POLARIS: A Geographic Pre-trained Model and its Applications in Baidu Maps

Jizhou Huang, Haifeng Wang, Yibo Sun, Yunsheng Shi, Zhengjie Huang, An Zhuo, Shikun Feng
Baidu Inc., China
{huangjizhou01,wanghaifeng,sunyibo,shiyunsheng01,huangzhengjie,zhuoan,fengshikun01}@baidu.com

ABSTRACT

Pre-trained models (PTMs) have become a fundamental backbone for downstream tasks in natural language processing and computer vision. Despite initial gains that were obtained by applying generic PTMs to geo-related tasks at Baidu Maps, a clear performance plateau over time was observed. One of the main reasons for this plateau is the lack of readily available geographic knowledge in generic PTMs. To address this problem, in this paper, we present POLARIS, which is a geographic pre-trained model designed and developed for improving the geo-related tasks at Baidu Maps. POLARIS is elaborately designed to learn a universal representation of geography-language by pre-training on large-scale data generated from a heterogeneous graph that contains abundant geographic knowledge. Extensive quantitative and qualitative experiments conducted on large-scale real-world datasets demonstrate the superiority and effectiveness of POLARIS. POLARIS has already been deployed in production at Baidu Maps since April 2021, which significantly benefits the performance of a wide range of downstream tasks. This demonstrates that POLARIS can serve as a fundamental backbone for geo-related tasks.

CCS CONCEPTS
• Information systems → Data mining.

KEYWORDS
pre-training, heterogeneous graph, graph neural network

1 INTRODUCTION

Pre-trained models (PTMs) are designed to learn a universal representation from large-scale raw text [6, 27, 30], unlabeled images [22], or videos [28], which have become a fundamental backbone for downstream tasks in natural language processing (NLP) and computer vision (CV) [3]. The most common paradigm for adapting PTMs to downstream tasks is sequential transfer learning [26] via supervised fine-tuning on labeled data. In this paradigm, downstream tasks can benefit from the knowledge learned by PTMs, which brings significant improvements [6].

The web mapping services provided by Baidu Maps, such as point of interest (POI) retrieval [12], POI auto-completion [7, 13], POI recommendation [4], and POI information page [29], have shown improved performance by applying PTMs. However, a clear performance plateau over time was observed in our practice, i.e., the performance gain remains marginal w.r.t. the optimization of generic PTMs. One of the main reasons for this plateau is the lack of geographic knowledge, which plays a vital role in improving tasks that necessitate computational support for geographic information (hereafter referred to as geo-related tasks). In this work, we focus on two types of geographic knowledge. (1) Toponym knowledge. A toponym refers to the name of a geo-located entity, such as a POI, a street, and a district. Toponym resolution [17], which aims at identifying and extracting toponyms from text, is a fundamental necessity for a wide range of geo-related tasks. However, the semantic meaning of most toponyms can hardly be captured by generic PTMs, because toponym knowledge is largely absent from or rarely seen in their training data. (2) Spatial knowledge. Spatial knowledge mainly includes the geographic coordinates of a geo-located entity and the spatial relationships between different geo-located entities, which is indispensable for geo-related tasks such as geocoding [9] and georeferencing [11]. However, the generic PTMs are incapable of handling geo-related tasks effectively, due to the absence of spatial knowledge and the lack of pre-training tasks for incorporating spatial knowledge.

To effectively learn a geographic pre-trained model from large-scale spatial data, we need to address two key challenges. (1) Heterogeneous data integration. Figure 1 illustrates the main difference between the training data used by generic and geographic PTMs. Generic PTMs are typically learned from multimodal data, including text, images, and speech. By contrast, the spatial data mainly include spatial location (a single POI), spatial correlation (a...
triplet of POIs), and human mobility (a sequence of POIs). However, the way to effectively integrate the three sources of spatial data with text data for training geographic PTMs has been little explored and remains a challenge. (2) **Geography-language pre-training.** Different from existing language model pre-training [6] and vision-language (image-text [22] and video-text [28]) pre-training that is designed to learn the semantic correlations between vision and language, geography-language pre-training necessitates learning the associations between geography and language, e.g., learning to associate “Beijing railway station” in text with its real-world geolocation information in the form of coordinates. To learn a cooperative knowledge of geography and language from unlabeled data, it is important to find effective backbone networks, pre-training tasks, and learning objectives.

In this paper, we present our efforts toward designing and implementing POLARIS, which is a geographic pre-trained model designed for improving a wide range of geo-related downstream tasks at Baidu Maps. Specifically, we first construct a heterogeneous graph that contains POI nodes and query nodes, using the POI database and search logs of Baidu Maps. To integrate spatial information with text, we construct edges between two nodes based on spatial correlation and human mobility data, as shown in Figure 1, which enable knowledge transfer between different modalities. To generate each input sequence for training POLARIS, we use the random walk algorithm to sample a sequence of nodes as an input document. In this way, we can automatically build large-scale training data, which facilitate comprehensive knowledge transfer and bridge the modality gap. Second, we use transformer as the backbone network to learn the representations of each node. To incorporate an input document’s graph information, we employ a transformer-based aggregation layer to encode the relations between multiple nodes in the document. To effectively learn comprehensive knowledge, we adopt masked language modeling (MLM) and geocoding as the pre-training tasks, which are elaborated to simultaneously learn toponym and spatial knowledge, as well as to balance both knowledge explorations. As such, we can learn a universal representation of geography-language by pre-training on domain-specific and cross-modality data.

We evaluate the performance of POLARIS on five geo-related tasks. Extensive experiments conducted on large-scale, real-world datasets collected from Baidu Maps show that POLARIS significantly outperforms the generic PTMs when applied to all 5 tasks. POLARIS has already been deployed in production at Baidu Maps since April 2021, which significantly benefits the performance of a wide range of downstream tasks. This demonstrates that POLARIS can serve as a fundamental backbone for geo-related tasks.

Our contributions can be summarized as follows:

- **Potential impact:** We suggest a practical and robust solution for training a geographic pre-trained model, named POLARIS, which can serve as a fundamental backbone for geo-related tasks. We document our efforts and findings on designing and developing POLARIS, and we hope that it could be of potential interest to practitioners working with pre-trained models and geo-related problems.

- **Novelty:** The design and development of POLARIS are driven by the novel idea that learns a universal representation of geography-language by pre-training on large-scale graph data with both toponym and spatial knowledge.

- **Technical quality:** Extensive quantitative and qualitative experiments, conducted on large-scale, real-world datasets, demonstrate the superiority and effectiveness of POLARIS. The successful deployment of POLARIS at Baidu Maps further shows that it is a practical and fundamental backbone for a wide range of geo-related tasks.

2 POLARIS

In this section, we introduce the design and implementation details of POLARIS, which mainly contains three parts: training data construction, model architecture, and pre-training tasks.

2.1 Training Data Construction

Our previous work [12] has demonstrated that the heterogeneous graph is able to significantly benefit POI retrieval task. Motivated by this, we construct large-scale training data based on a heterogeneous graph that contains both toponym and spatial knowledge for pre-training POLARIS. Specifically, we first construct a unified heterogeneous graph that contains POI nodes and query nodes using the POI database and search logs of Baidu Maps. Then, we construct edges between two nodes based on spatial relationships between POIs to integrate spatial information with text. The toponym data mainly include POI names and addresses, which are derived from the POI database and are stored in unstructured text format. The spatial data consist of POI geographic coordinates, POIs that co-locate within individual geographic regions, and POIs that co-occur in same sessions from search logs, which are stored using numerical digits or in triplet format (i.e., non-text format).

To bridge the gap between the text and non-text representations, we build a heterogeneous graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{O}_\mathcal{V}, \mathcal{R}_\mathcal{E}) \), where \( \mathcal{V} \) denotes the set of nodes, \( \mathcal{E} \) the set of edges, \( \mathcal{O}_\mathcal{V} \) the set of node types, and \( \mathcal{R}_\mathcal{E} \) the set of edge types. Each node \( v \in \mathcal{V} \) and each edge \( e \in \mathcal{E} \) are associated with their corresponding mapping functions \( \psi(e) : \mathcal{V} \rightarrow \mathcal{O}_\mathcal{V} \) and \( \phi(e) : \mathcal{E} \rightarrow \mathcal{R}_\mathcal{E} \). As shown by Figure 2, the node types \( \mathcal{O}_\mathcal{V} \) include POI and query. The edge types \( \mathcal{R}_\mathcal{E} \) include Query-click-POI, Origin-to-Destination, and POI-(co-locate-with)-POI. Next, we detail each element of the heterogeneous graph.

2.1.1 POI Node and Query Node. Each query node is the text used to search a user’s desired POI. Each POI node represents a POI in the POI database. We organize each POI node in the form of the concatenation of the following three types of text information: (1) the full POI name, (2) the POI address, and (3) the POI type. We separate each type of text information with a [SEP] token. We also equip each POI node with its real-world location information, i.e., the geographic coordinates of it.

2.1.2 Query-click-POI Edge. After typing in a query, a user would click on the desired POI from a list of ranked POIs that the POI search engine suggested. This process produces large-scale query-POI pairs, where the different expressions of each POI can bridge the semantic gap between queries and POIs. For example, users usually make spelling errors or use abbreviations, which would lead to poor results when directly matching query and POI text information. Motivated by this observation, we model the relations between
queries and POIs using the Query-click-POI edge. Specifically, we select the top 4 searched queries for each POI and connect an Query-click-POI edge for every POI and its historical query nodes.

2.1.3 Origin-to-Destination Edge. A user’s mobility behavior produces a visited POI sequence in search logs. From which, the origin POI and destination POI can be extracted to construct the Origin-to-Destination edge between two POIs. Specifically, we perform a 2-gram sliding window on the POI sequences and link the adjacent POIs with such edge type.

2.1.4 POI-(co-locate with)-POI Edge. POIs that co-locate within individual geographic regions may exhibit a high degree of spatial similarity and share common features. Motivated by this observation, we build an additional spatial relations between different POIs by introducing the POI-(co-locate with)-POI edge. As shown in the middle of Figure 2, we quantize the Earth’s surface as a grid and construct the co-location edge between POIs that lie in the same cell of the grid. This can be regarded as analogy to the surrounding words, where each word in a sentence is surrounded by a textual context. In this way, each POI node in the heterogeneous graph has a spatial context representing the real-world spatial distribution.

Specifically, we use the S2 geometry\(^1\) library to construct the grid, which discretizes the Earth’s surface hierarchically based on space-filling curves. We choose the S2 library because it uses spherical projections to avoid the distortion brought by alternatives that use planar projection, thus preserving the Earth’s correct topology. The S2 library support 31 levels of hierarchy, where its grid’s cells have different area coverage. For simplicity, we use the term “S2 cell” to refer to the cells generated by the S2 library. We use the hierarchy of level 15 to construct the co-location edges between the POIs in the same S2 cell covering the area of ~200 m\(^2\).

2.1.5 Random Walk Sampling. Upon \(G\), we use the random walk algorithm to sample a sequence of nodes as an input document \(D = \{v_1, v_2, ..., v_n\}\), where \(n\) is the length of \(D\). Each node \(v_i\) has a text representation consisting of a sequence of words \(W_i = \{s_1^i, ..., s_j^i, ..., s_k^i\}\). During random walk sampling, at time step \(i\), we sample the node by considering the influence of different edges. Specifically, we use the weighted probability distributed over the neighbors of \(v_i\) as the transition probability \(p(u|v)\):

\[
p(u|v) = \begin{cases} 
\frac{1}{|N(\epsilon_{v,i})|}, & \forall \phi(\epsilon_{u,i}) \in \text{Query-click-POI edge} \\
\frac{1}{|N(\epsilon_{v,i})|}, & \forall \phi(\epsilon_{u,i}) \in \text{Origin-to-Destination edge} \\
\frac{1}{|N(\epsilon_{v,i})|}, & \forall \phi(\epsilon_{u,i}) \in \text{POI-(co-locate with)-POI edge}
\end{cases}
\]

where \(|N(\epsilon_{v,i})|\) denotes the number of \(v\)’s neighborhood with corresponding edge and \(\lambda_i\) is the weight corresponding to different edges. Algorithm 1 shows the details.

\[\text{Algorithm 1 Random Walk Sampling on Heterogeneous Graph}\]

**Input:** Heterogeneous Graph \(G = (V, E, O_A, R_E); n \text{ walk length} \]

**Output:** Sequence of text nodes \(D = \{v_1, v_2, ..., v_n\} \)

1. for each \(v \in G\) do
2. \(D = \{v_i\} \)
3. for \(i = 0 \) to \(n \) do
4. \( \text{draw } u \text{ according to Equation } 1. \)
5. \(D.\text{push}(u) \)
6. \(v \equiv u \)
7. end for
8. end for

2.2 Model Architecture

As shown in Figure 3, the two major components in POLARIS’s model architecture are a multi-layer bidirectional transformer [32] encoder and a transformer-based aggregation (TranSAGE) layer. The document in traditional NLP tasks consists of multiple sentences, where discourse structure of the document should be considered. By contrast, the input document \(D\) for pre-training POLARIS

\[\text{https://s2geometry.io}\]
consists of a sequence of nodes, and there is no discourse structure in $D$. Therefore, instead of concatenating all the nodes and modeling them as one text sequence, we use the transformer encoder to get each node’s hidden vector separately, and employ the TranSAGE layer to capture the relations between each node and its neighbors. As a graph-contextualized representation, the output of the TranSAGE is fused with each node’s vector. Then, each node’s fused representation is used for the pre-training tasks.

Formally, for each node $v_i$ in document $D$, we first use the sentence-piece algorithm to tokenize its text representation $W_i$ into a sub-word sequence $S_i = \{s_i^{j_1}, ..., s_i^{j_L}\}$, where $s_i^{j}$ denotes a sub-word token and $L$ is the length of the sub-word sequence. Then, we insert a “[CLS]” token to the head of the tokenized sequence and use a multi-layer bidirectional Transformer [32] encoder to get the node $v_i$’s vector representation $(\hat{h}_i^{CLS}, \hat{H}_{context}^i)$ as follows:

$$ (\hat{h}_i^{CLS}, \hat{H}_{context}^i) = \text{Transformer}([\text{CLS}], s_i^{j_1}, ..., s_i^{j_L}) $$  \hspace{1cm} (2)

where $\hat{H}_{context}^i = \{\hat{h}_i^{j_1}, ..., \hat{h}_i^{j_L}\}$ and $\hat{h}_i^{j} \in \mathbb{R}^{db}$ is the hidden vector of $s_i^{j}$, $\hat{h}_i^{CLS} \in \mathbb{R}^{db}$ is the final hidden vector of the special token “[CLS]”, which is the aggregated representation of $S_i$.

After we obtain each node’s vector representation, we use the TranSAGE layer to model the relations between different nodes. This layer adopts a node type-aware multi-head attention mechanism to aggregate the heterogeneous graph information. The vanilla multi-head attention [32] mechanism projects an input sequence into a key matrix and a value matrix with corresponding weights. The attention scores are computed by the dot product of these two matrices. Considering such attention mechanism is agnostic of the different node’s type in the heterogeneous graph, we use different weights for the projection based on the node’s type. Specifically, we first pack the aggregated representations of all nodes together into a matrix $H = \{\hat{h}_1^{CLS}, ..., \hat{h}_n^{CLS}\}$. Then, we use the TranSAGE layer to compute the output matrix $\hat{H}$ as follows:

$$ \hat{H} = \text{concat}(\text{head}_1, ..., \text{head}_d)W^O $$

$$ \text{head}_j = \text{softmax}\left(\frac{Q_jK_j^T}{\sqrt{d}}\right)H $$

$$ Q_j = \text{Q-Linear}_{\psi(v_j)}(H) $$

$$ K_j = \text{K-Linear}_{\psi(v_j)}(H) $$

where $\text{Q-Linear}_{\psi(v_j)} : \mathbb{R}^{db} \rightarrow \frac{\mathbb{R}^{db}}{L}$ is a linear projection indexed by each node’s type, and $L$ is the number of the head. We use this projection layer to convert $H$ into a query matrix $Q_j$ for the $j$-th head, where nodes with different types are computed with unique parameters. Similarly, we use another linear projection $K-\text{Linear}_{\psi(v_j)}$ to compute a key matrix $K_j$ for the $j$-th head.

Finally, we apply another attention-based module to each sub-word representation $\hat{h}_i^{j}$ with its corresponding graph contextualized representation $\hat{h}_i^{CLS}$ to compute the pre-training objectives as follows:

$$ (\hat{\tilde{K}}_i^{CLS}, \hat{\tilde{H}}_{context}^i) = \text{Transformer}(\hat{h}_i^{CLS}, \hat{H}_{context}^i) $$

$\hat{\tilde{K}}_i^{CLS}$ is used for training the geocoding task and $\hat{\tilde{H}}_{context}^i$ is used for training the masked language modeling (MLM) task.

### 2.3 Pre-training POLARIS

#### 2.3.1 MLM

We use the whole word mask (WWM) strategy to make predictions for the phrases in each document. We use a query component analysis module deployed at Baidu Maps to split each document at the granularity of geographic entities. Each geographic entity in a document has a 15% probability of being masked and predicted by the language model during the training process. For each
word in the selected entity, we replace the word with a "[MASK]" token with 70% probability, replace the word with a misspelled word with 10% probability, replace the word with a random word with 10% probability, and leave the word unchanged with 10% probability. The words in a query that do not match any words in the target POI name are treated as misspelled words. With this training procedure, we can learn four types of toponym knowledge in the MLM task as follows. (1) The natural language description of POI name and address. (2) The relationships between POI name, address, and type. (3) The relationships between query, POI name, and address. (4) The possible misspelling of POI name and address.

2.3.2 Geocoding. The geocoding task is designed to learn the relation between text representation and geographic coordinates of a POI. Specifically, we adopt a feed-forward layer for each POI node to predict the IDs for multi-level S2 cells converted from the POI’s coordinates. In order to model the relations between text and coordinates in a more fine-grained manner, we set the highest level of S2 cell to 22, which covers an area approximate to 2 m². However, in the granularity of level 22, the Earth’s surface is divided into 105 trillion S2 cells. Directly predicting these S2 cell’s IDs will introduce an overwhelming number of output classes. Thus, as shown in Figure 4, we use a one-to-one mapping to covert each S2 cell to a token consisting of alphabet and numbers, and then make predictions on each character of the token.

Input: Baidu Techn Park, No.10 Xibeiwang East Road, Haidian District, Beijing, China
Output: ...
35f1c (L7)
35f1b (L8)
35f1ac (L9)
35f1a9 (L10)

Figure 4: Illustration of the geocoding task. Each parallelogram represents an S2 cell of the corresponding level. Each string marked with colors at bottom right represents its corresponding S2 cell’s token.

In such representation, for every two more levels, the length of the token increases by one. The token of level $2n – 1$ and $2n$ differs in the last character. For example, a same point can be represented by four consecutive-level (from 7 to 10) S2 cell tokens: "351c", "35f1b", "35f1ac", and "35f1a9". Thus, for each position in the token, we predict three labels: two for the last character of $2n – 1$ and $2n$ level, respectively, and one for the penultimate character of these two consecutive levels. In this way, sufficient information can be predicted for decoding all levels less than the max level supported by the length of the token.

Specifically, for each label of a specific position in the output token, we use a softmax layer to compute the probability of classifying $q_i$ as a certain character by:

$$Pr(c_i|q_i) = \text{softmax}(h^l_{CLS}W_c),$$ (5)

where $W_c \in \mathbb{R}^{d_w \times 16}$ is the trainable parameter, and $Pr(c_i|q_i)$ is the probability vector of a category $c_i \in \{0, \ldots, 15\}$ representing 6 English letters (a to f) and 10 Arabic numerals (0 to 9).

3 EXPERIMENTS
In this section, we present the results of POLARIS on five geo-related tasks and the ablation experiments.

3.1 Geo-related Tasks

Task #1: Query Intent Classification. The query intent classification [19] task aims at predicting the intent behind a query, which plays an important role in the POI search engine of Baidu Maps. We define four intents, including the search for a specific POI, for a specific type of POI, for addresses, and for bus routes.

To evaluate its performance, we randomly sample 60,000 queries from the search logs of Baidu Maps and manually annotate them.

In our experiments, we use $h^l_{CLS}$ (see Equation 2) as the input representation, based on which a linear layer is adopted to perform the classification. We use accuracy as the evaluation metric, which represents the proportion of correctly predicted queries.

Task #2: Query-POI Matching. The query-POI matching [12, 13] task aims at identifying the more relevant POI for a given query from a list of POIs. There are four levels of relevance: the POI is exactly matched to the query, highly relevant, weakly relevant, and irrelevant. The relevance score predicted by the matching model is an important feature for ranking the candidate POIs in the search engine of Baidu Maps.

To construct the dataset for this task, we randomly sample real-world queries from the search logs of Baidu Maps. For each query, we randomly sample 6 POIs from the top 10 POI candidates ranked by the POI search engine of Baidu Maps, as well as another four random POIs. Then, we ask annotators to manually examine the 10 candidate POIs of each query and assign relevance levels to them.

In our experiments, we also use $h^l_{CLS}$ to perform this task and employ accuracy as the evaluation metric.

Task #3: Address Parsing. The task of address parsing [18] aims at parsing an address into a sequence of fine-grained geo-related chunks. In this work, we designed 22 chunk types, among which 9 are for different levels of geographic areas, 2 for roads, 3 for different types of POI, 5 for details of POI location, and 3 for the auxiliary words that describe the location. Plenty of modules in Baidu Maps, including a rule-based geocoding framework and the query understanding module of the POI search engine, rely on the output of the address parsing model.

To build the dataset for this task, we utilize two data sources. One data source comes from the daily processed delivery address data from the geocoding service of Baidu Maps. The other is collected from queries in the search logs. Then, we ask the annotators to annotate every possible chunk of the addresses and queries.
We formulate this task as a sequence labeling problem and use POLARIS + CRF as the model architecture. We employ the entity-level $F_1$ score as the evaluation metric to measure the performance of chunk detection.

**Task #4: Geocoding.** Given a geo-located entity reference in text, the geocoding task [9] aims at resolving the input to a corresponding location on Earth. As described in section 2.3.2, we directly predict S2 tokens for the input text. The geocoding task is an essential service of mapping applications, its output is also a crucial feature required by other services like POI retrieval.

To collect the dataset for geocoding task, we use the same delivery address data as those built for above-mentioned address parsing task. After processing by the geocoding service, each address is correlated with an S2 token converted from their geographic coordinates. In this way, we can automatically construct large amounts of training data. Since the accuracy of our existing geocoding service cannot reach 100%, we manually annotate 2,000 addresses as the development set and test set, respectively. Different from the well-formatted address descriptions derived from our POI database for pre-training POLARIS, the address descriptions in this dataset are generated by different users with varied knowledge and are not well formatted. As such, the data leakage problem can be avoided. Therefore, we can use this dataset to sufficiently validate a model’s ability to correlate text with geographic coordinates.

The model architecture used for fine-tuning this task is the same as that described in Section 2.3.2. We use “Accuracy@N km” as the evaluation metric, which measures the percentage of predicted locations that are apart with a distance less than N km to their actual physical locations. In our experiments, we set N to 3.

**Task #5: Next POI Recommendation.** Given a user’s sequence of historical POI visits, the task of next POI recommendation (NPR) [4, 33] aims at recommending a list of POIs that the user is most likely to visit consequently. NPR is an essential feature in the information page of Baidu Maps, which can help users explore the neighborhood with minimal operations.

To construct the dataset for NPR task, we use the POI visiting data in Beijing within a 6-month period from Baidu Maps. For each POI sequence, we process a sliding window with a randomly selected width from 3 to 6 to get some sub-sequences, where the last POI is regarded as the label and the rest of POIs are taken as the historical visits. For evaluation, we use 6 million POIs in Beijing as candidate POIs for retrieval.

In fine-tuning phase, we use a two-tower approach similar to that used in our previous work [12]. Taking POLARIS as a feature encoder, we calculate the similarity between the graph-based visiting sequence representation and the target POI representation.

The key difference between pre-training and fine-tuning is that the input graph only contains POI-POI edges constructed with the POI visiting sequences. In evaluation phase, we first generate all the vectors for input sequences and candidate POIs. Then, we use the HNSW [24] algorithm to process an approximate K-nearest neighbor search for retrieving the target POI. We use Acc@K as the evaluation metric, which calculates the proportion of the recommended POI sequences where the visited POI appears within the top K positions. In our experiments, we set K to 50.

The five tasks and datasets used to evaluate POLARIS are summarized in Table 1.

| Task | Problem Formulation | Applicable Service | #Train | #Dev | #Test | Metric |
|------|---------------------|--------------------|--------|------|-------|--------|
| Query intent classification | Sequence classification | POI search engine | 50,000 | 5,000 | 5,000 | Accuracy |
| Query-POI matching | Sequence pair classification | POI search engine | 140,614 | 4000 | 4000 | Accuracy |
| Address parsing | Sequence labeling | POI information processing | 125,009 | 10,000 | 10,000 | $F_1$ entity-level |
| Geocoding | Sequence classification | POI information processing | 2,171,114 | 1,000 | 1,000 | Acc@N km |
| Next POI recommendation | Relevance ranking | POI recommendation | 6,398,231 | 300,000 | 300,000 | Acc@K |

### 3.2 Experimental Setup

#### 3.2.1 Datasets.** In our experiments, we construct the heterogeneous graph using search logs within a 3-month period from Baidu Maps. The heterogeneous graph contains 40 million POI nodes, 120 million query nodes, 175 million Query-click-POI edges, 1,574 million Origin-to-Destination edges, and 363 million POI-(co-locate with)-POI edges. We use random walk algorithm on the heterogeneous graph to sample a sequence of nodes as an input document. The sampling weights of Query-click-POI, Origin-to-Destination, and POI-(co-locate with)-POI edges are set as $\lambda_1 = 0.5$, $\lambda_2 = 0.25$, and $\lambda_3 = 0.25$, respectively. We sample 800 million documents from the graph, which contain 400 billion words. Each document contains an average of 10 nodes.

#### 3.2.2 Baselines.** We evaluate POLARIS against three strong generic PTMs as follows:

- BERT [6] is a transformer-based pre-trained language model, which has made impressive gains on many NLP tasks.
- RoBERTa [21] is a variant of BERT with improved pre-training strategies, which further improve the outperforms BERT on several NLP tasks.
- ERNIE [30] is another variant of BERT, which facilitates continuous learning of multiple pre-training tasks. It outperforms BERT and RoBERTa on many Chinese NLP tasks.

We also perform ablation experiments over a number of facets of POLARIS to figure out their relative importance, which include:

- **POLARIS** is the complete model depicted in Section 2.
- **POLARIS w/o a specific type of edge.** This group contains three settings for removing the edge of Origin-to-Destination (O-t-D), query-click-POI (Q-c-P), and POI-(co-locate with)-POI (P-c-P) when constructing the heterogeneous graph, respectively.
- **POLARIS w/o heterogeneous graph.** In this setting, we remove all edges in the heterogeneous graph and pre-train
Table 2: Comparison of pre-trained models on five geo-related tasks. Average means the averaged score of five tasks.

| Pre-trained Model | Query Intent Classification | Query-POI Matching | Address Parsing | Geocoding | Next POI Recommendation | Average |
|-------------------|-----------------------------|--------------------|----------------|-----------|-------------------------|---------|
| BERT [6]          | 0.8875                      | 0.8279             | 0.8452         | 0.4592    | 0.1092                  | 0.6258  |
| RoBERTa [21]      | 0.8907                      | 0.8285             | 0.8497         | 0.4618    | 0.1115                  | 0.6284  |
| ERNIE [30]        | 0.8919                      | 0.8290             | 0.8511         | 0.4636    | 0.1198                  | 0.6311  |
| POLARIS           | **0.9161**                  | **0.8332**         | **0.8794**     | **0.6545**| **0.1556**              | **0.6878**|
| - w/o geocoding task | 0.9068                      | 0.8050             | 0.8682         | 0.5436    | 0.1301                  | 0.6507  |
| - w/o heterogeneous graph | 0.9101                      | 0.8025             | 0.8688         | 0.5809    | 0.1359                  | 0.6596  |
| - w/o O-t-D edge   | 0.9076                      | 0.8129             | 0.8715         | 0.6273    | 0.1403                  | 0.6719  |
| - w/o Q-c-P edge   | 0.9155                      | 0.8072             | 0.8784         | 0.6381    | 0.1413                  | 0.6761  |
| - w/o P-c-P edge   | 0.9148                      | 0.8305             | 0.8780         | 0.6164    | 0.1458                  | 0.6771  |

POLARIS using the text representation of each node, as described in section 2.1.1.
- **POLARIS w/o geocoding task.** In this setting, we remove the pre-training task of geocoding described in Section 2.3.2.

All the pre-training and fine-tuning procedures are implemented using the PaddlePaddle deep learning framework [23]. We use Adam optimizer [15], with the learning rate initialized to $5 \times 10^{-5}$ and gradually decreased during the process of training. The hyper-parameters of all PTMs are the same as those used in BERT_BASE [6] (number of hidden layers = 12, hidden layer size = 768, number of attention heads = 12, number of total parameters = 110M). The training takes about one week on 16 A100 GPUs.

### 3.3 Results and Analysis

Table 2 shows the main experiment results. The last column “Average” presents the averaged score of five tasks.

#### 3.3.1 Overall Performance

We first evaluate whether POLARIS can improve the performance of five geo-related tasks. Before POLARIS, we have applied ERNIE to improve these tasks at Baidu Maps and obtained initial gains. However, a clear gain plateau over time was observed due to the lack of geographic domain knowledge in ERNIE. This motivated us to design and develop a geographic pre-trained model POLARIS for improving the geo-related tasks at Baidu Maps. The results in Table 2 show that POLARIS significantly outperforms all three generic PTMs (i.e., BERT, RoBERTa, and ERNIE) by a large margin, and achieves a highest average score of 0.6878. This demonstrates that our model is more effective in dealing with geo-related tasks. One of the main reasons for this superiority is the geographic knowledge comprehensively learned by POLARIS.

#### 3.3.2 Ablation Studies

We perform ablation experiments to better understand the relative importance of different facets of POLARIS.

First, we study the effect of pre-training tasks. Compared with POLARIS, the average score of "POLARIS w/o geocoding task" dramatically decreases by an absolute 3.71%, which is the largest drop observed among all ablation experiments. This demonstrates the impact brought by the geocoding task. The main reason is that geographic knowledge plays a vital role in geo-related tasks. Therefore, the ability to learn a universal representation of geography-language is crucial for pre-training a geographic model.

Second, we evaluate the effect of heterogeneous graph. The results in Table 2 show that removing the graph (POLARIS w/o heterogeneous graph) hurts performance significantly on all tasks. This demonstrates the significance of graph structure in the training data, which can effectively integrate spatial knowledge with text.

Third, we examine the impact of different edge types. The results show that removing individual edges (POLARIS w/o P-c-P edge, w/o O-t-D edge, and w/o Q-c-P edge) hurts performance on all tasks. This indicates that all types of edges are essential for pre-training a geographic model, and they can work together as complements. Removing the O-t-D edge (POLARIS w/o O-t-D edge) leads to the largest drops among the three ablations of edges, this demonstrates the importance of human mobility data in geo-related tasks.

Finally, we evaluate the performance of individual tasks. (1) Compared with ERNIE (the best performing generic PTM), the performance of geocoding trained with POLARIS achieves the largest increase of 19.09% (by absolute value). Moreover, the “POLARIS w/o geocoding task” model performs worse than the POLARIS model on the geocoding task, with a large drop of 11.09% (by absolute value). The main reason is that the performance of a geocoding model heavily relies on its ability to correlate text with geographic coordinates. This demonstrates that POLARIS has learned sufficient geographic knowledge about text and coordinates during pre-training. (2) The task that has the second highest benefits among five tasks is next POI recommendation, with an absolute improvement of 3.58% over ERNIE. Moreover, removing the O-t-D edge (POLARIS w/o O-t-D edge) leads to a drop of 1.53% on next POI recommendation task. The main reason is that prior knowledge of the distribution of human mobility data is crucial for next POI recommendation task. This indicates that POLARIS has learned the distribution of human mobility from the training data. (3) Among the generic PTMs, ERNIE outperforms BERT and RoBERTa on five tasks. This shows that the optimization made by ERNIE for dealing with Chinese NLP tasks, such as the masking strategy that masks phrases and named entities rather than individual sub-words, can be a benefit for Chinese toponym masking.

### 3.4 How Much Geographic Knowledge POLARIS Has Learned: A Qualitative Study

#### 3.4.1 Embedding projection

For an intuitive understanding of how much geographic knowledge POLARIS has learned, we encode POIs
Figure 5: A 2D t-SNE projection of top 500 searched POIs in 31 provinces excluding Hong Kong, Macao, and Taiwan of China.

(a) The t-SNE visualization of embeddings produced by BERT. 
(b) The t-SNE visualization of embeddings produced by POLARIS.

Figure 6: Nearest Neighbors of “Huangpu District - Shanghai + Beijing”. The darker bar represents the district locating in Beijing.

(a) Results ranked by BERT. 
(b) Results ranked by POLARIS.

Figure 7: Nearest Neighbors of “Guangdong Province - Guangzhou + Kunning”. The darker bar represents Yunnan Province, the capital city of which is Kunning.

(a) Results ranked by BERT. 
(b) Results ranked by POLARIS.

Figure 6 and Figure 7, we test the geographic analogy of “district of a city” and “capital city of a province”, respectively. We show the top 10 neighbors ranked by their cosine similarity to the query. In Figure 5, the query is set to “Huangpu District - Shanghai + Beijing”, and the candidate neighbors are set to district names of all Chinese provinces. We can observe that POLARIS recalls more Beijing’s districts than BERT. In Figure 7, the query is set to “Guangdong Province - Guangzhou + Kunning”, and the candidate neighbors are set to all Chinese province names. We can observe that POLARIS recalls the target neighbor, “Yunnan Province”, with the highest score.
Moreover, in both Figures, the cosine similarity predicted by BERT is less discriminative than that by POLARIS. Such observations show that POLARIS has learned the spatial relationships between different geo-located entities.

4 RELATED WORK
Here we briefly review the closely related work in the fields of domain-specific PTMs and PTMs utilizing multi-source data.

4.1 Domain-specific PTMs
Existing domain-specific PTMs mainly lie in the domain of health-care [1, 14], biomedical [8, 10, 16], and academic & research [2]. Most of them learn the domain-specific knowledge by pre-training on domain-specific corpora with the MLM pre-training task. The most relevant work to ours is OAG-BERT [20], which is an academic PTM pre-trained using the heterogeneous structure knowledge from an academic knowledge graph. However, they do not model the graph structure explicitly, like our proposed method.

4.2 PTMs with Multi-source Data
Most existing multimodal PTMs are designed to model the relations between text and image [22], video [28], and audio [5]. However, pre-training a geographic PTM requires modeling the relations between text and numerical data, such as geographic coordinates. Such an intersection of multiple modalities of text and numbers has not been well explored in the literature.

5 CONCLUSIONS AND FUTURE WORK
This paper presents an industrial solution for building a geographic pre-trained model which has already been deployed at Baidu Maps. We propose a framework, named POLARIS, that comprehensively learns geographic domain knowledge. Sampled from a heterogeneous graph constructed upon the POI database and the search logs of Baidu Maps, the documents used for pre-training POLARIS are injected with toponym and spatial knowledge. The backbone network of POLARIS contains an aggregation layer for modeling the graph structure entailed in the input documents. POLARIS adopts two pre-training objectives, including masked language model and graph structure entailment in the input documents. POLARIS adopts the geographic pre-trained model with data from more modalities such as satellite images and street view images.

REFERENCES
[1] Emily Alsentzer, John Murphy, et al. 2019. Publicly Available Clinical BERT Embeddings. In ClinicalNLP ‘19, 72–78.
[2] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. ArXiv abs/1907.11692 (2019).
[3] Diederik P. Kingma, and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In ICLR.
[4] Jochen L. Leidner. 2007. Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding. SIGIR Forum 41, 2 (dec 2007), 124–126.
[5] Hao Li, Wei Lu, Pengjuan Xie, and Linlin Li. 2019. Neural Chinese Address Parsing. In NAACL. 3421–3431.
[6] Xiaofei Liu, Da Yin, Xingjian Zhang, Kai Su, Can Wu, Hongxia Yang, and Jie Tang. 2021. OAG-BERT: Pre-train Heterogeneous Entity-augmented Academic Language Models. ArXiv abs/2103.02410 (2021).
[7] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and Robust Approximate Nearest Neighbor Search using Hierarchical Navigable Small World Graphs. In NeurIPS ’18, 35–43.
[8] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations. In NAACL. 746–751.
[9] Sinno Jialin Pan and Qiang Yang. 2009. A Survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 21, 10 (2009), 1345–1359.
[10] Matthew E. Peters, Mark Neumann, Mohit Iyyer, et al. 2018. Deep Contextualized Word Representations. In NAACL. 2227–2237.
[11] Chen Sun, Austin Myers, Carl Vondrick, et al. 2019. VideoBERT: A Joint Model for Video and Language Representation Learning. In ICCV. 7465–7472.
[12] Yibo Sun, Zhihu Huang, Chenyuan Yuan, Xiao Fan, Haofeng Wang, et al. 2021. GEDIT: Geographic-Enhanced and Dependency-Guided Tagging for Joint POI and Accessibility Extraction at Baidu Maps. In CIKM. 4135–4144.
[13] Lauren's van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. Journal of machine learning research 9, 11 (2008).
[14] Emily Alsentzer, John Murphy, et al. 2019. Publicly Available Clinical BERT Embeddings. In ClinicalNLP ‘19, 72–78.
[15] Diederik P. Kingma, and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In ICLR.