Point-of-Interest Oriented Question Answering with Joint Inference of Semantic Matching and Distance Correlation

Yifei Yuan
CUHK
HongKong SAR
yfyuan@se.cuhk.edu.hk

Jingbo Zhou
Baidu Inc.
China
zhoujingbo@baidu.com

Wai Lam
CUHK
HongKong SAR
wlam@se.cuhk.edu.hk

Abstract

Point-of-Interest (POI) oriented question answering (QA) aims to return a list of POIs given a question issued by a user. Recent advances in intelligent virtual assistants have opened the possibility of engaging the client software more actively in the provision of location-based services, thereby showing great promise for automatic POI retrieval. Some existing QA methods can be adopted on this task such as QA similarity calculation and semantic parsing using pre-defined rules. The returned results, however, are subject to inherent limitations due to the lack of the ability for handling some important POI related information, including tags, location entities, and proximity-related terms (e.g. "nearby", "close"). In this paper, we present a novel deep learning framework integrated with joint inference to capture both tag semantic and geographic correlation between question and POIs. One characteristic of our model is to propose a special cross attention question embedding neural network structure to obtain question-to-POI and POI-to-question information. Besides, we utilize a skewed distribution to simulate the spatial relationship between questions and POIs. By measuring the results offered by the model against existing methods, we demonstrate its robustness and practicability, and supplement our conclusions with empirical evidence.

1 Introduction

Point-of-Interest (POI) oriented question answering (QA) problem is a special QA task which aims to answer users' questions by generating a list of POIs. With the rapid development of smart agents (e.g. Amazon Echo) and intelligent virtual assistants (e.g. Apple Siri and Google Assistant), there are many POI oriented queries being requested everyday. Some examples of these questions are "Where can I take my kid to have fun nearby New York City" or "Where can we go in LA with my friends". The answers are typically a list of POIs such as parks, malls, or restaurants corresponding to the details provided by users in the questions. According to some statistics, there are millions of POI oriented QA questions being requested per day on a mobile search engine in China.

Generally speaking, semantic parsing and similarity matching methods are utilized to tackle the POI oriented QA problem in current solutions. Nevertheless, both of them are subject to inherent limitations and deserve to be improved. Semantic parsing based methods convert the questions to formal representations (such as SQL queries) using pre-defined rules, then get the POI results from the query. Difficulties arise, however, when the form of the questions varies from person to person. Besides, due to ambiguous expressions of a specific tag, semantic parsing methods always fail to match mentioned tags to the POI database. Furthermore, based only on tag information, it is almost impossible for semantic parsing methods to make use of the distance correlation between questions and POIs.

Another line of solution is adopting similarity matching models for calculating the similarity score between questions and POIs. Recent years have witnessed rapid growth in various kinds of semantic similarity based QA systems such as Convolutional Neural Network Architecture (Hu et al., 2014), LSTM Based Answer Selection (Tan et al., 2015, 2016), and Cross-Attention Based Question Answering System (Hao et al., 2017). Despite the success in common landscapes, most existing studies of this family cannot work well for POI oriented QA, since it is ineffective for them to handle the unique properties of POI elements such as tags and locations. As a result, a significant gap remains between academic proposals and the industry standard of implementing location based services.

It is nontrivial to extend existing QA models to handle the challenges of POI oriented QA. In general, the unique challenges for this problem mainly

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 542–550
December 4 - 7, 2020. ©2020 Association for Computational Linguistics

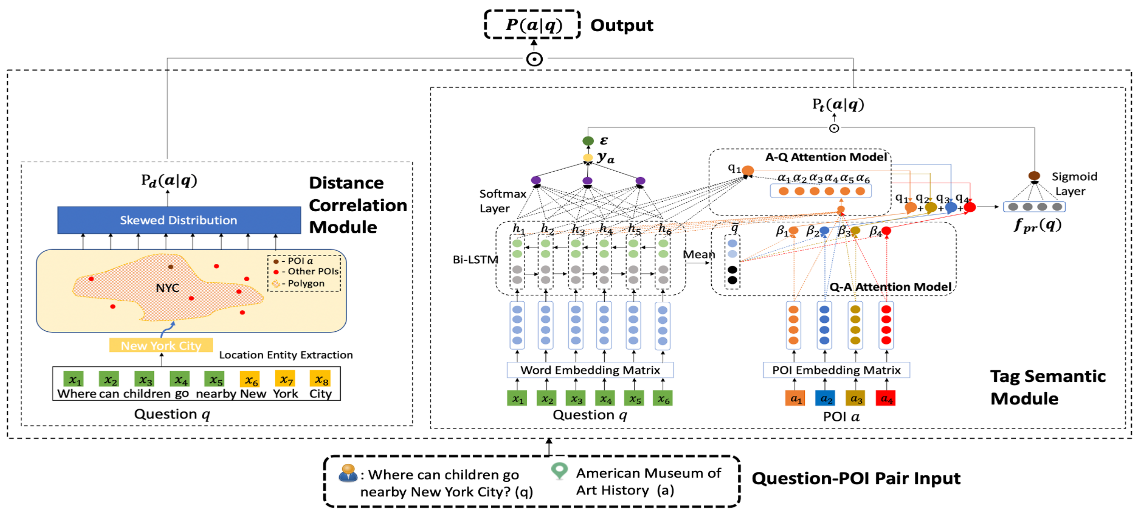

$P(a|q)$
Output
$P_d(a|q)$
Distance Correlation Module
Skewed Distribution
NYC
- POI a
- Other POIs
- Polygon
New York City
Location Entity Extraction
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8
Where can children go nearby New York City
Question q
$P_t(a|q)$
ε
y_a
Softmax Layer
A-Q Attention Model
q_1
$\alpha_1 \alpha_2 \alpha_3 \alpha_4 \alpha_5 \alpha_6$
q_1 q_2 q_3 q_4
Sigmoid Layer
$f_{pr}(q)$
h_1 h_2 h_3 h_4 h_5 h_6
Bi-LSTM
Mean
$\bar{q}$
β_1 β_2 β_3 β_4
Q-A Attention Model
Word Embedding Matrix
POI Embedding Matrix
x_1 x_2 x_3 x_4 x_5 x_6
Question q
a_1 a_2 a_3 a_4
POI a
Tag Semantic Module
: Where can children go nearby New York City? (q)
American Museum of Art History (a)
Question-POI Pair Input

Figure 1: The overall architecture of our model. Generally, the model is made up of two parts, namely tag semantic module p_t and distance correlation module p_d. The model takes the Question-POI pair as input, and the probability of choosing a POI given the question as output.

come from two aspects. First of all, when asking POI oriented questions, people tend to emphasize certain needs, which correspond to some POI properties, such as the popular users of the POI, the service provided by the POI, and the types of the POI, etc. Hereafter we name all such POI properties as tags. Identifying such information in the question that is related to the tags of POI is crucial in this task, thus creating a bottleneck. Take the question "Where can children go nearby New York City" as an example, the word "children", being regarded as both a question term and a POI tag, plays an important role in identifying the corresponding POIs. Second, proximity-related terms such as "nearby" and "close" deserve special treatment. Considering the same example, if there is "nearby" in the question, the candidate POIs should be mainly located outside New York City; whereas if without, the candidate POIs should be within New York City. Furthermore, for different location entities such as "nearby New York City" v.s. "nearby Manhattan", the distance scopes of "nearby" are also different. In contrast, traditional QA methods are not able to treat these terms in their models properly and thus leading to a poor performance on POI oriented QA.

In this paper, we propose a POI oriented QA model with Joint Inference (named as PJI for short) to tackle the challenges mentioned before. PJI mainly has two modules which are named as tag semantic module and distance correlation module. The tag semantic module is used to automatically search for relevant POIs based on semantic tag information. Besides, in order to capture specific patterns buried in questions and POIs, we develop a novel cross attention based question embedding structure. Therefore the mutual influence between questions and POIs is taken into account. In the distance correlation module, we adopt a skewed distribution on three-level locations including city, district, Area of Interest (AOI) to fit the distance distribution between candidate POIs and mentioned location terms in the question. Both modules are fused together and optimized in an end-to-end manner for retrieving the final POI list. Our major contributions can be summarized as follows:

- We tackle the POI oriented QA problem by proposing a new deep learning model with joint inference.
- We leverage two neural network modules to build a bridge between questions and POIs on both POI tags and question location terms. We also adopt a skewed distribution method to deal with proximity-related terms.
- We design a special embedding structure using cross attention mechanism to obtain a more precise and flexible representation of questions.
- We conduct comprehensive experiments on two real-world datasets enabling the evaluation of the results from different perspectives. Experimental results demonstrate significant improvements of PJI over all the state-of-the-art baselines.

2 Related Work

QA with Semantic Parser Semantic parsing shines at handling complex linguistic constructions and obtains reasonable performance on question answering problems. Traditionally, semantic parsers like AMR (Banarescu et al., 2012) and SQL (Androutsopoulos et al., 1995) map sentences to formal representations of their underlying meaning (Shen and Lapata, 2007; Yao et al., 2014; Hill et al., 2015; Talmor et al., 2017). By leveraging a knowledge base, semantic parsing is reduced to query graph generation and stage searching.

Neural Approaches for QA With the recent development in deep learning, neural networks have achieved great success in question answer problems (Salakhutdinov and Hinton, 2009; Collobert et al., 2011; Socher et al., 2012; Hu et al., 2014; Tan et al., 2015, 2016). Most of these models use a deep neural network like GRU (Chung et al., 2014) and LSTM (Hochreiter and Schmidhuber, 1997) to handle the long texts required for QA. Further improvements like attention mechanism are applied to focus on the most relevant facts (Hao et al., 2017; Zhao et al., 2019). The relevance score of each QA pair is the cosine similarity of the semantic vectors. The final answers to each question are then sorted by the similarity score.

Probabilistic Deep Learning Models The base of probabilistic deep learning models is to use the neural network as a conditional model parameterised by the weights in the network when some inputs are given. The output is obtained by optimizing the parameters in the model with the estimates provided by Bayesian framework. Several probabilistic models have been used in tasks like question answering with knowledge graph and link prediction (Wang et al., 2007, 2014; Zhang et al., 2018). The main advantage of this complete separation of the neural network from Bayesian model is that the good features generated by the network are well used to make predictions, which gives the model high flexibility and accuracy.

3 Our Model

3.1 Preliminaries

Point of interest (POI) is a dedicated geographic entity on an online map where someone may find useful information, like a restaurant, a hotel, or a travel spot. Compared with the common entities in knowledge graph, POI has two important properties which are tags and location. Tags refer to a short text (one or several words) in a POI describing its service (e.g. fast food or entertainment), its major users (e.g. kids or lovers), its types (e.g. restaurants or shopping mal), etc. Users are greatly facilitated by informative tags when searching for the POIs. In addition, each POI has three location properties named as location entities recording the POI located city, district, and area of interest (AOI). Here AOI refers to a polygonal area in a 2D map which usually contains several POIs, New York Central Park for example. For each location entity, it is possible to find a set of POIs within the entity.

Given a question q, the POI oriented question answering seeks to parse the question, then return a set of POIs which can be seen as the answer result according to the question. For example when q is the question “Where can children go on weekend in New York City?”, the answer is a set of POIs which are places for kids to play in New York City satisfying the information request conveyed by the user. It is possible to further rank the POIs according to some POI recommendation algorithms but it is beyond the scope of this paper.

3.2 Model Overview

In our framework, the dataset is a set of question-POI pairs which can be represented as D $=\{q_i, a_i\}_{i=1}^{N}$, where q_i refers to a question, a_i refers to a POI answering the question. Our model PJI aims to retrieve the correct POIs with respect to each question which corresponds to the function $P(a_i|q_j)$ returning the probability that POI a_i satisfies the question q_j. The overall structure of our model is illustrated in Fig 1. Our model consists of two neural network modules, as described below:

Tag Semantic Module In the POI oriented QA, some question terms correspond to POI tags and thus serving as a bridge between questions and POIs. In Fig 1, “children” is both a term in the question and a tag of POI. However, it is not easy to match the query terms to POI tags directly. For example, terms such as “kids”, “baby” can also correspond to the tag “children”. In our model, for each question q_j, we learn the probability that a tag y_a is included in the question q_j, which can be represented as $P(y_a|q_j)$. Given the question embedding and tags, POIs with the corresponding tags are chosen as the answers at the tag level. We then develop a neural network module specialized for calculating $P(a_i|y_a, q_j)$, which is the likelihood of POI a_i being selected given the tag y_a and the question q_j. Above all, the likelihood of choosing POI a_i as the answer to the question q_j is the marginal probability mass function over all tags:

$$p_t(a_i|q_j, \theta_t) = \sum_{y_a \in V_t} P(a_i|y_a, q_j) * P(y_a|q_j) \quad (1)$$

which sums out all possibilities of tag variables, where V_t refers to the tag set.

Distance Correlation Module Apart from tags, there also exists a distance correlation between questions and POIs. In this module, we first extract the location entity using NER tools, since the location entity vocabulary is very large and has fixed names. The questions usually contain some proximity-related terms, such as "nearby" and "close to". It is hard to confidently determine whether a POI is in or out an extracted location entity polygon considering such proximity-related terms. Thus, instead of directly identifying the candidate POIs by the location entity appeared in the question, we also calculate the probability of candidate POIs considering both the distance to the extracted entity polygon and the question context. With this motivation, we introduce another probability function $P(a_i|y_l, q_j)$, which captures the probability of POI a_i being the answer of the question q_j if the location entity y_l appears in q_j. We denote the likelihood of choosing POI a_i given the question q_j based on distance correlation as:

$$p_d(a_i|q_j, \theta_d) = \sum_{y_l \in V_d} P(a_i|y_l, q_j) * P(y_l|q_j) \quad (2)$$

where $P(y_l|q_j) = 1$ if y_l appears in q_j, otherwise $P(y_l|q_j) = 0$, $y_l \in V_d$ and V_d refers to the location entity set.

Overall Formulation With the two modules above, the parameters of the function $p(a_i|q_j)$ can be estimated by maximizing the log-likelihood as follows:

$$\max_{\theta_t, \theta_d}(\frac{1}{N}\sum_{i=1}^{N} logp_t + \frac{1}{N}\sum_{i=1}^{N} logp_d) \quad (3)$$

3.3 Neural Network Module for Tag Semantic Matching

Due to the linguistic diversity of describing a certain tag, it is almost impossible to recognize the tag with exact matching. Therefore, we build a tag recognizer which can be jointly trained with the model. After that, we can get the POIs given the tag and question representations with a cross attention architecture.

QA Embedding We use two dense d dimensional vector representations of questions in the module. The first one is represented as $f_{ent}(\cdot) : q \rightarrow R^d$, which takes the Word2Vec vectors as input, then feeds them into a Bi_LSTM neural network with a pooling layer. It helps to capture the sequence information in the question and is used in POI tag recognition. The other one is denoted as $f_{pr}(\cdot) : q \rightarrow R^d$, which leverages attention mechanism to distinguish and catch the most important information in questions. Rather than apply a simple attention layer, we introduce a special cross attention mechanism tailored to this task originally first brought by Hao et al. (2017). The answer POI is embedded with function $g(\cdot) : a \rightarrow R^d$, which calculates the average value of POI tag vectors obtained from Word2Vec.

Cross Attention Mechanism Similar to f_{ent}, the structure f_{pr} consists of a Bi_LSTM network with a pooling layer, whereas the output of it interacts with the POI representation and takes the attention weights into account. The final attentive embedding consists of POI-towards-question embedding and question-towards-POI embedding. In POI-towards-question step, we train weights between every state in the Bi_LSTM hidden layer and POI tag, then get a set of weighted question vectors regarding each POI tag. The following formulas are proposed to calculate the vectors:

$$\alpha_{mn} = softmax(h(W^T[h_n; e_m] + b)) \quad (4)$$

$$f_{pr}(q)_m = \sum_n \alpha_{mn} h_n \quad (5)$$

where h_n denotes the question hidden layer vector. e_m denotes the POI tag embedding vector. α_{mn} is the weight of attention from the tag e_m to the nth word in the question. $h(\cdot)$ is an activation function.

In question-towards-POI step, we learn a set of weights between the question pooling layer vector and POI tag. Using the weighted question vectors in the first step and the weights in the second step, we can then get the final weighted double-sided attentive question vector by multiplying and adding them up.

$$f_{pr}(q) = \sum_m \beta_m f_{pr}(q)_m \quad (6)$$

$$\beta_m = softmax(h(W^T[f_{ent}(q); e_m] + b)) \quad (7)$$

where β_m denotes the attention of question towards answer aspects.

POI Tag Recognition We exploit the question context to build the tag recognizer. For instance, if the question contains the word "dating", it means that the target audience is lovers and the POI type should be like parks and restaurants. Specifically, we embed the question to a d dimensional vector using embedding function $f_{ent}(\cdot) : q \rightarrow R^d$ as described above. Then given the embedding vector of the question q, we set the likelihood of choosing tag y_a by adding a softmax layer as follows:

$$P(y_a|q) = softmax(W_y^T f_{ent}(q)) \quad (8)$$

$$= \frac{exp(W_y^T f_{ent}(q))}{\sum_{y' \in V_t} exp(W_{y'}^T f_{ent}(q))} \quad (9)$$

where V_t refers to the tag set in the POI dataset.

Tag Based POI Retrieval Since the number of POIs in the dataset is often very large, it is necessary to obtain some candidate POIs based on tag information and discard the irrelevant ones. Having $P(y_a|q)$, we can get POI tag y_a with the highest score. We then filter out POIs with the tag y_a from the dataset and form the candidate set. Precisely, we introduce a Dirac delta function ϵ to accomplish this process. For POIs with y_a, the function $\epsilon_{y_a}(a)$ is set to 1, while for POIs without, $\epsilon_{y_a}(a)$ is set to 0. The filtering of POI greatly reduces the workload of subsequent process, and has a significant effect for large-scale data.

After obtaining the tag y_a in the question q_j and the function ϵ, the next step is to retrieve the corresponding POIs, which is represented as $P(a_i|y_a, q_j)$ in Section 3.2. Suppose questions are embedded using the embedding function $f_{pr}(\cdot) : q \rightarrow R^d$. In this function, the final output question embedding is the weighted cross-attentive vectors where informative patterns in questions are strongly focused. The likelihood of choosing a_i given question answer embedding and POI tag can be represented as follows:

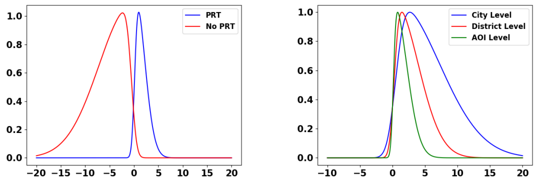
$$P(a_i|y_a, q) = sigmoid(f_{pr}(q)^T g(a_i)) \cdot \epsilon_{y_a}(a_i) \quad (10)$$

3.4 Neural Network Module for Distance Correlation

This section mainly discusses the approach for matching the POIs to the question based on the aspect of distance correlation. As discussed in Section 3.2, the first step of distance correlation module is to find location entities in the question. In our model, we assume that there are three types of location entities: city, district, AOI (Area of Interest). AOI is a location entity on the map with boundaries (e.g. Central Park) which usually belongs to a district (e.g. Manhattan) of a city (e.g. New York). We first build a dictionary storing all of the location entities and their corresponding scopes as well as types. For every question, we extract the location terms in the question with an NER (named entity recognition) tool before mapping them onto the dictionary. Note that the location term extracted directly from questions can be hierarchical. For example, AOIs may appear in the form of District+AOI (e.g. Manhattan Central Park) or City+AOI (New York Central Park) or City+District+AOI (New York Manhattan Central Park) or just itself (Central Park). Thus, we set the priority order to AOI > district > city when conducting entity mapping.

Proximity-related Terms While retrieving the POIs according the location entity, another factor we should consider is whether the question contains some proximity-related terms such as "nearby", "close to" , or "neighboring". When these terms appear in the question, people are actually expecting POIs which are close to, or outside the location boarder. It implies that the model should avoid simply returning POIs within the location polygon. Fig 2(a) shows the real-data distribution of POI with respect to questions with and without proximity-related terms according to real-world data used in our experiments.

In addition, concerning questions with proximity-related terms, the area of the location entity also has an important impact on the probability distribution of the distance between the selected POI coordinate and location entity polygon. As shown in Fig 2(b), when asking city-level questions with proximity-related terms (e.g. "Where can children go nearby New York?"), the result may contain POIs located in city suburban district or outside the city; while as for AOI-level questions (e.g. "Where can children go nearby Central Park?"), the result may only contain POIs outside



(a) (b)

Figure 2: The POI probability distribution concerning distance. X axis is the log distance between the POI and the location entity polygon. If POI is outside the polygon, the distance is positive, otherwise is negative. Y axis is the probability the POI is recommended. (PRT denotes proximity-related terms.

but close to the border of the AOI. This is because the area of a city is much bigger than that of an AOI. With different area sizes of the location entity, the probability distribution functions are quite different.

Distance Correlation Calculation The probability of choosing POI a_i given the location entity in question q_j has a proportional relationship with the distance between the POI and the location entity polygon. That is, if a POI is very far away from the expected location, the probability we recommend it is close to zero. Apart from the distance, as discussed above, proximity-related terms and location entity areas should also be taken into account when calculating the likelihood.

Given the location entity y_l extracted from the question q_j, all factors, including the distance between the polygon of y_l and POI a_i, the area size of y_l and proximity-related terms, have an impact on the likelihood distribution. Specifically, we propose a *skewed distribution* based model, which takes the distance from POI to the location entity $d(a_i, y_l)$, indicator function $\tau(q_j)$, as well as the area of location entity $s(y_l)$ as inputs. $\tau(q_j)$ is the indicator function that $\tau(q_j) = -1$ if the question contains proximity-related terms, otherwise $\tau(q_j) = 1$. The probability of choosing POI a_i having the location entity y_l and the question q_j is:

$$P(a_i|y_l, q_j) = sigmoid(W_d f(\frac{\tau(q_j)d(a_i, y_l)}{s_y})) \tag{11}$$

$$f(x) = \frac{2}{\omega}\phi(\frac{x-\xi}{\omega})\Phi(\alpha\frac{x-\xi}{\omega}) \tag{12}$$

$$\phi(x) = \frac{1}{\sqrt{2\pi}}e^{\frac{-x^2}{2}} \tag{13}$$

$$\Phi(x) = \int_{-\infty}^{x}\phi(t)dt = \frac{1}{2}(1 + erf(\frac{x}{\sqrt{2}})) \tag{14}$$

Where $f(x)$ is the skewed normal distribution of x, W_d is what we want to optimize. Note that α, ξ, ω are hyper parameters. Given the formulation above, we can see if the questions do not contain proximity-related terms, $\tau(q_j)$ value is equal to 1, POIs inside the polygon scope are what we need. As for questions containing proximity-related terms, the smaller polygon area s_y is, the steeper the distribution curve will be, as a result, POIs closer to the polygon boundary will be more likely to be selected.

3.5 Inference

During inference, ideally we want to find the candidate POIs given the question q_j. In the aspect of POI tags, we select the tag y_a receiving the maximum score from $P(y_a|q_j)$. Then we reduce the candidate POI number by filtering out the POIs whose corresponding tag is equal to y_a. After that, we calculate the semantic probability of choosing the POI as the answer. In distance correlation stage, the computation is quadratic in the number of location entities and thus is too expensive. We first extract the location entity y_l by NER and calculate the distance from POI coordinates to the polygon of y_l afterwards. Take the question in Fig 1 as an example, after the extraction step, we obtain the location entity "New York City". Finally, we select the top 5 candidate POIs with top scores as the result.

4 Experiments

4.1 Experiment Setup

Datasets We construct two large-scale datasets, both of which are based on queries extracted from query logs of a widely used mobile search engine App and POIs obtained from an online map service provider. In order to filter the POI related questions from the search engine App, we design a set of templates such as "where can [*] go in [*]" and keep all the queries that match with the templates. To construct the ground truth of question-POI pairs used in the training period, given questions satisfying the templates, we crawl the related website clicked by the user inquiring the question, then calculate the similarity between the website text and POIs. Finally we choose POIs that are most similar to the website as the answer POI. For determining the answer POIs, we sort the POIs according to their probability and choose the top-K result as the final output. Moreover, all the datasets are anonymized due to privacy concerns.

- Dataset A. This dataset mainly contains POI related questions whose geographic entities are located in Beijing. We sample questions out of one month records satisfying the template, and construct 11,000 question-POI pairs. The question data is divided into two parts randomly. The training set contains 10,900 question-POI pairs. The testing set is made up of 100 questions which do not appear in the training set.
- Dataset B. In this dataset, the location of the questions is not restricted in Beijing. We randomly sample questions to construct the question-POI pairs which covers most of the cities and many popular visited districts and AOIs in China. Similarly, there are 350,900 question-POI pairs in the training set, and 100 questions in the testing set.

On average, the length of questions in 2 datasets is 37.8 Chinese characters. The average length of POIs including its tag information (name, tags, city, district and AOI) is 30.3 characters. We later evaluate our model on these two datasets by the percent of hits at K (%hits@K) which is the percent of question-POI pairs whose POI appears in top-K retrieved POI.

Baselines We compare our model with several state-of-the-art baselines to show the effectiveness of our model. The first two are semantic parser based methods using tag information and the left ones are deep learning methods based on semantic matching.

- Template Matching Method (TMM) This method first converts the questions into SQL queries according to the templates, then retrieves POIs from database.
- StanfordCoreNLP Stanford CoreNLP is an integrated NLP toolkit providing a wide range of linguistic analysis tools. We use it as a Chinese semantic parser to recognize the tags. Based on the tool, we can turn the question into SQL queries according to the semantic characteristics of the tags.
- Bi-LSTM It is a basic deep neural network model which takes the Word2Vec vectors of query and answer as input and their cosine similarity as output (Tan et al., 2015). It utilizes a Bi-LSTM layer to capture question semantic features and then feed them into a pooling layer. This model takes the max margin hinge loss as the loss function.
- Bi-LSTM+ATT (AQA) Compared with Bi-LSTM, in this model, each Bi-LSTM output vector will be multiplied by a softmax weight, which is determined by the answer embedding.
- Bi-LSTM+C-ATT (CAQA) This is a state-of-the-art end-to-end neural question answering model introduced by (Hao et al., 2017). It considers the double-sided attention containing question-to-answer attention and answer-to-question attention.

4.2 Experiment Results

Overall Performance We compare our model with all the baselines whose results are shown in Table 1. Conclusions observed are listed as follows. (1) Compared with typical neural network based models, semantic parsing based methods have a higher %hits@K rate on the whole. However, with the lack of flexibility, their %hits@K rate is worse than our PJI model. (2) In general, models with attention mechanism reach better performance than models without. Bidirectional attention models achieve higher %hits@K rate than unidirectional one, which indicates there exists several parts in the questions as well as POI attributes that should be put emphasis on. (3) Our model achieves the best overall performance among all the models. In terms of %hits@K rate, no matter what K is, the rate of our model is beyond 95%. Our model utilizes several neural network modules instead of calculating the semantic similarity directly. Moreover, thanks to the cross-attention question embedding structure, our model puts strong emphasis on the distance and tag related patterns of both questions and POIs. In addition, we use a special probability distribution to handle questions with proximity-related geographic terms which are treated the same as normal questions in the baselines.

	Dataset A			Dataset B		
	hits@1	hits@3	hits@5	hits@1	hits@3	hits@5
TMM	84.9%	84.9%	84.9%	82.8%	82.8%	82.8%
CoreNLP	88.9%	88.9%	88.9%	86.5%	86.5%	86.5%
Bi-LSTM	55.6%	56.0%	61.7%	29.1%	29.4%	31.1%
AQA	65.2%	67.3%	68.8%	37.5%	38.9%	35.6%
CAQA	69.2%	56.1%	68.1%	42.1%	42.8%	49.0%
PJI	98.6%	98.9%	99.0%	97.2%	97.9%	99.1%

Table 1: The overall performance over two datasets.

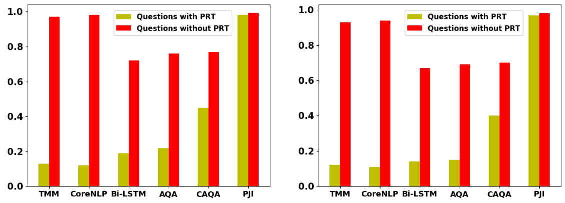
Three-level Location Performance. Table 2 shows the %hits@5 of two datasets where questions contain city, district and AOI location entities, respectively. As shown in the table, no matter which model we use, city level questions obtain the best result compared to other two types. The reason is that the number of cities in the whole nation is rather small and there is almost no duplicate city names among them. However, both AOI and district names can have a lot of duplications thus causing ambiguity and noise. Moreover, due to the hierarchical nature of the location entity, AOI names appear in different formats, which increases the difficulty of POI retrieval. Therefore, the template-based method and the end-to-end similarity matching method may be far from meeting the real-world demands of POI oriented QA. Despite the challenges we mentioned above, our model still outperforms all the baselines on city, district and AOI questions.

	Dataset A			Dataset B		
	Cit.	Dis.	AOI	Cit.	Dis.	AOI
TMM	94.1%	92.0%	91.3%	93.2%	91.6%	90.2%
CoreNLP	96.0%	93.4%	58.3%	90.3%	90.1%	31.8%
Bi-LSTM	92.1%	76.2%	8.4%	90.9%	62.1%	3.5%
AQA	92.4%	78.5%	9.1%	92.2%	62.8%	9.3%
CAQA	92.9%	79.4%	9.7%	92.6%	63.1%	9.7%
PJI	100%	99.6%	97.3%	99.8%	99.2%	97.0%

Table 2: The %hits@5 rate on questions containing different location entities.

4.3 Proximity-related Term Analysis

Fig 3 shows the %hits@5 with and without proximity-related terms on Dataset A and B. From the result we can conclude that all existing baselines cannot handle questions with proximity-related geographic terms. For traditional neural network QA models, the model has no idea how important these words are and considers them just as normal words. As a result, the results returned do not make sense to the users. Nevertheless, this problem gets tackled by the distance probability module in our model. Therefore, our model outperforms the baselines when it comes to these kinds of problems to a great extent.



(a) Dataset A (b) Dataset B

Figure 3: The %hits@5 rate concerning questions with and without proximity-related terms on two datasets.

5 Conclusion

In this paper, we propose a novel deep learning framework with joint inference to solve the POI oriented question answering task. Our main contributions lie in three aspects. First, this model handles the POI oriented QA with the help of tag semantic module and distance correlation module. Second, by introducing a cross attention based question embedding structure, we achieve a precise and flexible representation of questions. Third, the proposed model can overcome several challenges of POI oriented QA including POI tag recognition, proximity-related term processing and diverse distance correlation. Extensive experiments on two real-world datasets are carried out to demonstrate the effectiveness of our model. The result shows that our approach outperforms all the baselines and state-of-the-art models.

Acknowledgments

The work described in this paper is partially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Codes: 14200719).

References

Ion Androutsopoulos, Graeme D Ritchie, and Peter Thanisch. 1995. Natural language interfaces to databases–an introduction. *Natural language engineering*, 1(1):29–81.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2012. Abstract meaning representation (amr) 1.0 specification. In *Parsing on Freebase from Question-Answer Pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle: ACL*, pages 1533–1544.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 221–231.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, pages 2042–2050.

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

Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 12–21.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 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*, pages 1201–1211. Association for Computational Linguistics.

Alon Talmor, Mor Geva, and Jonathan Berant. 2017. Evaluating semantic parsing against a simple web-based question answering model. *arXiv preprint arXiv:1707.04412*.

Ming Tan, Cicero Dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 464–473.

Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*.

Chao Wang, Venu Satuluri, and Srinivasan Parthasarathy. 2007. Local probabilistic models for link prediction. In *Seventh IEEE international conference on data mining (ICDM 2007)*, pages 322–331. IEEE.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph and text jointly embedding. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1591–1601.

Xuchen Yao, Jonathan Berant, and Benjamin Van Durme. 2014. Freebase qa: Information extraction or semantic parsing? In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pages 82–86.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Ji Zhao, Dan Peng, Chuhan Wu, Huan Chen, Meiyu Yu, Wanji Zheng, Li Ma, Hua Chai, Jieping Ye, and Xiaohu Qie. 2019. Incorporating semantic similarity with geographic correlation for query-poi relevance learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1270–1277.