Incorporating Semantic Similarity with Geographic Correlation for Query-POI Relevance Learning

Ji Zhao,[1] Dan Peng,[1] Chuhan Wu,[1,2] Huan Chen,[1] Meiyu Yu,[1] Wanji Zheng,[1]
Li Ma,[1] Hua Chai,[1] Jieping Ye,[1] Xiaohu Qie[1]
[1]Didi Chuxing, Beijing, China
[2]Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, USA
{zhaojijet, pengdan, chenhuan, yumeiyu, zhengwanji}@didiglobal.com
{malimarey, chaihua, yejieping, tiger.qie}@didiglobal.com
cwdg9@mail.missouri.edu

Abstract

Point-of-interest (POI) retrieval that searches for relevant destination locations plays a significant role in on-demand ride-hailing services. Existing solutions to POI retrieval mainly retrieve and rank POIs based on their semantic similarity scores. Although intuitive, quantifying the relevance of a Query-POI pair by single-field semantic similarity is subject to inherent limitations. In this paper, we propose a novel Query-POI relevance model for effective POI retrieval for on-demand ride-hailing services. Different from existing relevance models, we capture and represent multi-field and local&global semantic features of a Query-POI pair to measure the semantic similarity. Besides, we observe a hidden correlation between origin-destination locations in ride-hailing scenarios, and propose two location embeddings to characterize the specific correlation. By incorporating the geographic correlation with the semantic similarity, our model achieves better performance in POI ranking. Experimental results on two real-world click-through datasets demonstrate the improvements of our model over state-of-the-art methods.

1 Introduction

Point-of-interest (POI) retrieval arises with the popularity of location based services, and has attracted considerable interest from both academic and industrial fields. Generally speaking, it searches a POI database to obtain a list of relevant destination locations when a user inputs a query. For online taxicab platforms like Didi, Lyft, and Uber, POI retrieval plays a significant role in providing on-demand ride-hailing services, since the retrieval results directly impact the success or failure of rides as well as long-term user experience. Besides, two particular conditions in ride-hailing scenarios further set extremely high standards for the quality of retrieval: (1) due to the limited sizes of smartphone screens, the mobile Apps can display only a few (e.g., ≤ 10) top ranked POIs, and the users capture much less (e.g., ≤ 3) POIs at first glance; and (2) unlike searching for food or hotels, ride-hailing users usually have definite destinations in mind, and rarely click other candidate POIs. Quickly and precisely retrieving the expected POI remains challenging.

There has been a large amount of literature on information retrieval, especially on Query-Document semantic matching. Traditional approaches conduct matching in term level (e.g., Okapi Best Matching (BM25) (**?**)) or latent space (e.g., Partial Least Square (PLS) (**?**)). The latent model bridges the semantic gap between words by reducing the matching dimensionality and correlating semantically similar terms, and thus outperforms the term level one. Along with the success of deep learning in speech recognition, computer vision, and natural language processing, studies on deep semantic matching models are also making great progress. For example, the Deep Structured Semantic Model (DSSM) (**?**) takes in bag of letter-trigrams of a query and a document, projects sentences to low-dimensional latent space, and measures the relevance as cosine similarity of their semantic vectors. Another example is the Convolutional Neural Network Architecture-I (ARC-I) (**?**), which uses a convolutional-pooling structure to extract both local word-level features and global sentence-level features for matching. Compared to the traditional approaches, these deep models are able to represent deep semantic structures of the queries and the documents.

While current relevance models mainly judge documents based on the semantic similarity scores, they are subject to inherent limitations due to the simple representation of relevance, and thus cannot handle the increasing complexity of Query-POI matching puzzle in POI retrieval.

Specifically, with the global deployment of the ride-hailing services, the POI retrieval solutions are facing more and more complex queries. Here, the textual complexity comes from: (1) incomplete queries and queries of variable lengths; (2) mixed keyboard inputs (e.g., English+Spanish, English+Chinese, Chinese Characters+Pinyin Alphabets); and (3) various compositions and orders of the terms in the query. A robust Query-POI relevance model has the ability to manage these cases.

In this paper, we aim to design a robust Query-POI relevance model for effective POI retrieval for on-demand ride-hailing services. Different from the existing models that quantify the relevance of a Query-POI pair by single-field semantic similarity, we capture and represent multi-field and local&global semantic features to measure the semantic similarity. Besides, ride-hailing customers are highly sensitive to the distances between origin-destination locations, especially when they search for chain stores like McDonald's and Starbucks. Through large-scale data analysis, we observe a hidden geographic correlation of the clicked Query-POI pairs[1]. As Figure 1 shows, over 50% origin-destination pairs are located within 4 km, which provides extra information for Query-POI ranking. For a Query-POI pair, we learn the geographic correlation, and integrate it with the semantic similarity for relevance analysis. Enriched with comprehensive information, our model is able to handle the increasingly complex queries.

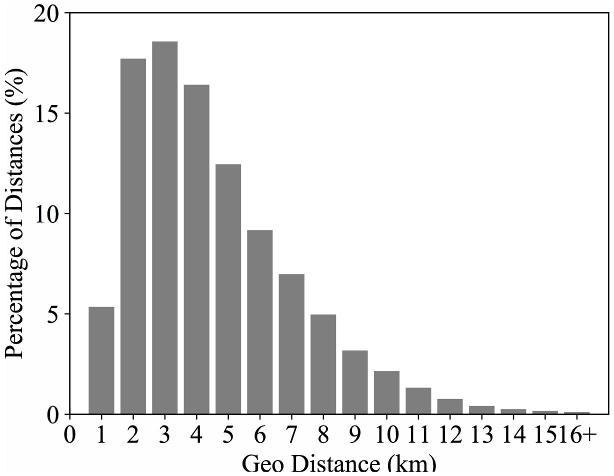


Figure 1: Percentages of distances between origin-destination pairs in a large-scale click-through dataset generated by ride-hailing services. Over 50% of the clicked origin-destination pairs are located within 4 km, showing potential for Query-POI geo-relevance learning.

Overall, the main contributions of our work are:

- We propose a novel Query-POI relevance model for effective POI retrieval for on-demand ride-hailing services.
- We exploit self-attention to draw intra-dependency within single-field texts, and adopt interactive attention to draw inter-dependency between multi-field textual attributes of POIs, to highlight the keywords of queries and POIs prior to semantic matching. The learned multi-field and local&global semantic features help us to handle the increasingly complex queries.
- We boost its overall performance by incorporating geographic correlation with semantic similarity in the relevance model. We observe a specific hidden correlation between origin-destination locations in ride-hailing scenarios, and propose two location embedding methods to characterize the geo-correlation. Additionally, we visualize the embeddings to give a better interpretation.
- We conduct extensive experiments on two real-world large-scale click-through datasets, and evaluate the functionality of each main module step by step. Experimental results show significant improvements in Query-POI relevance over state-of-the-art models.

2 Related Work

2.1 Neural-network Based Semantic Models

Deep neural networks have shown the effectiveness in discovering hierarchical features from raw training data for various tasks (**?**; **?**; **?**; **?**; **?**; **?**; **?**; **?**). Among them, the DSSM (**?**) and the ARC-I (**?**) are the most related to our work. The DSSM uses a deep neural network to map the raw bag-of-words term vectors of search queries and Web documents into semantic vectors. The relevance score of a pair of query and document is the cosine similarity of their corresponding semantic vectors. However, bag-of-words representations cannot keep the contextual structure within the query or documents. In contrast, the ARC-I captures both word level and sentence level contextual structures. It uses pre-trained word embeddings to present the sentences, and then takes multiple convolutional and max-pooling layers to capture the global features for matching. A multi-layer perceptron is used to calculate the matching degree of the two sentences. One drawback of these models is that they only capture simple representations and structures of queries and documents, and are at the risk of losing important information for relevance analysis.

2.2 Attention Mechanisms

In recent years, attention mechanisms have been considered for sequential tasks (**?**; **?**; **?**). In the paper (**?**), the authors proposed a transformer model, which is composed of an encoder and a decoder. In each component, multi-head attention is used to jointly attend to information from different representation subspaces at different positions. Seo *et al.* (**?**) proposed a bi-directional attention flow (BIDAF) network for machine comprehension. In this network, attention is used in two directions: from context to query and from query to context. The bi-directional attention flow mechanism acquires a query-aware context representation which can reduce errors caused by early summarization.

2.3 Spatial Representation

Spatial information is any information related to a specific location. Zhang *et al.* (**?**) proposed a deep learning based model, named ST-ResNet, which can predict citywide crowd flows. In ST-ResNet, the crowd flow and the distance of nearby region are used to model spatial dependency. And in (**?**), spatial information is reflected in people's GPS trajectories. A RNN is used to deal with these historical GPS trajectories. The proposed model DeepTransport can simulate and predict human mobility and transportation mode. Li *et al.* (**?**) used a road network to estimate travel time. Besides, DeepWalk (**?**) and Line (**?**) study how to embed large information networks into low-dimensional vector spaces.

[1] For POI retrieval in location based services, a query generally consists of a user's explicit textual input and the user's implicit geographic information (which is automatically extracted by the mobile Apps to locate the user and provide corresponding services).

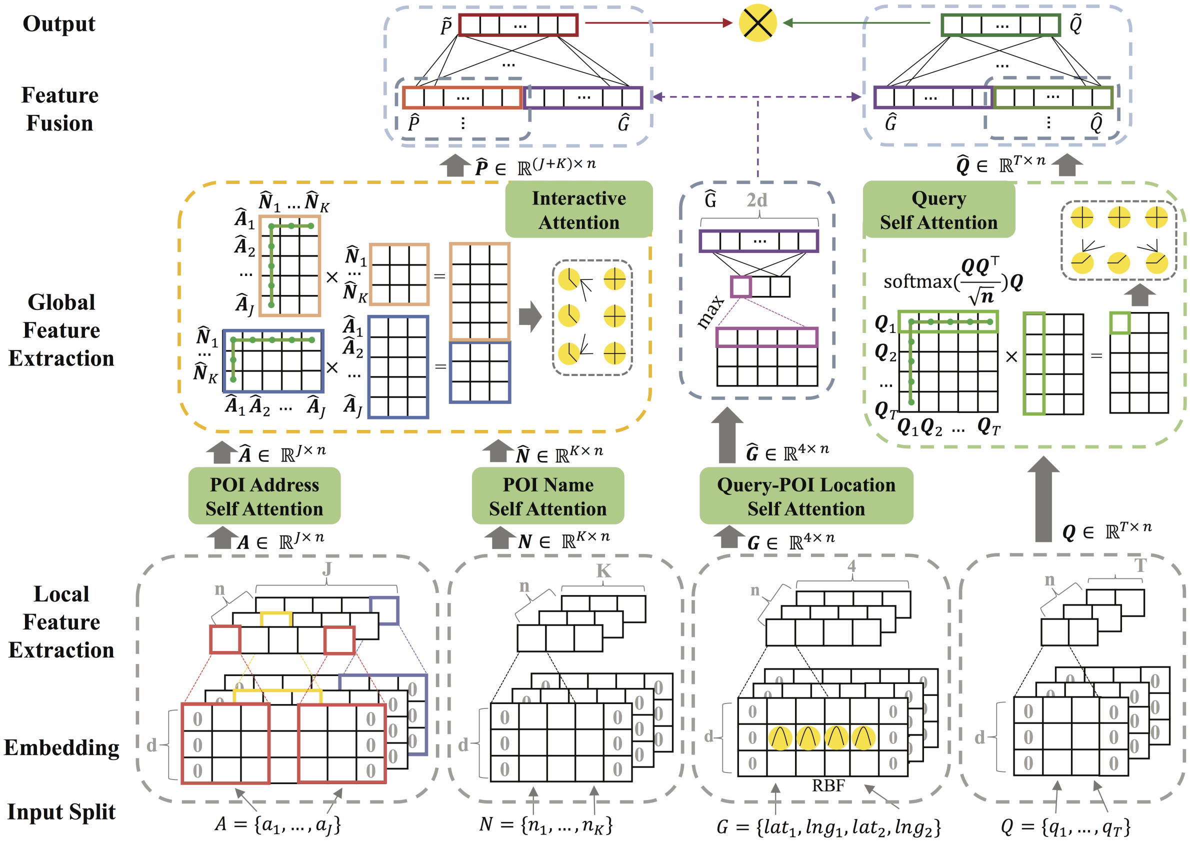

Output
Feature Fusion
Global Feature Extraction
Local Feature Extraction
Embedding
Input Split
$\tilde{P}$
$\tilde{Q}$
$\hat{P}$
$\hat{G}$
$\hat{Q}$
$\hat{\boldsymbol{P}} \in \mathbb{R}^{(J+K)\times n}$
$\hat{\boldsymbol{Q}} \in \mathbb{R}^{T\times n}$
Interactive Attention
Query Self Attention
$\mathrm{softmax}(\frac{QQ^\top}{\sqrt{n}})Q$
2d
max
$\hat{\boldsymbol{A}} \in \mathbb{R}^{J\times n}$
$\hat{\boldsymbol{N}} \in \mathbb{R}^{K\times n}$
$\hat{\boldsymbol{G}} \in \mathbb{R}^{4\times n}$
POI Address Self Attention
POI Name Self Attention
Query-POI Location Self Attention
$\boldsymbol{A} \in \mathbb{R}^{J\times n}$
$\boldsymbol{N} \in \mathbb{R}^{K\times n}$
$\boldsymbol{G} \in \mathbb{R}^{4\times n}$
$\boldsymbol{Q} \in \mathbb{R}^{T\times n}$
J
K
4
T
n
d
RBF
$A = \{a_1, \ldots, a_J\}$
$N = \{n_1, \ldots, n_K\}$
$G = \{lat_1, lng_1, lat_2, lng_2\}$
$Q = \{q_1, \ldots, q_T\}$

Figure 2: Illustration of the Query-POI relevance model. It has six layers. For a Query-POI pair, the model first splits their texts and locations into multiple attributes. After embeddings of letter&word and latitude&longitude, it captures the semantic similarity and geographic correlation by convolutional neural networks with attention mechanisms. Then, it learns their comprehensive feature vectors, and finally outputs their relevance as cosine similarity score.

3 The Query-POI Relevance Model

3.1 Model Overview

As Figure 2 shows, our Query-POI relevance model is hierarchical and consists of six layers:

- **Input Splitting Layer** splits the texts into words and letters, and divides the geographic coordinates into latitudes and longitudes. The textual attributes of POI include POI Address and POI Name.
- **Embedding Layer** embeds each word, letter, and coordinate to a low-dimensional dense vector. Embeddings are randomly initiated and then trained with convolutional neural networks (CNNs).
- **Local Feature Extraction Layer** uses CNNs to separately capture the local semantic and the geographic features of the input Query-POI pair.
- **Global Feature Extraction Layer** applies attention mechanisms to the local features to enrich them with global information, and uses max pooling to extract the salient global features. Specifically, we learn the intra-dependency within texts and locations by self-attention, and the inter-dependency between multiple POI attributes by interactive attention.
- **Feature Fusion Layer** integrates the semantic and the geographic feature vectors to represent the comprehensive features of Query and POI.
- **Output Layer** calculates the cosine similarity score of integrated semantic similarity and geographic correlation for the Query-POI pair.

3.2 Semantic Representation

We use CNNs with attention mechanisms to learn the low-dimensional dense embeddings of letters and words, as well as the semantic feature vectors of queries and POIs.

Letter&Word Embeddings We first split the texts into words and letters, and then use both letter embedding and word embedding to reduce the dimensionality of the text representations. Here, letter granularity representation keeps the critical information of incomplete and/or mixed texts, and word level representation captures more contextual information for sophisticated languages like Chinese.

While one-hot encoding is high-dimensional and cannot capture the semantic meanings of words, the low-dimensional word2vec embedding (?) heavily depends on the text corpus and needs to be pre-trained for semantic analysis. Different from these two approaches, our letter&word embedding vectors are initiated with random real numbers, and trained along with the CNNs under the guidance of the loss function. These directly learned embedding vectors capture more information for later semantic matching.

Both embedding vectors are d-dimensional and stored in an embedding matrix. For a letter&word vocabulary of size V, the embedding matrix is $\mathbf{M} \in \mathbb{R}^{V \times d}$. Let $Q = \{q_1, \ldots, q_T\}$, $A = \{a_1, \ldots, a_J\}$, and $N = \{n_1, \ldots, n_K\}$ be the split strings of Query, POI Address, and POI Name, respectively. At the end of training, the embedding matrix $\mathbf{M}$ will contain all of the embeddings for each word and letter in the vocabulary. Embeddings for rare words and letters are set by default. Three sub-matrices of sizes $T \times d$, $J \times d$, and $K \times d$ contain the embedding vectors of the input text strings, respectively.

Convolutional Networks We feed the three parts of embedding vectors to three 3-layer CNNs to independently extract their local semantic features. Each CNN has n kernels of size $3 \times d$. After applying a zero padding to the feature matrices, convolving the kernels with the Query embedding vectors, and concatenating the n feature vectors of size $T \times 1$ by column, the CNN outputs the semantic feature matrix $\mathbf{Q} \in \mathbb{R}^{T \times n}$ of Query. Correspondingly, the other two CNNs output the local semantic features of POI Address $\mathbf{A} \in \mathbb{R}^{J \times n}$ and POI Name $\mathbf{N} \in \mathbb{R}^{K \times n}$.

Self-Attention Attention mechanisms aim to distinguish and catch the most important information from the whole. Formally, given a Query and a set of Key-Value pairs, the Attention is the sum of values weighted by the similarity of the query and corresponding keys. We apply the self-attention mechanism (?) to the semantic feature matrices to learn the intra-dependency within the texts, which suggests more latent syntactic and semantic features.

In our model, the self-attention of Query is calculated as

$$\mathbf{X}_q = self\_attention(\mathbf{Q}) = softmax(\frac{\mathbf{Q}\mathbf{Q}^\top}{\sqrt{n}})\mathbf{Q}.$$

Then the semantic matrix is fine-tuned by a position-wise feed-forward network with two linear transformations and a rectified linear unit (ReLU), i.e.,

$$\widehat{\mathbf{Q}} = max(0, \mathbf{X}_q \times \mathbf{W}_{q1} + \mathbf{B}_{q1}) \times \mathbf{W}_{q2} + \mathbf{B}_{q2}.$$

Here, $\mathbf{W}_{q1}, \mathbf{W}_{q2} \in \mathbb{R}^{n \times n}$ are weight matrices, and $\mathbf{B}_{q1}, \mathbf{B}_{q2} \in \mathbb{R}^{T \times n}$ are bias matrices. We also update the POI semantic feature matrices as $\widehat{\mathbf{A}} \in \mathbb{R}^{J \times n}$ and $\widehat{\mathbf{N}} \in \mathbb{R}^{K \times n}$.

Interactive Attention While self-attention shows the ability to extract the key information from the texts, interactive attention is able to link and fusion information from different sources. We apply interactive attention to the two semantic feature matrices of POI to learn the inter-dependency between the two complementary attributes. The inter-dependency highlights the mapping between POI Name and POI Address, and can be regarded as a strong support for multi-field searching and matching.

By tuning the feature vectors with inter-dependency, we learn more sophisticated semantic features $\widehat{\mathbf{A2N}} \in \mathbb{R}^{K \times n}$ and $\widehat{\mathbf{N2A}} \in \mathbb{R}^{J \times n}$, and concatenate the two matrices by row to get the global semantic feature matrix $\widehat{\mathbf{P}} \in \mathbb{R}^{(J+K) \times n}$:

$$\widehat{\mathbf{A2N}} = softmax(\frac{\widehat{\mathbf{N}}\widehat{\mathbf{A}}^\top}{\sqrt{n}})\widehat{\mathbf{A}}, \quad \widehat{\mathbf{N2A}} = softmax(\frac{\widehat{\mathbf{A}}\widehat{\mathbf{N}}^\top}{\sqrt{n}})\widehat{\mathbf{N}},$$

$$\widehat{\mathbf{P}} = max(0, [\widehat{\mathbf{A2N}}; \widehat{\mathbf{N2A}}] \times \mathbf{W}_{p1} + \mathbf{B}_{p1}) \times \mathbf{W}_{p2} + \mathbf{B}_{p2}.$$

Here, $\mathbf{W}_{p1}, \mathbf{W}_{p2} \in \mathbb{R}^{n \times n}$ are weight matrices, and $\mathbf{B}_{p1}, \mathbf{B}_{p2} \in \mathbb{R}^{(J+K) \times n}$ are bias matrices.

Semantic Feature Vectors The above matrices collect all of the semantic feature vectors for each word and letter, from which we take and concatenate the maximal value of each column as feature representatives. Through a fully-connected network, we finally get the semantic feature vectors $\widehat{Q}, \widehat{P} \in \mathbb{R}^{1 \times 2d}$ for further semantic analysis.

3.3 Geo-Correlation Extraction

Location Embeddings While the geographic correlation between the origin-destination pairs in Figure 1 shows potential for geo-relevance learning, it is hidden and cannot be directly described by raw location information. Like letter&word embeddings to texts, we embed location to get the dense vector representations for feature extraction.

Before that, the digital map that covers all the POIs is divided into $L_{at} \times L_{ng}$ grids of size $100m \times 100m$. Each grid, indexed by $\{(i,j) | i \in [0, L_{at}-1], j \in [0, L_{ng}-1]\}$, contains a number of POIs. To reduce the size of embedding matrix, all of the POIs in a grid share the geographic coordinate (i.e., latitude-longitude pair) of the point at the top-left corner of the gird. The area of a grid is empirically set as $100m \times 100m$ to get discretized intervals with little information loss. Overall, there are $L_{at} \times L_{ng}$ coordinates. We split the coordinates into latitudes and longitudes and separately embed them to further reduce the size of the embedding matrix (i.e., from $(L_{at} \times L_{ng}) \times d$ to $(L_{at} + L_{ng}) \times d$). Besides, this split also shows advantages of training when the locations in each grid are sparse. Let $G = \{lat_1, lng_1, lat_2, lng_2\}$ be the raw split coordinates of Query and POI, and $\mathbf{\Phi} \in \mathbb{R}^{L_{at} \times d}$ and $\mathbf{\Psi} \in \mathbb{R}^{L_{ng} \times d}$ be the embedding matrices.

Embeddings tested in our model include:

(1) One-hot Vectors: Each latitude interval and longitude interval is encoded with an one-hot vector. These high-dimensional, sparse vectors failed to capture the geographic correlation in our experiments.

(2) Coordinate Embeddings: These embedding vectors are low-dimensional and dense. For each origin-destination pair, their 4 d-dimensional latitude and longitude vectors are randomly initiated and trained through a 3-layer CNN. The embeddings after training would preserve the geographic

correlation, since the CNN directly takes in the Query-POI pairs and targets the goal of matching.

(3) Kernel Embeddings: The faithfulness of the above embeddings may be degraded due to the Boundary Effect. For a POI located in the boundary of a grid, its nearest neighbor is probably from a neighboring grid, rather than in its grid. To mitigate the effect, we fine-tune the coordinate embeddings to be the sum of weighted embeddings of itself and its neighboring embeddings. For a POI in grid g_{ij} with latitude vector Φ_i and longitude vector Ψ_j, its latitude embedding is tuned to be the sum of the 3 closest embeddings:

$$\widehat{\Phi}_i = w_{i-1}\Phi_{i-1} + w_i\Phi_i + w_{i+1}\Phi_{i+1}.$$

It is the same for longitude embedding. Here, Φ_{i-1} and Φ_{i+1} are neighboring latitude embeddings, and the weights are calculated through RBF kernel tuning over the distances between the POI and the centers of the grids.

Geographic Feature Vector Similar to the process of semantic feature extraction, we feed the 4 (or 12 with Kernel Embeddings) separate embedding vectors of the Query-POI pair to the 3-layer CNN to get the geographic feature matrix $\widehat{\mathbf{G}} \in \mathbb{R}^{4\times n}$. After that, we learn the geographic feature vector $\widehat{G} \in \mathbb{R}^{1\times 2d}$ for further relevance measure. For each pair of Query-POI, there is only one geographic feature vector which has absorbed the geographic correlation.

Sem&Geo Vectors We concatenate the semantic and the geographic feature vectors of Query and POI by column, and feed them into a fully connected layer to get the comprehensive feature vectors $\widetilde{Q}, \widetilde{P} \in \mathbb{R}^{1\times 2d}$, i.e.,

$$\widetilde{P} = [\widehat{P}, \widehat{G}] \times \mathbf{W}_p + B_p,$$

$$\widetilde{Q} = [\widehat{Q}, \widehat{G}] \times \mathbf{W}_q + B_q.$$

Here, $\mathbf{W}_p, \mathbf{W}_q \in \mathbb{R}^{2d\times d}$ are weight matrices, and $B_p, B_q \in \mathbb{R}^{1\times d}$ are bias vectors.

Relevance Score and Loss Function We follow the relevance function and the loss function of DSSM. The relevance is calculated as cosine similarity of the feature vectors. What makes our model different is that it considers not only the semantic similarity but also the geographic correlation of the Query-POI pair, and it has enriched the semantic features with intra-dependency and inter-dependency before matching. The relevance score is:

$$R(P,Q) = cos(\widetilde{P}, \widetilde{Q}) = \frac{\widetilde{P}\widetilde{Q}^\top}{\|\widetilde{P}\| \times \|\widetilde{Q}\|}.$$

During training, for each Query Q, the positive (most relevant) POI P^+ is determined by real-world click-through data, while the 4 negative POIs $\{P_j^-\}$ are randomly selected from unclicked POIs. Given a Query, the probability of a POI to be clicked is calculated by a softmax function of its relevance score with a smoothing factor γ, i.e.,

$$Pr(P|Q) = \frac{exp(\gamma R(P,Q))}{\sum_{P'\in\{P^+, P_j^-\}} exp(\gamma R(P',Q))}.$$

We set the loss function as

$$L(\Lambda) = -\log \prod_{Q,P^+} Pr(P^+|Q).$$

The model parameters Λ are trained to minimize the loss, or equivalently, to maximize the likelihood of the clicked pairs, using AdaGrad optimizer.

4 Experiments

4.1 Experimental Setup

Datasets We process two large-scale datasets of real-world ride-hailing orders. The data are collected by Didi Chuxing, and have been desensitized due to privacy issues. The basic statistics of the datasets are shown in Table **??**.

(1) Dataset A: This dataset collects the ride-hailing orders in Chengdu, China. As part of the GAIA Initiative[2], it is publicly available to the academic community[3]. We randomly select 115724 and 15261 queries out of one month records as training data and test data, respectively. For each query, the App displays a list of possibly relevant POIs for users. In statistics, there are 711824 POIs for training and 95369 for testing. The corresponding average numbers of recalls are 6.15 and 6.24. On average, Query, POI Name, and POI Address contain 4.6, 9.4, and 19 letters, respectively. Notably, letters in Chinese datasets include Chinese characters and Pinyin letters, while words include Chinese phrases and Pinyin phrase. The average physical distance of the clicked Query-POI pairs is about 4 km.

(2) Dataset B: Orders are generated nationwide and in a much larger scale. There are 1589392 queries and 6854458 POIs in total. In this dataset, the input texts are more diverse and more complex, and the driving distances are longer and more difficult to predict. We evaluate our model on this dataset to verify its generalization and robustness.

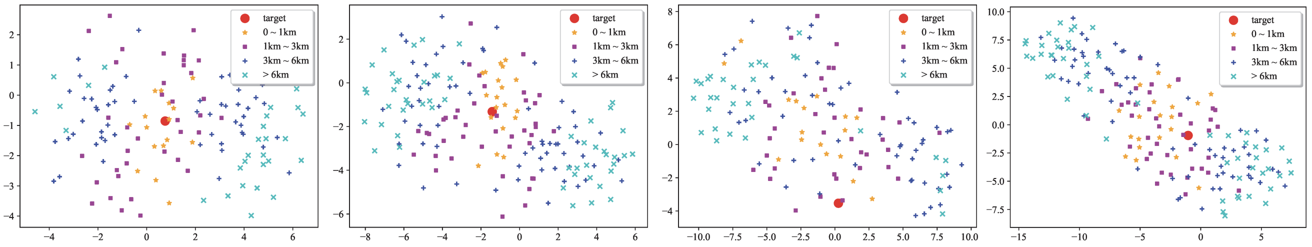
Table 1: The basic statistics of the datasets.

	Dataset A (Chengdu)		Dataset B (Nationwide)	
	Training	Testing	Training	Testing
Total Num of Query	115,724	15,261	1,476,645	112,747
Total Num of POIs	711,824	95,369	12,654,847	947,878
Avg Num of Recalls	6.15	6.24	8.57	8.41
Avg Len of Query	4.61	4.65	3.21	3.24
Avg Len of POI Addr	19.04	19.05	18.93	18.89
Avg Len of POI Name	9.38	9.42	7.87	7.96
Avg Distance (km)	4.05	4.04	7.28	8.53

During training, positive samples are the clicked Query-POI pairs, and negative samples are randomly generated according to POIs' proportions. In testing, for each query, the model calculates the similarity score of each POI in the recalled list. The clicked one is marked as positive, while all the others are negative.

[2] https://gaia.didichuxing.com

[3] https://outreach.didichuxing.com/appEn-vue/POI?id=11



(a) PALM lat embedding.

(b) PALM lng embedding.

(c) DeepWalk lat embedding.

(d) DeepWalk lng embedding.

Figure 3: Two-dimensional t-SNE projection of the d-dimensional geographic vectors of POIs. The randomly selected POIs around the central POI are within different physical distances, thus are classified with different colors. The location embeddings generated by PALM (a)-(b) preserve the physical distances in the embedding space. For the same POIs, the embeddings generated by DeepWalk (c)-(d) preserve the physical relation to some extent, but also show large chaos in the embeddings.

Evaluation Metrics Given a Query, the retrieval system recalls a list of POIs and runs the relevance model to rank the POIs according to their similarity scores. We evaluate the performance of all the relevance models by averaged Normalized Discounted Cumulative Gain (NDCG) (**?**) of the displayed POI list at truncation levels 3 and 10. As we have mentioned before, display positions have more significant impact on user satisfaction in mobile applications than that of the traditional retrieval systems. Considering both relevance and position of each POI in the result list, the NDCG score is a widely used offline measure of ranking quality. POI lists with higher NDCG scores are more likely to meet the users' need.

Compared Models We compare the proposed relevance model with two state-of-the-art models to show the effectiveness of our model. To evaluate the functionalities of the main modules, we compare the basic model with additional modules step by step. Models evaluated include:

(1) DSSM: This model uses a 3-layer deep neural network to learn the semantic structures in queries and POIs. The first hidden layer takes in bag-of-words term vectors and hashes the words into letter n-gram vectors for semantic feature learning. After multi-layer projection, the network outputs the corresponding semantic vectors and then calculates their cosine similarity. It considers only basic semantic similarity.

(2) ARC-I: This model uses word embedding pre-trained with word2vec to represent the texts, and then uses a convolutional neural network to learn the semantic features, and finally compares the feature vectors with a multi-layer perceptron. Same as DSSM, ARC-I measures relevance by simple semantic similarity.

(3) DPAM: This is our basic Deep Attention Model. It first splits the texts into words and letters to cover more diverse and more complex queries. Both word&letter embeddings are trained through CNNs directly targeting the goal of matching. After getting the semantic features as what ARC-I does, DPAM applies self-attention and interactive attention mechanisms to highlight keywords and dependencies of the texts, and to enrich the feature vectors with more sophisticated semantic information. Finally, it calculates the cosine similarity of the upgraded multi-field semantic vectors.

(4) PALM: Besides DPAM, this model extracts and incorporates the geographic correlation based on Coordinate Embeddings. The location embeddings as well as the geographic features are learned through a 3-layer CNN. Then, a fully-connected layer will integrate the geographic features and the semantic features, and the output layer will calculate the cosine similarity of the integrated feature vectors.

(5) PALM+: This model keeps the same structure as PALM, but uses Kernel Embeddings to capture the influence of neighboring grids on geographic correlation.

We implement the models with Google's TensorFlow. The environment is CPU: $2{*}$E5-2630v4, Memory: 256G, and GPU: Tesla P40. The embedding dimensionality is $d = 64$, the CNN kernel number is $n = 64$, the batch size is 128, the learning rate is 0.001, and the cosine smoothing factor is $\gamma = 25$. Other parameters are learned during training.

4.2 Experimental Results

Visualization of Location Embedding A faithful location embedding should capture the intrinsic relation of the locations, so as to properly map them to the latent space for feature extraction. To analyze the faithfulness of our location embeddings, we visualize the embeddings of a central POI and another 100 randomly selected POIs. We project their d-dimensional embeddings to 2 dimensions by t-SNE (**?**).

As Figure **??** shows, the central POI is surrounded by POIs with different distances in the embedding space. The colors differentiate the POIs with different physical distance intervals from each group. For example, orange means that a POI is within $1km$ to the central POI in the physical space, while cyan-blue means that the physical distance between a POI and the central POI is far more than $6km$. The embeddings generated by PALM preserve the latitude&longitude distances of the physical space.

We also visualize the embeddings generated by DeepWalk (**?**). DeepWalk uses truncated POI sequences obtained from random walks, rather than the origin-destination pairs, to learn the latent representations of locations. This would generate false clicked origin-destination pairs for the training. For the same central POI and the same neighboring POIs, the embeddings generated by DeepWalk preserve the physical relation to some extent. However, there is also large chaos in the embeddings. This comparison demonstrates the faithfulness of our location embeddings.

Case Study on Attention Attention mechanisms in our model highlight the keywords of texts as well as the mapping between POI Name and POI Address. The following is an example of Query-POI matching with attention.

Query: “joy city”

POI$_1$ Name: “Chengdu **Joy City**”

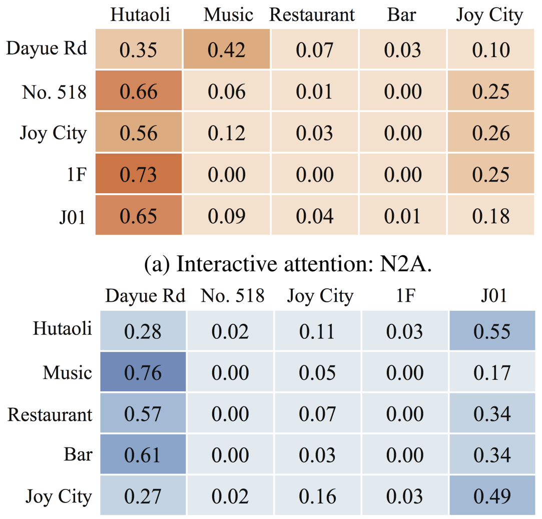
POI$_1$ Addr: “Intersection of **Dayue Rd** and **Taipingyuan** Middle 3rd Rd, Wuhou District”

POI$_2$ Name: “**Hutaoli** Music Restaurant & Bar (in Joy City)”

POI$_2$ Addr: “1F-**J01** Joy City, No. 518 **Dayue Rd**”

The query from Dataset A is “joy city”, a shopping mall in Chengdu, China. The retrieval system returns two POIs, which match the query well. However, with attention mechanisms, the first POI (which is exactly the shopping mall) recognizes its keywords and mapping dependencies, i.e., “Joy City”, “Dayue Rd” and “Taipingyuan”, while the second POI (which is a restaurant named Hutaoli in the shopping mall) gives more attention to keywords “Hutaoli”, “J01”, and “Dayue Rd”.

In Figure **??**, the first matrix shows the normalized attention weights of name words to each address word, and the second signifies which address words are the most relevant to each name word. For $N2A$, the name word “Hutaoli” (in deep orange) is strongly related to each address word; and for $A2N$, “Dayue Rd” and “J01” (in deep blue) are tightly bound to each name word. In our model, their relevance scores are 0.87 and 0.79, respectively. Therefore, POI 1 is marked as being more relevant to the query, which is consistent with the real-world click statistics.



	Hutaoli	Music	Restaurant	Bar	Joy City
Dayue Rd	0.35	0.42	0.07	0.03	0.10
No. 518	0.66	0.06	0.01	0.00	0.25
Joy City	0.56	0.12	0.03	0.00	0.26
1F	0.73	0.00	0.00	0.00	0.25
J01	0.65	0.09	0.04	0.01	0.18

(a) Interactive attention: N2A.

	Dayue Rd	No. 518	Joy City	1F	J01
Hutaoli	0.28	0.02	0.11	0.03	0.55
Music	0.76	0.00	0.05	0.00	0.17
Restaurant	0.57	0.00	0.07	0.00	0.34
Bar	0.61	0.00	0.03	0.00	0.34
Joy City	0.27	0.02	0.16	0.03	0.49

(b) Interactive attention: A2N.

Figure 4: Attention matrices of N2A and A2N. The attention weights differentiate the strong inter-dependencies between POI Name and POI Address from the weak ones.

Table 2: Comparison results in NDCG.

	Dataset A (Chengdu)		Dataset B (Nationwide)	
	NDCG@3	NDCG@10	NDCG@3	NDCG@10
DSSM	0.8246	0.8989	0.7617	0.8810
ARC-I	0.8298	0.9024	0.7558	0.8788
DPAM	0.8383	0.9058	0.7822	0.8907
PALM	0.8407	0.9050	0.8116	0.9022
PALM+	**0.8465**	**0.9110**	**0.8124**	**0.9027**

Quantitative Analysis of NDCG Table **??** summarizes the comparative results in terms of NDCG@3 and NDCG@10 on the two click-through datasets. On both datasets, our model beats DSSM and ARC-I.

Specifically, with multi-level text embedding and multi-field attention mechanisms, DPAM captures more sophisticated semantic features. Therefore, compared to DSSM and ARC-I, DPAM ranks the POIs with more accurate relevance scores. On the other hand, PALM outperforms DPAM, due to the integration of geographic features. The improvement is more distinct on Dataset B, where the input texts are more diverse and the driving distances are doubled the length in Dataset A. In PALM, geographic correlation significantly complements semantic similarity to model the relevance of Query-POI pairs. Finally, PALM+ which uses Kernel Embeddings to mitigate the problem of boundary effect achieves the best performance.

Figure **??** depicts the training loss of the five models on Dataset A. The validation loss shows a similar trend. Both models converge with the increase of epochs. Compared to our models, DSSM and ARC-I converge faster but to higher loss values (about 0.75). The loss values of PALM and PALM+ decrease more consistently than that of DPAM, and close to 0.70, indicating the functionality of the geographic module.

Overall, the results demonstrates the empirical effectiveness of integrating semantic similarity and geographic correlation for relevance modeling.

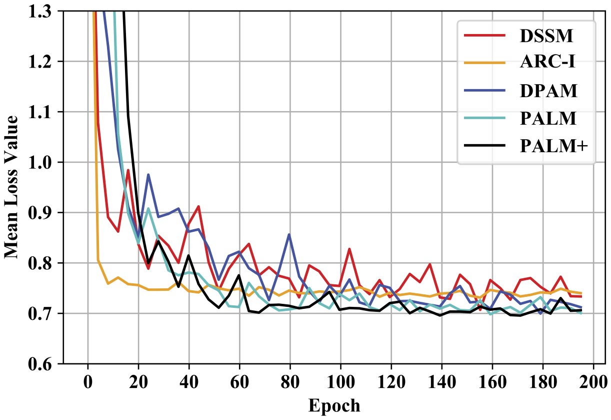


Figure 5: Training loss on Dataset A.

5 Conclusion and Discussion

In this paper, we propose a novel Query-POI relevance model for POI retrieval. The main contributions lie in three aspects. First, we exploit multi-level text embeddings and multi-field attention mechanisms to improve semantic matching for complex queries. Then, we propose two faithful location embeddings to learn the specific geographic correlation in ride-hailing scenarios, and incorporate the geographic correlation with the semantic similarity to capture more comprehensive features for Query-POI relevance analysis. Third, we conduct extensive experiments on two real-world click-through datasets to evaluate the model, and the experimental results show significant improvements in terms of NDCG over state-of-the-art models.

Notably, the model is effective for POI retrieval in various location based services, including but not limited to ride-hailing (e.g., Didi, Lyft, and Uber), local business search (e.g., Yelp), and on-demand food delivery (e.g. Uber Eats and Meituan-Dianping). Since ride-hailing users are highly sensitive to the retrieval result, their searching actions potentially set high standards for the relevance model. Hence, we take the ride-hailing services as the example in our paper. Although it is possible to get a unified view of matching in POI retrieval and POI recommendation, at present we have not compared our model (which focuses on the Query-POI relevance) with any POI recommendation models (which mainly solve the User-POI semantic gap), since they are based on different rationales, techniques, and datasets.

6 Acknowledgments

We thank all the reviewers and ACs for their helpful comments. We acknowledge the valuable support from Yashu Liu, Qichao Sun, and Guangju Peng.

References

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(Aug):2493–2537.

Hinton, G., and Salakhutdinov, R. 2011. Discovering binary codes for documents by learning deep generative models. *Topics in Cognitive Science* 3(1):74–91.

Hu, B.; Lu, Z.; Li, H.; and Chen, Q. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems (NIPS)*, 2042–2050.

Huang, P.-S.; He, X.; Gao, J.; Deng, L.; Acero, A.; and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Conference on Information and Knowledge Management (CIKM)*, 2333–2338.

Järvelin, K., and Kekäläinen, J. 2000. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 41–48.

Li, Y.; Fu, K.; Wang, Z.; Shahabi, C.; Ye, J.; and Liu, Y. 2018. Multi-task representation learning for travel time estimation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1695–1704.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS)*, 3111–3119.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery & data mining*, 701–710.

Robertson, S.; Walker, S.; Jones, S.; Hancock-Beaulieu, M.; and Gatford, M. 1996. Okapi at trec-3. In *Overview of the 3rd Text REtrieval Conference (TREC–3)*, 109–126.

Rosipal, R., and Krämer, N. 2006. Overview and recent advances in partial least squares. In *Proceedings of the 2005 International Conference on Subspace, Latent Structure and Feature Selection (SLSFS)*, 34–51.

Salakhutdinov, R., and Hinton, G. 2009. Semantic hashing. *International Journal of Approximate Reasoning* 50(7):969–978.

Seo, M.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Shen, Y.; He, X.; Gao, J.; Deng, L.; and Mesnil, G. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM)*, 101–110.

Socher, R.; Huval, B.; Manning, C. D.; and Ng, A. Y. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 1201–1211.

Song, X.; Kanasugi, H.; and Shibasaki, R. 2016. Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2618–2624.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web (WWW)*, 1067–1077.

Tur, G.; Deng, L.; Hakkani-Tür, D.; and He, X. 2012. Towards deeper understanding: Deep convex networks for semantic utterance classification. In *Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5045–5048.

Van Der Maaten, L., and Hinton, G. 2008. Visualizing high-dimensional data using t-SNE. 2579–2605.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS)*, 5998–6008.

Yu, A. W.; Dohan, D.; Luong, M.-T.; Zhao, R.; Chen, K.; Norouzi, M.; and Le, Q. V. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

Zhang, J.; Zheng, Y.; Qi, D.; Li, R.; Yi, X.; and Li, T. 2018. Predicting citywide crowd flows using deep spatio-temporal residual networks. *Artificial Intelligence* 259:147–166.