Joint Intent Detection and Entity Linking on Spatial Domain Queries

Lei Zhang[1,2]*, **Runze Wang**[3,2], **Jingbo Zhou**[2]†, **Jingsong Yu**[1], **Zhenhua Ling**[3], **Hui Xiong**[4]
[1] Peking University, [2] Business Intelligence Lab, Baidu Research,
[2] National Engineering Laboratory of Deep Learning Technology and Application, China,
[3] University of Science and Technology of China, [4] Rutgers University
1801210727@pku.edu.cn, wrz94520@mail.ustc.edu.cn, zhoujingbo@baidu.com,
yjs@ss.pku.edu.cn, zhling@ustc.edu.cn, hxiong@rutgers.edu

Abstract

Continuous efforts have been devoted to language understanding (LU) for conversational queries with the fast and wide-spread popularity of voice assistants. In this paper, we first study the LU problem in the spatial domain, which is a critical problem for providing location-based services by voice assistants, but is without in-depth investigation in existing studies. Spatial domain queries have several unique properties making them be more challenging for language understanding than common conversational queries, including lexical-similar but diverse intents and highly ambiguous words. Thus, a special tailored LU framework for spatial domain queries is necessary. To the end, a dataset *SMQ* was extracted and annotated based on the real-life queries from a voice assistant service. We then proposed a new multi-task framework MELIP that jointly learns the intent detection and entity linking tasks on the *SMQ* with invented hierarchical intent detection method and triple-scoring mechanism for entity linking. A specially designed spatial GCN is also utilized to model spatial context information among entities. We have conducted extensive experimental evaluations with state-of-the-art entity linking and intent detection methods, which demonstrated that MELIP can outperform all baselines with a significant margin.

1 Introduction

The past few years have witnessed the successful deployment of voice assistants on smart speakers (e.g. Amazon Echo) and mobile devices (e.g. Apple Siri and Google Assistant). As a critical step to facilitate informative responses by voice assistants, language understanding (LU) has attracted tremendous research attention in recent years (Wang et al., 2018; Chen et al., 2019; Haihong et al., 2019). LU typically includes the intent detection which detects the categorical intent label, and the slot filling which indicates the slot type mentioned by certain words (Zhang et al., 2019; Liu and Lane, 2016).

In this paper, we first investigate the LU problem in the spatial domain. With the continuous improvement of their intelligence, virtual assistants are designed to provide many location-based services such as recommending restaurants (Luo et al., 2020) and providing route planning (Chen et al., 2013). We name all such queries as spatial domain queries that usually contain some spatial information.

Similar to the LU of common conversational queries, there are also two main tasks for the LU of spatial domain queries. The first part is intent detection which aims to classify a user query into a scenario for further processing. An example of such intent can be "asking for the location information of POI". (Here POI refers to Point of Interest, which is a place on a map like a restaurant or a shop.) Table 1 shows some examples of query intent in spatial domain queries. The second part is entity linking (Fang and Chang, 2014; Sevgili et al., 2019) which aims to map potential ambiguous mentioned words (hereafter we name them as mentions) in a query to their corresponding entities in spatial Knowledge Bases (KBs) for providing relevant information and services. For example, for the queries "where is a place to play?" and "where is an interesting place?", we have to link both the "place to play" and "interesting place" to POIs with tag "entertainment venue", and return the corresponding POIs.

Building LU system for spatial domain queries has several unique challenges that have not been studied in-depth in previous works. At first, spatial

[*] Lei Zhang and Runze Wang contributed equally to the paper. This work was done when they were interns at the Baidu Research

[†] Jingbo Zhou is corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4937–4947
November 16 - 20, 2020. ©2020 Association for Computational Linguistics

Query Type	Mention Type	Query Intent
Ask for the distance information between two place	AROUND, AOI	Ask for the distant from AROUND to AOI
Ask for the time information between two place	CITY, PROVINCE	Ask for the time from CITY to PROVINCE
Ask for the location information of one place	POI	Ask for the location information of POI
Ask for the information of one place except location	POI	Ask for evaluation of POI
Ask for a recommendation	TAG	Ask for recommendations for attractions
Only one entity	POI	Ask for the information of POI

Table 1: The examples of defined query intents.

domain queries usually have lexical-similar but diverse intents. For example, the query "How far is it from here to Beijing Gymnasium?" and query "How far is it from here to Beijing's gymnasium?" are almost the same (with only one word difference), but their intents are totally different (the first intent is the distance from here to a place, and the second one is the distance from here to a tag, i.e. gymnasium). The intents of spatial domain queries are enormous, and the actual intent is conditioned on the type of entity referenced in a query.

Second, the entity linking in spatial domain queries is also a challenging problem. The mentions in spatial domain queries are quite diverse and ambiguous. For example, "Juqi" is a common dialect word in Beijing, but it also refers to a famous restaurant brand; "Braised Chicken Rice" is a popular food in China, but it is also the name of many bistros. Moreover, the candidate entities even share the same surface names. For example, there are two "Xinhua Garden" in Beijing and many ones in China. How to correctly distinguish and link the entities is a challenging problem.

To tackle the above challenges, in this paper, we propose a novel model MELIP tailored for language understanding of spatial domain queries, with working on a human-labeled real-life spatial domain query dataset *SMQ* (short for spatial domain queries). The core of MELIP is a multi-task learning framework that jointly learns the main tasks of the intent detection and entity linking. To overcome the lexical-similar but diverse intent challenge, we propose a hierarchical intent detection method with two auxiliary tasks which are *query type* prediction and *mention type* prediction. The query type task classifies each query into seven types as shown in Table 2, and the *mention type* task classifies all mentions referenced in each query into the ten types shown in Table 3. The final query intent detection model is built on top of *query type* and *mention type* tasks. For handling the challenge of entity linking, we propose a triple-scoring mechanism to distinguish candidate entities. In addition, to encode the spatial context information, we conduct a spatial graph convolutional network (SGCN)(Vashishth et al., 2019) to model the relationship between entities by pre-training the entity's embedding vectors. The query intent detection module and entity linking module interact with each other by jointly training and sharing knowledge in MELIP.

SMQ is a real-life spatial domain query dataset collected from DuerOS[1], one of the largest voice assistant services in China. *SMQ* has 55,000 pieces of spatial domain queries with human-labeled ground truth. We have conducted extensive experiment evaluation with the state-of-the-art query intent detection and entity linking methods on *SMQ*, and the results show that MELIP can significantly better handle these two tasks.

We summarize our contributions as follows:

- We first study the LU problem for spatial domain queries on a real-life dataset collected from a voice assistant services.
- We propose a multi-task framework MELIP to jointly train the entity linking and query intent detection tasks on spatial domain queries.
- We conducted extensive experimental evaluations to demonstrate the effectiveness of the proposed framework.

2 Related Works

2.1 Intent Detection

Intent detection task aims to classify the intent of queries and is always considered as a text classification task (Kim, 2014; Lai et al., 2015; Yang et al., 2016; Joulin et al., 2017; Xia et al., 2018). Considering the complexity of the label, some hierarchical text classification methods(Huang et al., 2019; Mao et al., 2019; Aly et al., 2019) have emerged to capture label hierarchies. Recently there are some joint models to jointly learn the intent detection and

[1] https://dueros.baidu.com/en/index.html

Type index	Query Type	Example
0	Ask for the distance information between two places	从上海到北京多少公里
1	Ask for the information between two places except distance and time	从上海到北京最近线路
2	Ask for the time information between two places	从上海到北京要多长时间
3	Ask for the location information of one place	上海市的准确位置在哪里
4	Ask for the information of one place except location	上海的土地面积
5	Ask for a recommendation	上海有哪些景点
6	Only one entity	上海迪士尼酒店

Table 2: The defined seven query types and their examples.

Type index	Mention Type	Type index	Mention Type
0	POI	5	BRAND
1	AREA	6	PROVINCE
2	AOI	7	AROUND
3	TAG	8	TIME
4	CITY	9	PERSON GROUP

Table 3: The defined ten mention types.The meaning of each type can be found in section 3.1

slot filling. Goo et al. (2018) utilized a slot-gated mechanism as a special gate function to model the relationship between the slot filling and intent detection. Qin et al. (2019) directly incorporated the token-level intent information for slot filling with a stack-propagation mechanism. Zhang et al. (2019) proposed a capsule-based neural network model which completed slot filling and intent detection. However, there are no existing studies about intent detection on spatial domain queries.

2.2 Entity Linking

Entity linking, which maps potentially ambiguous mentions in the text to their corresponding entities in KBs, is a fundamental but important stage in many text understanding tasks.Previous works usually focused on long well-formed texts, such as news or articles (Ganea and Hofmann, 2017; Nie et al., 2018; Le and Titov, 2018; Zhou et al., 2019; Yang et al., 2019; Martins et al., 2019; Sakor et al., 2019).

To the best of our knowledge, there are no existing studies to handle the spatial entity linking problem. Whereas, we propose a triple-scoring mechanism and spatial graph convolutional network (SGCN) for spatial entity linking. By jointly using the query intent detection task, the final entity linking performance was further improved.

3 Preliminary

3.1 Basic Notations

The point-of-interest (POI) knowledge base (POI-KB) is used as our knowledge base for entity linking, which contains nearly 60 million entities with several spatial relevant descriptions. POI is a special entity which means a certain point on the map. The description of a POI can be listed as $\{ID, NAME, TAG, PROVINCE, CITY, AREA, AOI, \ldots\}$, where TAG is the type of a POI and each POI may have multiple TAGs. For example, the TAG of KFC is *gourmet food* and *fast food restaurant*. The $AREA$ refers to the county or prefecture-level city of the POI. AOI (area of interest) is the geographical range of an entity below the entity's $AREA$ and above POI. $PROVINCE$, $CITY$, $AREA$, and AOI are all location descriptions of the entities in the POI-KB.

3.2 Task Description

The main task in this paper is to parse user queries with intent detection task and entity linking task. Denoting the user query as $q = \{w_0, w_1, \ldots, w_i, \ldots, w_{N_q}\}$, the candidate intent set as $\mathbb{I} = \{qi_0, \ldots, qi_{N_{qi}}\}$, the query intent detection task can be formulated as:

$$\{q, \mathbb{I}\} \xrightarrow{predict} qi_j, \tag{1}$$

We assume each q contains N_m mentions $M = \{m_0, \ldots, m_i, \ldots, m_{N_m}\}$, and the entity linking task is to mapping each mention to an entity from its candidate entity set $\mathbb{C}_i = \{e_0, \ldots, e_{N_e}\}$. The entity linking task can be formulated as:

$$\{q, M, \{\mathbb{C}_i\}_{i \in \{1, \cdots, N_m\}}\} \xrightarrow{map} \{e_j\}_{j \in \{1, \cdots, N_m\}}. \tag{2}$$

Besides, we also propose two auxiliary tasks to help the two main tasks above query type prediction and mention type prediction. We assume that each query q has a query type $qt \in \mathbb{T}_q = \{qt_0, \ldots, qt_{N_{qt}}\}$, each mention m_j in q has a mention type $mt_j \in \mathbb{T}_m = \{mt_0, \ldots, mt_{N_m}\}$. The query type prediction can be formulated as:

$$\{q, \mathbb{T}_q\} \xrightarrow{predict} qt, \tag{3}$$

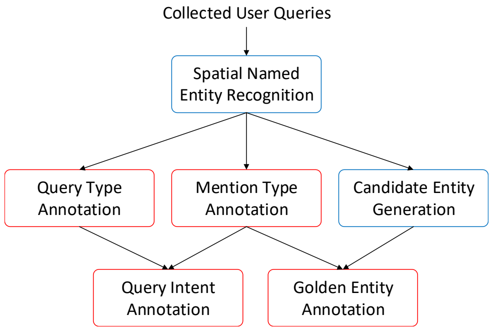


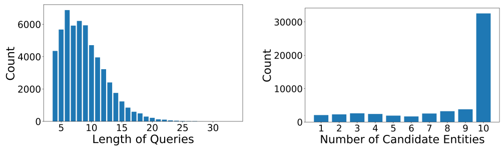
Figure 1: The generation process of our *SMQ*

and the mention type prediction can be formulated as:

$$\{q, M, \mathbb{T}_m, \} \overset{predict}{\longrightarrow} \{mt_j\}_{j\in\{1,\cdots,N_m\}}. \tag{4}$$

These two auxiliary tasks can improve the two main tasks performance and we will describe it in the following section.

4 Corpus Construction



(a)

(b)

Figure 2: The data statistics on *SMQ*.

All queries in *SMQ* are collected from DuerOS, one of the largest voice assistant service provider in China. As illustrated in Figure 1, we develop *SMQ* dataset with six processes. The processes with red boxes are all accomplished by trained annotators. The processes with blue boxes are finished with the help of some algorithms and tools. We will describe these blue processes in detail as follows.

4.1 Spatial Named Entity Recognition

The spatial named entity recognition (Spatial-NER) is performed on each query to extract potential mentions. We first use an enterprise named entity recognition tool (Jiao et al., 2018).[2] It recognizes all spatial mentions in the input query. Meanwhile, due to the name diversity of POI, we also build a Trie-Tree with entity names in POI-KB, and match potential spatial mentions in the query with maximum prefix matching. The final set is the union set of the two spatial mention sets above.

4.2 Candidate Entity Generation

For entity linking task, we first need to generate candidate entities for each mention in the query. A Synonyms Tool(Hai Liang Wang, 2017) is used to find all synonyms for each mention in each query Then, we use three methods to filter all the entities in the POI-KB conditioned on each mention and its synonyms: string-match-based method (SM-based method), edit-distance-based method (ED-based method) and word-embedding-based method (WE-based method).

The SM-based method selects all entities whose surface name is a sub-string of the given mention or its synonyms. The ED-based method calculates the edit distance (ED)(Yujian and Bo, 2007) between all entities in POI-KB and the given mention or its synonyms. To accelerate the calculation of ED, we filter all candidates if the length of one text is more than twice as long as another text. The WE-based method first converts entities, mentions, and mentions synonyms into high-dimensional vectors. Then we calculate the similarity with the dot-product mechanism between each entity and a given mention or its synonyms. These candidate entities generated by the three methods are collected together and then fed into the next spatial filter.

Furthermore, a spatial filter is conducted on candidate entities to filter entities that have no spatial relationship to the input query. Here we define a *query location* attribute for each query. If a query contains one or more places, its *query location* is the location attributes set of these places. If a query has no place, the location where the user asked this query is considered as its *query location*. We only keep the candidate entities that have the same locations as the *query location*.

4.3 Dataset Statistics

We summarize the statistics of *SMQ* in Figure 5. In Figure 2(a), we can see that the query length is very short whose average length is only 8. The candidate entity number of mentions in each query is shown in Figure 2(b) whose average number is 8.12. Besides, in *SMQ*, the number of query intents is 100, the number of query types is 7, and the number of mention types is 10. More statistics information can be found in the Appendix 8.4.

5 Model Architecture

We illustrate the proposed multi-task framework MELIP in Figure 3, consisting of a query intent

[2] https://github.com/baidu/lac

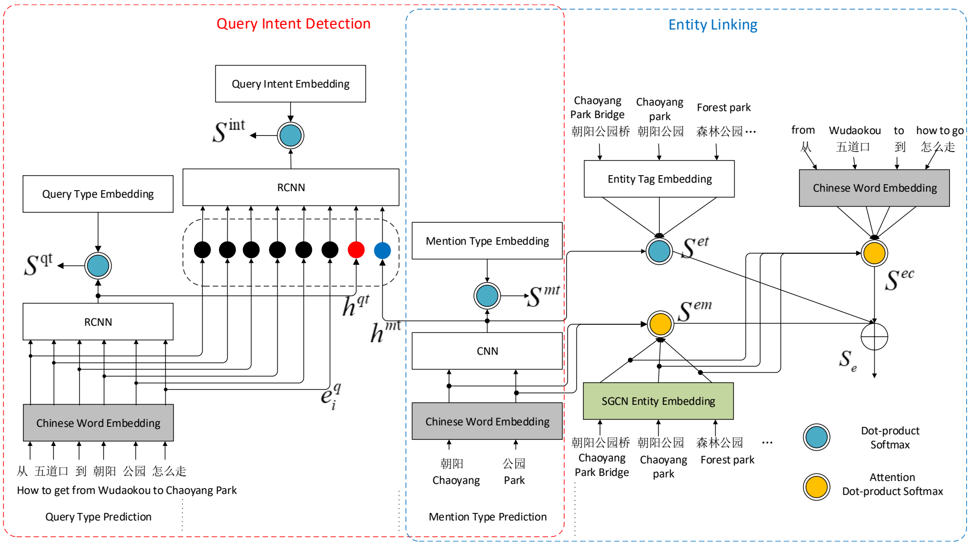


Figure 3: The architecture of the proposed multi-task framework MELIP. It consists of two main tasks(query intent detection and entity linking) and tow auxiliary tasks(query type prediction and mention type prediction).

detection module and an entity linking module in the POI-KB. Query intent detection module is supported by two auxiliary tasks: query type prediction task and mention type prediction task. Mention type prediction task also supports entity linking task. To sum up, MELIP is a multi-task framework with four tasks: two main tasks and two auxiliary tasks. Hereafter, we use superscript of qt to denote query type, m to mentions, mt to mention type, int to query intent and el to entity linking.

We record $L^{int}, L^{el}, L^{qt}, L^{mt}$ as the loss functions for query intent detection task, entity linking task, query type prediction task and mention type prediction task respectively. The final loss function is defined as follows.

$$L = \lambda_1(L^{int} + L^{el}) + \lambda_2(L^{qt} + L^{mt}), \qquad (5)$$

We jointly trained MELIP through the above loss function. Next, we will describe the two main tasks and how the two auxiliary tasks support them.

5.1 Hierarchical Intent Detection

To handle the lexical-similar but diverse intent problem in spatial domain queries, we design a hierarchical classification structure to detect query intent. It utilizes the hierarchical relationship among query type, mention type and query intent to obtain the final intent, which is illustrated in Figure 4(a).

Given a user query q, it is first divided into word sequences by a word segmentation tool *jieba*[3]. Then, each word sequence is fed into a pre-trained word embedding module (Li et al., 2018; Qiu et al., 2018) to generate a query word embedding vector $\boldsymbol{e}_i^q$, where i stands for the *i-th* word sequence in the query q and $1 \leq i \leq |q|$.

As shown in Query Intent Detection part of Figure 3 (left), the input of query intent detection module contains three parts: query word embedding vectors $\{\boldsymbol{e}_i^q\}$, hidden representation $\boldsymbol{h}^{qt}$ of the query (red dot) from the query type prediction module (see Section 5.1.1) and hidden representation $\boldsymbol{h}^{mt}$ of mentions (blue dot) from the mention type prediction module (see Section 5.1.2). The combined input is fed into a RCNN module to generate the final hidden state h^q of query q $h^q = RCNN([e_1, ..., e_i^q, ...; h^{qt}]; h^{mt})$. We also embed each query intent into a low dimensional space and denote it as $\{\boldsymbol{v}_j^{int}\}_{j \in (0,1,...,N^{int})} \in \mathbb{R}^d$, where d is the hidden dimension size and N^{int} is the number of query intents. Then the prediction score of query q to be *j-th* query intent is calculated with dot-product and softmax as follows:

$$S_j^{int} = \frac{\exp(\boldsymbol{v}_j^{int}\boldsymbol{h}^q)}{\sum_j^{N^{int}} \exp(\boldsymbol{v}_j^{int}\boldsymbol{h}^q)}, \qquad (6)$$

The loss function L^{int} are calculated to the cross entropy on $\{S_j^{int}\}$. Next, we will describe in detail two auxiliary tasks.

5.1.1 Query Type Prediction

The query type prediction model aims to classify the query into different query types as shown in Table 2. Then embedded query vector $\{\boldsymbol{e}_i^q\}_{1 \leq i \leq |q|}$ of query q is fed into an RCNN(Lai et al., 2015)

[3] https://github.com/fxsjy/jieba

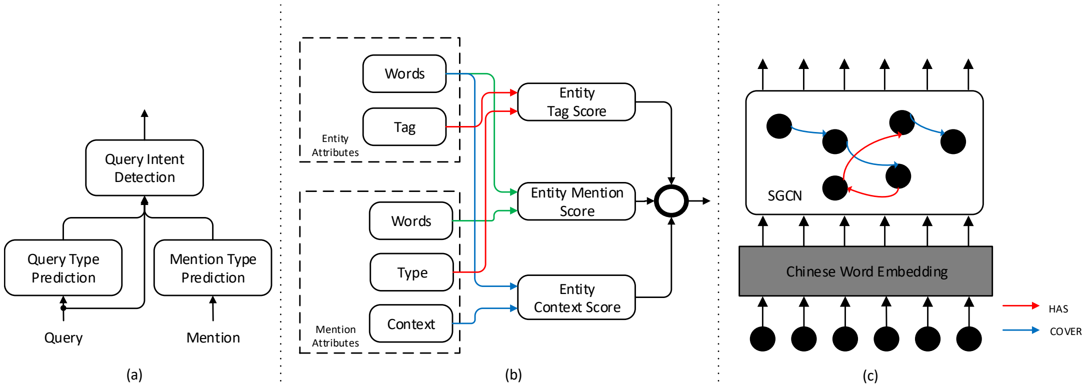


Figure 4: (a): The hierarchical intent detection structure. (b): The triple-scoring mechanism for entity linking. (c): The spatial GCN-based model for pre-training entity embedding.

module to generate the hidden representation $\boldsymbol{h}^{qt}$ for query type prediction. The query type embedding module generates a set of vectors $\{\boldsymbol{v}_i^{qt}\}_{i\in(0,1,\ldots,6)} \in \mathbb{R}^d$, where each vector stands for one query type representations (the total number of query types is 7). Finally, the prediction score for *i-th* query type S_i^{qt} is calculated by dot-product between $\boldsymbol{v}_i^{qt}$ and $\boldsymbol{h}^{qt}$ with softmax, similar with Eqn. 6. The loss function L^{qt} is set to the cross entropy on $\{S_i^{qt}\}$.

5.1.2 Mention Type Prediction

The objective of mention type prediction is to classify each mention (recognized by spatial NER introduced in Section 4.1) into 10 mention types as shown in Table 3. At first, each mention in a query is first divided into a sequence of words. The mention word vectors $\{\boldsymbol{e}_i^m\}$ are generated after the same pre-trained word embedding module used in query type prediction module. Then a CNN module is used to output the final hidden representation $\boldsymbol{h}^{mt}$ for the mention. Similar with query type prediction, the mention type embedding module also generates a set of vectors $\{\boldsymbol{v}_i^{mt}\}_{i\in(0,1,\ldots,10)} \in \mathbb{R}^d$, where each vector stands for one mention type representation. The mention type score $\{S_i^{et}\}$ for each mention on *i-th* type is calculated by dot-product between $\boldsymbol{v}_i^{mt}$ and $\boldsymbol{h}^{mt}$ with softmax, similar with Eqn. 6. The loss function L^{mt} is set to the cross entropy on $\{S_i^{mt}\}$.

As illustrated in Figure 3, the hidden representation $\boldsymbol{h}^{mt}$ of mention is also fed to query intent detection module as part of the input. It is worth noting that many queries may have several mentions. When training, we flatten the training queries by their mentions. If a query has two mentions, the query intent detection module will learn it twice conditioned on the different mentions. During the test, we record all the query intent scores conditioned on every mention and calculate their average score as the final query intent detection score:

$$\tilde{S}_j^{int} = \sum_k^{N_m} S_{jk}^{int}, \tag{7}$$

where N_m stands for the mention number in the query and S_{jk}^{int} is calculated with Eqn. 6

5.2 Entity Linking

As shown in Figure 4(b), the entity linking module utilizes a triple-scoring mechanism to calculate similarity scores between each mention and its candidate entities. These three scores are entity-tag score S^{et}, entity-mention score S^{em}, and entity-context score S^{ec}. Next, we will describe them.

Considering that entities are quite diverse and ambiguous, and entities with the same name may belong to different tags. Therefore, we will make full use of the tag information of each mention and candidate entities to overcome the ambiguity of entities. We calculate the similarity score between mention type and each candidate entity tag. This similarity is called entity-tag score S^{et}. As illustrated in Figure 3, the entity tag embedding module first generates the tag attribute of each candidate entity from the POI-KB. Then a pre-trained Chinese word embedding module is conducted to convert each entity TAG into a high-dimensional vector $\boldsymbol{e}_t^e$. To avoid error propagation, we use $\boldsymbol{h}^{mt}$ as the mention type embedding to calculate the entity-tag score instead of using the embedding vector of the predicted mention. S^{et} is calculated with the same as Eqn. 6 on $\boldsymbol{h}^{mt}$ and $\boldsymbol{e}_t^e$.

To estimate the similarities between candidate entities and mentions, the entity-mention score S^{em} is calculated between mention word vectors and entity embedding vectors. The mention word vectors are extracted from mention type prediction module, and the entity embedding vectors are generated from a spatial GCN-based (SGCN) entity embedding module which is specially designed for spatial domain entities. SGCN entity embedding module will be described in the next section. The entity-mention score S^{em} is calculated by the model proposed by Le et. al. (Le and Titov, 2018), which is designed with **Attention**, **Dot-product** and **Softmax** for calculating entity linking scores.

Finally, we consider the mention of contextual information with each candidate entity. The mention contextual context is defined as the original query word sequence but without the mention words. The same Chinese word embedding module as above is used to generate the context word vectors. Then, we calculate the entity-context score S^{ec} between the context word vectors and candidate entity embedding vectors by the model used for entity-mention score(Le and Titov, 2018).

The entity linking result is selected based on the average of S^{ec}, S^{em}, S^{et}, and the loss function L^{el} is also conducted to cross entropy on the final average score.

5.2.1 SGCN Entity Embedding

The SGCN entity embedding module is conducted to model the spatial information between candidate entities and help distinguish them. It is initialized with a pre-trained entity vector set that is generated from a pre-training model shown in Figure 4(c). The input in this pre-training model contains all entities in POI-KB, all of their tags, and their locations. Two edges are defined between the input items for the following GCN.

- HAS: if an entity has a tag, then this entity and tag have a HAS edge.
- $COVER$: if an entity covers another entity in spatial, then these two entities have a COVER edge. The $COVER$ relation is generated by entity location attributes.

The entity graph with two edges will be fed into a two-layer GCN to generate the final pre-trained entity embedding. It should be noticed that we generate the entity embedding with all the POI-KB attributes, but we only use the entity embedding to the modules. The entity tag embedding and entity location embedding are abandoned, but they have been taken into account into entity embedding.

6 Experiments

6.1 Setting

We divide the queries in *SMQ* into 44,000 for training, 5500 for validation, and 5500 for testing. It took 2.5 hours of training on the Tesla P100 GPU. At test time, we evaluated the four tasks using the percentage of correctly predicted queries(i.e. Accuracy). It was worth noting that when we tested entity linking performance, we removed these mentions that are of type $PROVINCE$, $CITY$, and $AREA$. It was because almost all the tested models achieved nearly 100% accuracy on these types and they can not help us evaluate the models. For detailed information about model configuration and parameter settings, please refer to the Appendix 8.1.

6.2 Baselines

We compared MELIP with the following baselines on *SMQ*. For query intent detection task, the text classification of **FastText** (Joulin et al., 2017), **CNN** (Kim, 2014) and **RCNN** (Lai et al., 2015) are evaluated. In addition, The Bidirectional Encoder Representation from Transformer(**BERT**)(Devlin et al., 2018) has achieved amazing results in many language understanding tasks including text classification. Therefore, BERT and BERT+RCNN (A fine-tuning method upon BERT) are evaluated.

Given that MELIP is also a multi-task model for intent detection, We compare our model with the existing state-of-the-art joint learning intent detection baselines:

- **Slot-Gated Atten** (Goo et al., 2018) proposed the slot-gated joint model to explore the correlation of slot filling and intent detection better.
- **Stack-Propagation**(Qin et al., 2019) adopted a joint model with Stack-Propagation which can directly incorporate the token-level intent information for slot filling, thus to capture the intent semantic knowledge.

For entity linking task, the following state-of-the-art baselines were compared:

- **MLR** (Le and Titov, 2018) is an advanced entity linking model on long well-formed context. We conducted it on *SMQ* by using our pre-trained entity embedding vectors as the initialization of entity embedding in it. The prior probability of each candidate entity was calculated as edit distance here.

Task	Model	Accuracy(%)
Query Intent detection	FastText	50
	CNN	76.38
	RCNN	77.33
	BERT	80.47
	BERT+RCNN	81.88
	Slot-Gated Atten	70.76
	Stack-Propagation	78.47
	MELIP	**83.20**
Entity Linking	MLR	67.75
	DCA	76.30
	MELIP	**89.37**
Mention Type Prediction	Fasttext	67.8
	CNN	80.63
	RCNN	80.47
	MELIP	**92.27**
Query Type Prediction	Fasttext	94
	CNN	95
	RCNN	95.6
	MELIP	**96.4**

Table 4: The Results on *SMQ* of different models.

- **DCA** (Yang et al., 2019) improves the MLR model. Here we experimented with it in the same way with MLR.

6.3 Results

We report the overall results of different query intent detection models and entity linking models on *SMQ* test data in Table 4. As we can see, for query intent detection task, MELIP achieved the best performance among all baselines whose accuracy is 83.20%. For the entity linking task, we evaluated MELIP with two baselines on *SMQ*. MLR (Le and Titov, 2018) and DCA (Yang et al., 2019) are both the best methods for entity linking on the AIDA CoNLL-YAGO entity linking benchmark dataset (Hoffart et al., 2011). We trained them using *SMQ* and their accuracy is only 67.75% and 76.30% respectively. Our MELIP achieved an accuracy of 89.37%, which is 13.07% higher than the DCA performance. Moreover, As we can see in table 4, our multi-task framework MELIP can also improve the performance of the two auxiliary tasks.

6.4 Ablation

To demonstrate the effectiveness of MELIP in joint learning query intent detection and entity linking tasks, we also report the ablation results in Table 5. For query intent detection task, we first removed all entity linking modules from MELIP, leaving only query type prediction, mention type prediction and query intent detection modules. The result shown in Table 5 was 2.30% less than entire MELIP. This proved that entity linking task can improve the performance of query intent detection.

Task	Model	Accuracy(%)
Query Intent detection	MELIP	83.20
	-EL	80.90(-2.30)
	-EL&MT	79.40(-3.80)
	-EL&QY	80.10(-3.10)
	-EL&MT&QY	77.93(-5.27)
Entity Linking	MELIP	89.37
	-QI	86.00(-3.37)
	-QI & MT	84.60(-4.77)

Table 5: Ablation Results for query intent detection and entity linking."EL" means entity linking task, "QI" means query intent detection task, "MT" means mention type prediction task and "QY" means query type prediction task.

Then, to prove the effectiveness of the two auxiliary tasks, we also removed them in order from the query intent detection module. When removing mention type prediction module, the result is shown in Table 5 was reduced by 3.80%. Similarly, after removing query type prediction model, the performance dropped to 80.10%. Furthermore, we removed both of the two auxiliary tasks and the result showed an accuracy of 77.93%. The above ablation researches prove that all tasks are a benefit to query intent detection task.

For entity linking task, after removing the query intent detection modules from MELIP, the accuracy declined 3.37% compared to the entire model. The two auxiliary tasks can also help to improve the entity linking task. As shown in Table 5, after removing mention type prediction module but provided the golden mention type embedding to calculate the entity-type score in the entity linking task, the accuracy of the model (being 84.60%) declined 4.77% compared to the best model.

7 Conclusion

In this paper, we study the language understanding problem on real-life spatial domain queries. We proposed a hierarchical intent detection method to overcome the lexical-similar but diverse intent challenge. We also designed a triple-scoring solution to entity linking from the diverse and ambiguous query words. Considering the interaction between query intent detection and entity linking, a multi-task framework MELIP is designed for jointly learning the two main tasks and two auxiliary tasks. The performance of MELIP on a large scale dataset *SMQ* is significantly better than the state-of-the-art models.

References

Rami Aly, Steffen Remus, and Chris Biemann. 2019. Hierarchical multi-label classification of text with capsule networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics(ACL)*.

Gang Chen, Sai Wu, Jingbo Zhou, and Anthony KH Tung. 2013. Automatic itinerary planning for traveling services. *IEEE transactions on knowledge and data engineering*, 26(3):514–527.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yuan Fang and Ming-Wei Chang. 2014. Entity linking on microblogs with spatial and temporal signals. *Transactions of the Association for Computational Linguistics(TACL)*.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep Joint Entity Disambiguation with Local Neural Attention. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing(EMNLP)*.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics(NAACL)*.

Hu Ying Xi Hai Liang Wang. 2017. Chinese Synonym Toolkit Synonyms.

E Haihong, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics(ACL)*.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust Disambiguation of Named Entities in Text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing(EMNLP)*.

Wei Huang, Enhong Chen, Qi Liu, Yuying Chen, Zai Huang, Yang Liu, Zhou Zhao, Dan Zhang, and Shijin Wang. 2019. Hierarchical Multi-label Text Classification: An Attention-based Recurrent Network Approach. In *Proceedings of ACM International Conference on Information and Knowledge Management(CIKM)*.

Zhenyu Jiao, Shuqi Sun, and Ke Sun. 2018. Chinese lexical analysis with deep bi-gru-crf network. *arXiv preprint arXiv:1807.01882*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the European Chapter of the Association for Computational Linguistics(EACL)*.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of the National Conference on Artificial Intelligence(AAAI)*.

Phong Le and Ivan Titov. 2018. Improving Entity Linking by Modeling Latent Relations between Mentions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics(ACL)*.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. 2018. Analogical reasoning on chinese morphological and semantic relations. *arXiv preprint arXiv:1805.06504*.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.

Hui Luo, Jingbo Zhou, Zhifeng Bao, Shuangli Li, J Shane Culpepper, Haochao Ying, Hao Liu, and Hui Xiong. 2020. Spatial object recommendation with hints: When spatial granularity matters. In *SIGIR*, pages 781–790.

Yuning Mao, Jingjing Tian, Jiawei Han, and Xiang Ren. 2019. Hierarchical Text Classification with Reinforced Label Assignment. *arXiv preprint arXiv:1908.10419*.

Pedro Henrique Martins, Zita Marinho, and André FT Martins. 2019. Joint Learning of Named Entity Recognition and Entity Linking. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics(ACL)*.

Feng Nie, Shuyan Zhou, Jing Liu, Jinpeng Wang, Chin-Yew Lin, and Rong Pan. 2018. Aggregated semantic matching for short text entity linking. In *Proceedings of the Conference on Computational Natural Language Learning(CoNLL)*.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A Stack-Propagation Framework with Token-Level Intent Detection for Spoken Language Understanding. *arXiv preprint arXiv:1909.02188*.

Yuanyuan Qiu, Hongzheng Li, Shen Li, Yingdi Jiang, Renfen Hu, and Lijiao Yang. 2018. Revisiting correlations between intrinsic and extrinsic evaluations of word embeddings. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data(CCL)*.

Ahmad Sakor, Isaiah Onando Mulang, Kuldeep Singh, Saeedeh Shekarpour, Maria Esther Vidal, Jens Lehmann, and Sören Auer. 2019. Old is gold: linguistic driven approach for entity and relation linking of short text. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics(NAACL)*.

Özge Sevgili, Alexander Panchenko, and Chris Biemann. 2019. Improving Neural Entity Disambiguation with Graph Embeddings. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics(ACL)*.

Shikhar Vashishth, Manik Bhandari, Prateek Yadav, Piyush Rai, Chiranjib Bhattacharyya, and Partha Talukdar. 2019. Incorporating Syntactic and Semantic Information in Word Embeddings using Graph Convolutional Networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics(ACL)*.

Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi-model based rnn semantic frame parsing model for intent detection and slot filling. *arXiv preprint arXiv:1812.10235*.

Yuan Xia, Jingbo Zhou, Jingjia Cao, Yanyan Li, Fei Gao, Kun Liu, Haishan Wu, and Hui Xiong. 2018. Intent-aware audience targeting for ride-hailing service. In *ECML/PKDD*, pages 136–151. Springer.

Xiyuan Yang, Xiaotao Gu, Sheng Lin, Siliang Tang, Yueting Zhuang, Fei Wu, Zhigang Chen, Guoping Hu, and Xiang Ren. 2019. Learning Dynamic Context Augmentation for Global Entity Linking. *arXiv preprint arXiv:1909.02117*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the conference of the North American chapter of the association for computational linguistics(NAACL)*.

Li Yujian and Liu Bo. 2007. A normalized Levenshtein distance metric. In *IEEE transactions on pattern analysis and machine intelligence(IEEE)*.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and S Yu Philip. 2019. Joint Slot Filling and Intent Detection via Capsule Neural Networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics(ACL)*.

Jingbo Zhou, Shan Gou, Renjun Hu, Dongxiang Zhang, Jin Xu, Airong Jiang, Ying Li, and Hui Xiong. 2019. A collaborative learning framework to tag refinement for points of interest. In *KDD*, pages 1752–1761.

8 Appendix

8.1 Setting

When training, the queries were further flattened with the way we described in section 5.2.2. We record $L^{int}, L^{el}, L^{qt}, L^{mt}$ as the loss functions for query intent detection task, entity linking task, query type prediction task and mention type prediction task respectively. The final loss function is defined as follows.

$$L = \lambda_1(L^{int} + L^{el}) + \lambda_2(L^{qt} + L^{mt}), \tag{8}$$

We jointly trained the two main task and two auxiliary task with the loss function Eqn. 8. λ_1 was 1 and λ_2 was 0.6. The hidden state size d is 300 for all CNN and RCNN modules. The Chinese word embedding modules were all initialed with Word2Vec(Li et al., 2018; Qiu et al., 2018). The GCN layer number in the SGCN pre-training model was 2. The N_{qi} was 100. The learning rate was set as 0.001 for RCNN, 0.0001 for CNN in query intent detection module and 0.01 for entity linking module. All the parameters were optimized with Adam optimizer(Kingma and Ba, 2014) and the batch size was 16. We trained the model with 20 epochs and an early stop mechanism was used when the accuracy on the validation set did not increase over ten batches. The hyper-parameters were evaluated on validation results.

8.2 Analysis

In order to further study the ability of MELIP on different query types, we divided test dataset into seven groups by query type. Then, we tested query intent detection and entity linking performances on them. The results are shown in Table 6.

For the query intent detection task, we can easily find that MELIP has the best performance on query type 6. This is because query type 6 is easier than other types and we also generated more data on it. For query type 2 and 4, there are less training and testing data on them. More data should be extracted on them to improve the MELIP performance. For query types 1, 3 and 5, we believe our MELIP could solve them well, with an accuracy close to 80%. For query type 0 with an accuracy of 57.23%, the worst performance is mainly because it is harder than other types. We will focus on dealing with it in our future work.

For the entity linking task, the accuracy of all query types is higher than 85%, which means that

Query Type	0	1	2	3	4	5	6
Train Number	4352	11364	1062	4872	102	4657	17590
Test Number	352	1482	72	766	24	498	2305
QI Acc.(%)	57.23	80.75	69.38	83.21	80.33	79.77	89.48
EL Acc.(%)	85.07	87.82	91.00	87.89	86.53	95.40	89.96

Table 6: .The results of query intent detection and entity linking on different query types."QI" means query type prediction task while "EL" means entity linking task. The query type index is the same as Table 2

our MELIP has the powerful ability to handle entity linking task in the spatial domain.

8.3 Dataset Annotation

As illustrated in Figure 1, we develop *SMQ* dataset with six processes. We have explained these blue processes. Now, we will describe these res processes in detail as follows.

Query Type Annotation This step marks each query as one of the query types described in Table 2. We sent the query to three trained annotators to accomplish this task. We consider this query a valid query only if more than two annotators have labeled the same type for the same query. Queries labeled for different types by three annotators are discarded. We also gave up those queries that could not be classified as one of the seven query types.

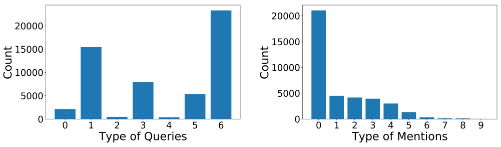
Mention Type Annotation Now, we annotate each mention generated from spatial named entity recognition as one of the ten mention types shown in Table 3. Three trained annotators are employed for this work and the annotation rules are the same with the query type annotation. Those mentions that do not fall into one of the ten mention types will be considered as common words in the query.

Query Intent Annotation After annotating all query types and mention types, we provide these results to three annotators to annotate the final query intent. The query intent is combined with query type and mention type with some easy rules. Some examples of query intent are shown in Table 1. The query intent annotation rules are the same as we described above. However, after labeling all queries, we will only keep the first 100 query intents in the order of their corresponding query numbers. Those query intents with fewer queries will be discarded.

Golden Entity Annotation In the last step, we will annotate the golden entity of each mention corresponding to. Three trained annotators are employed to do this work and the generation rules are the same as above. Besides the original query, mentions and candidate entities, annotators are provided with more entity attributes to help them distinguish candidate entities. Finally, each mention in the query will be labeled to a certain candidate entity as its corresponding entity in the POI-KB.

8.4 Dataset Statistics

We summarize the more statistics of *SMQ* in Figure 5. From Figure 5(a), we can find that the "only one entity" query has the highest weight. This is because many users only ask a simple entity as a query. In Figure 5(b), the mention type POI has the highest weight. This is the characteristic of spatial domain data that usually contains some special entity that is a certain point on the map.



(a) (b)

Figure 5: The more data statistics on *SMQ*. The type indexes in (a) and (b) are the same as Table 2 & 3