouping them into 21 root types (level 01), 857 types on level 02, 404 types on level 03, and 17 types on level 04. Thus, we run our approach to generating POI type embeddings for each layer separately.

5 Experiments and Results

This section describes the experimental evaluation we conducted. Subsection 5.1 presents the set of assessments used to evaluate the vector embeddings generated in this approach; Subsection 5.2 discusses the parameter configuration needed to execute the method correctly. Subsection **??** presents and discusses the evaluations' results.

5.1 Evaluation Schema

To evaluate Sentence-ITDL, we chose the ITDL [Yan *et al.*, 2017] to compare because it is the pioneer and basis of other works related to POI type embeddings. In addition, among the related work, ITDL is the only one that generates embeddings focused on POI type, which is the focus of our approach, also providing the test set for comparison. In contrast, the other approaches create more extensive representations that do not focus only on the POI type and do not make

[4]https://www.yelp.com/dataset

Table 1. Attributes in Yelp Database

Attribute	Description
business_id	unique identifier for each POI
address	POI address
city	city containing the POI
state	state containing the POI
postal_code	postcode associated with POI location
latitude	geographic coordinates
longitude	geographic coordinates
stars	POI rating given by users
review_count	amount of comments related to POI
is_open	flag which indicates whether the POI is open or not
attributes	general information, such as number of accents, among others
categories	specifies the types associated with the POI
hours	POI opening hours

Figure 7. Yelp dataset example

the test set available for comparison.

In the evaluation, we used the test set[5] provided by [Yan *et al.*, 2017]. This set was built from two tasks with the collaboration of 25 volunteers. Consequently, this set contemplates the human judgment about the semantic relatedness and similarity of POI types. The first task, named Binary HIT Evaluation (BHE), consists of identifying the less similar POI Type, considering three different POI types disposed in a triple. For example, in a set composed of types: dentist, education, and orthodontist, volunteers should determine which type is most different according to their judgment. Implicitly, this task results in binary relationships between the most similar (the two types not chosen are the most similar). In this task were presented 80 triples, and for each one, the participants chose the least similar POI Type between them.

The second task, named Ranking-based HIT Evaluation (RHE), aimed to indicate the relationship of similarity between two different POI types. For example, given the bar and nightclub types, volunteers should select a value between 01 and 07 (the higher the number, the greater the similarity and vice versa). The higher the RHE value, the greater the similarity relationship. Based on these values, was calculated Spearman's correlation coefficient [Ramsey, 1989] for both people's and models' results. This coefficient indicates not only the strength but also the direction of the association that exists between two variables [Ramsey, 1989]. This experiment used 70 pairs of POI types. To complement this assessment, we used the Mean Square Error (MSE) calculation between the values given by the models and the values given by people to measure which ones best capture the intensity of the relationship between the POI types identified by the participants.

On the set of tests, were compared types of different levels. For instance, the pubs type (level 03) was a test example compared to the cupcake type (level 02). It is essential to highlight that we seek to evaluate the POI type embeddings for different levels of hierarchy to understand how each method captured the semantics at these levels. Therefore, for each line of the set, we defined the most general type, then, the types compared to the same level were generalized. This process resulted in the following datasets being separated by levels:

- BHE Level 01 (30 test triples);
- BHE Level 02 (42 test triples);
- RHE Level 01 (16 test tuple);
- RHE Level 02 (43 test tuple);
- RHE Level 03 (08 test tuple).

The new test datasets only had level types smaller than 03 since no type is greater than this level in the original set. Consequently, for BHE, our assessments will be at levels 01 and 02; for RHE, our tests will be at levels 01, 02, and 03.

5.2 Parameters Configuration

To run the experiments, we must define parameters related to Sentence-ITDL, Word2Vec, and ITDL. For Sentence-ITDL, defining the size of the sentences to be created (s) and the bin distance (b) to the bin used is essential. Regarding the ITDL parameters, it is necessary to determine the number of bins and the importance (w) of the uniqueness and popularity of each POI type. The value w varies in $[0, 1]$. When $w = 0$, we considered only the check-ins of the POIs. On the other hand, when $w = 1$, we considered only the probability of POI occurrence in a bin. If $w = 0.5$, the algorithm considers the two elements balanced.

In our experiments, we settled the importance to $w = 0.5$, for our goal is to analyze how sentences of different sizes constructed considering distance influence the resulting vector embeddings, no matter the popularity and uniqueness ratio. For the number of discrete bins and the dimension of the vectors, we used the same configuration defined by our baseline: 26 bins and 70 for vector dimension. For the sentence size, we define the sentence sizes $s = \{2, 3, 4, 5\}$ to analyze the results as the sentence size grows. For the Word2Vec window size, we used $window\_size = \{2, 3, 4, 5\}$ because this window needs to be the same size as the processed sentence. We used the Euclidean Distance metric to measure the distance among POIs. However, other distance metrics might be used.

[5] https://github.com/BoYanSTKO/place2vec

6 Conclusion and Further Work

Capturing the semantics of POI types by learning vector embeddings and using them to reason about places has already shown promising results in several works. In this article, we demonstrated that incorporating the distance among POIs in this learning process enables, for some hierarchy levels, a more precise differentiation of each type of POI. We also demonstrated that using the POI distance in a spatial region allows us to accurately distinguish the POI types. Furthermore, we show that using the distance also enables each POI type vector to present similarity values closer to the values of human judgment. Moreover, applying our technique requires little effort, using Word2Vec's natural structure, and not requiring many extra computational resources.

We have also revealed that separately considering the levels of types enables capturing the POI type's semantics without interfering among the levels. The results demonstrate that each level can present different configurations that influence the POI type semantics. Thus, using vector embeddings by hierarchy level allows one to perform more general or specific operations. Furthermore, we demonstrate that Sentence-ITDL performs better at levels with a greater variety of POI types, indicating that this method suits these situations. For levels with smaller types, the Sentence-ITDL still presents values of similarity closer to human judgment.

In the future, we intend to directly apply the distance value to penalize the co-occurrence relationship in Word2Vec's training set. In other words, instead of considering distance ranges mapped to indexes, we intend to use each distance to penalize the relationship between the POIs uniquely. In this way, different distances will have other influences on each POI.

Acknowledgments

This research had financial support granted by the Paraíba State Research Foundation (FAPESQ), grant #2021/3217.

References

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155.

Carreira-Perpinán, M. A. (2015). A review of mean-shift algorithms for clustering. *arXiv preprint arXiv:1503.00687*, 1(1):1–28.

Chen, M., Zhu, L., Xu, R., Liu, Y., Yu, X., and Yin, Y. (2021). Embedding hierarchical structures for venue category representation. *ACM Transactions on Information Systems (TOIS)*, 40(3):1–29.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 1(1):1–16.

Ding, J., Yu, G., Li, Y., Jin, D., and Gao, H. (2019). Learning from hometown and current city: Cross-city poi recommendation via interest drift and transfer learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(4):1–28.

Gao, S. and Yan, B. (2018). Place2vec: visualizing and reasoning about place type similarity and relatedness by learning context embeddings. In *Adjunct Proceedings of the 14th International Conference on Location Based Services*, pages 225–226, Rämistrasse 101, 8092 Zurich, Switzerland. ETH Zurich, ETH Zurich.

Harispe, S., Ranwez, S., Janaqi, S., and Montmain, J. (2015). Semantic similarity from natural language and ontology analysis. *Synthesis Lectures on Human Language Technologies*, 8(1):1–254.

Hu, S., He, Z., Wu, L., Yin, L., Xu, Y., and Cui, H. (2020). A framework for extracting urban functional regions based on multiprototype word embeddings using points-of-interest data. *Computers, Environment and Urban Systems*, 80:101442.

Hu, Y. and Han, Y. (2019). Identification of urban functional areas based on poi data: A case study of the guangzhou economic and technological development zone. *Sustainability*, 11(5):1385.

Huang, J., Wang, H., Sun, Y., Shi, Y., Huang, Z., Zhuo, A., and Feng, S. (2022). Ernie-geol: A geography-and-language pre-trained model and its applications in baidu maps. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3029–3039, Washington, DC, USA. SIGKDD.

Laplante, P. A. (2015). *Encyclopedia of Information Systems and Technology-Two Volume Set*. CRC Press, Taylor & Francis.

Li, Z., Kim, J., Chiang, Y.-Y., and Chen, M. (2022). Spabert: A pretrained language model from geographic data for geo-entity representation. *arXiv preprint arXiv:2210.12213*, 1(1):1–13.

Liu, K., Yin, L., Lu, F., and Mou, N. (2020). Visualizing and exploring poi configurations of urban regions on poi-type semantic space. *Cities*, 99:102610.

Liu, X., Andris, C., and Rahimi, S. (2019). Place niche and its regional variability: Measuring spatial context patterns for points of interest with representation learning. *Computers, Environment and Urban Systems*, 75:146–160.

Liu, X., Hu, J., Shen, Q., and Chen, H. (2021). Geo-BERT pre-training model for query rewriting in POI search. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2209–2214, Punta Cana, Dominican Republic. Association for Computational Linguistics. DOI: 10.18653/v1/2021.findings-emnlp.190.

Liu, Y., Yang, Z., Li, T., and Wu, D. (2022). A novel poi recommendation model based on joint spatiotemporal effects and four-way interaction. *Applied Intelligence*, 52(5):5310–5324.

McInnes, L., Healy, J., and Astels, S. (2017). hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In Bengio, Y. and LeCun, Y., editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop*

Track Proceedings. DBLP.

Mou, X., Cai, F., Zhang, X., Chen, J., and Zhu, R. (2020). Urban function identification based on poi and taxi trajectory data. In *Proceedings of the 3rd International Conference on Big Data Research*, ICBDR '19, page 152–156, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3372454.3372468.

Ramsey, P. H. (1989). Critical values for spearman's rank order correlation. *Journal of educational statistics*, 14(3):245–253.

Silva, S. D., Campelo, C. E. C., and de Oliveira, M. G. (2022). Generating POI type embeddings based on variableword2vec sentences. In Santos, L. B. L. and de Arruda Pereira, M., editors, *XXIII Brazilian Symposium on Geoinformatics - GEOINFO 2022, São José dos Campos, SP, Brazil, November 28 30, 2022*, pages 15–26. MCTIC/INPE.

Tobler, W. R. (1970). A computer movie simulating urban growth in the detroit region. *Economic geography*, 46(sup1):234–240.

van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Wang, Z., Li, H., and Rajagopal, R. (2020). Urban2vec: Incorporating street view imagery and pois for multi-modal urban neighborhood embedding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:1013–1020. DOI: 10.1609/aaai.v34i01.5450.

Wu, S. *et al*. (2021). *Design and Implementation of LBW–A Mental Health Application for Children*. PhD thesis, Graduate School of Vanderbilt University, Nashville, Tennessee.

Yan, B., Janowicz, K., Mai, G., and Gao, S. (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, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3139958.3140054.

Yao, Y., Li, X., Liu, X., Liu, P., Liang, Z., Zhang, J., and Mai, K. (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):825–848.

Zhai, W., Bai, X., Shi, Y., Han, Y., Peng, Z.-R., and Gu, C. (2019). Beyond word2vec: An approach for urban functional region extraction and identification by combining place2vec and pois. *Computers, Environment and Urban Systems*, 74:1–12.

Zhang, L., Sun, Z., Zhang, J., Wu, Y., and Xia, Y. (2022). Conversation-based adaptive relational translation method for next poi recommendation with uncertain check-ins. *IEEE Transactions on Neural Networks and Learning Systems*, 1(1):1–14. DOI: 10.1109/TNNLS.2022.3146443.