IALP 2016

Improving the Effectiveness of POI Search by Associated Information Summarization

Hsiu-Min Chuang, Chia-Hui Chang*, Chung-Ting Cheng
Dept. of Computer Science and Information Engineering
National Central University

Outline

- Introduction
- System architecture
  - POI associated information extraction & summarization
  - POI searches
- Experiment
- Conclusion

Introduction

- Ubiquity of mobile devices and smartphones
  - Mobile applications and services, especially in LBS
- A market research report by *comScore* in 2014
  - 90% of users have used a local search
  - Finding an address/POI, products/services needed; querying TEL of a business
- **Point-of-interest** (**POI**) is a human construct which is associated with a location
  - A POI can be represented by a (name, address, category, associated information) tuple

Some Results are Not Complete

- The results of a query “Global Village Organization (地球村美日語)” in Taoyuan on Google Maps and Bing Maps
  - Not complete (vs. official Website)

(a) Google Maps

(b) Bing Maps

(c) Official Website

Motivation for Mining POIs from the Web

- Manual annotation for constructing a POI database is costly and insufficient
- When a POI query can't be found on maps, we can search it by search engines
- The Web has become a media for publishing information
  - Business/organization Websites contain the associated information such as service/product, and descriptions.
  - The POIs associated information are especially essential for POI search

Why POI Database Construction Is Important?

- **POI data is the primary element of all LBS**
  - 78 % of local-mobile search-result bring offline purchases
  - Local search is business
- **Data is king**
  - In the past, Google bought geo-data in many countries
  - Today, Google Maps owns the largest POI data in the world
  - In the future, we might pay Google for using the data
- **Construct our POI database**

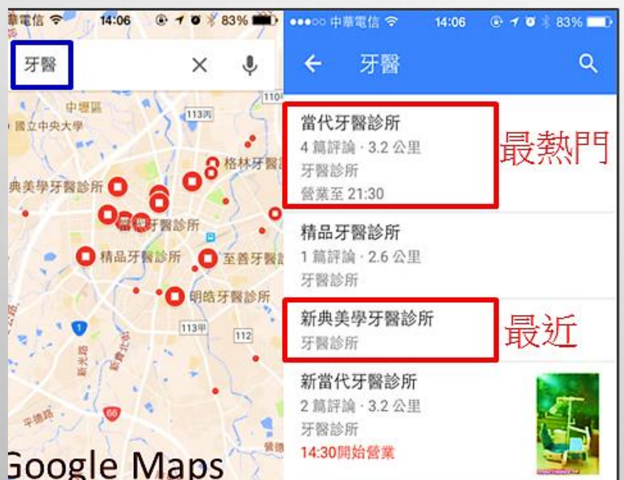
POI Database Construction and Search

Four problems

- Construct models for address and POI name recognition
- Crawl address-bearing pages scattered across the Web
- Pair a POI name with respect to the address
- **POI search on maps**
  - Multiple search-results integration
  - POI ranking by **relevance** and **distance**

Different Ranking Strategies

- Comparison of the results from Google Maps and Yahoo local search by a general query, i.e., “dentist (牙醫)”
  - Different criteria (relevance, distance, and rating)
- No match POIs for a specific POI query in user’s scope
  - Expand the scope until matching the POI



(a) Google Maps

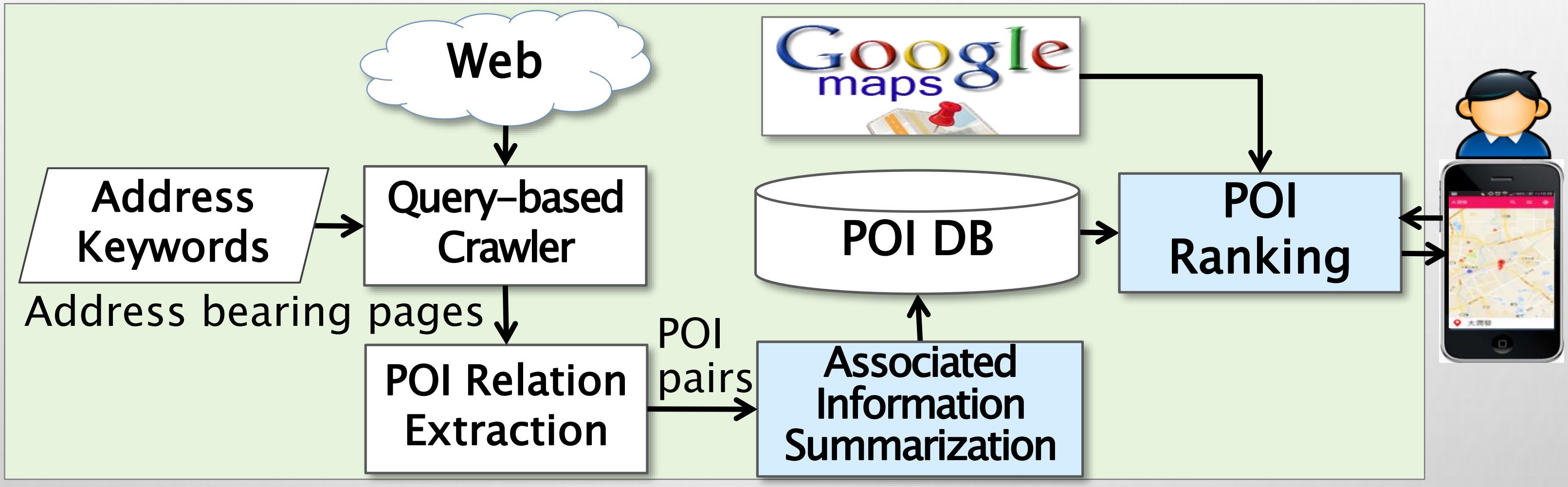
(b) Yahoo local search

Related Work

- **Information extraction**
  - **POI-name recognition:** Rae **[SIGIR'12]**
    - Collect POIs by Wikipedia and social media as seeds
    - Train a POI-name model by CRF
    - Predict the locations of POIs from Flickr images
- **Information retrieval**
  - **POI ranking:** Bauer et al. **[WWW'16]**
    - Offline-search retail-locations from Web
    - Produce an ontology of purchase needs
    - Rank by the relevance and distance

System Architecture for POI Search

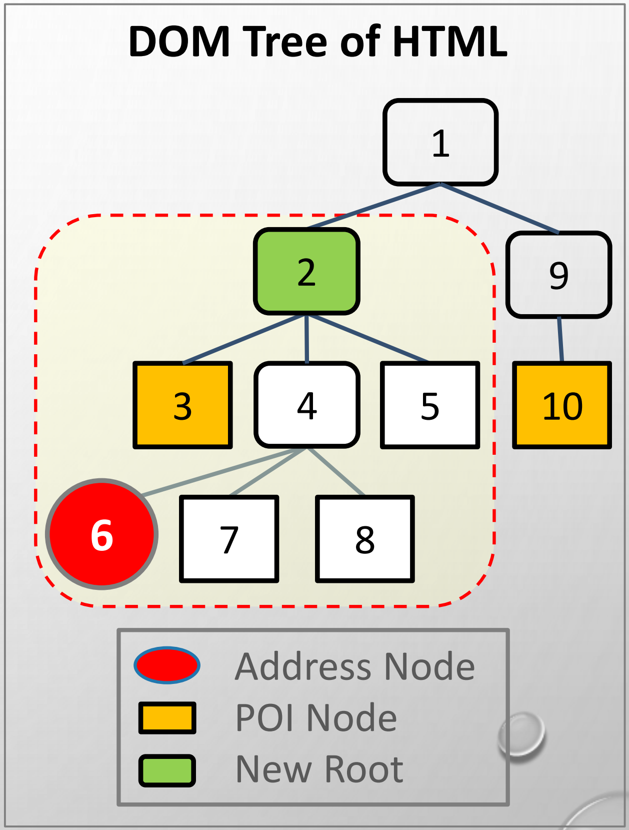
Goal: Automatically extract POIs from Web, construct a POI-DB to enable POI search on maps



1. **Associated information summarization** for POI retrieval
2. **POI ranking** for multiple search-results integration

1. POI Associated Information Extraction

- Two sources: **Webpages, Google snippets**
  - For each POI pair in Webpages
    1. Find address node in DOM tree of HTML
    2. Find the corresponding POI node
    3. Find the lowest common ancestor as root
    4. Obtain the sentences in the sub-tree as associated Information for the pair
  - Collect top 10 Google snippets by "address + POI-name" as a query
    - Complement information



1. POI Associated Information Summarization

- Select the most representative sentences for each POI pair using the query likelihood model
- Rank the sentences

$$P(s \mid p) = \frac{P(p \mid s)P(s)}{P(p)} \propto P(p \mid s)$$

- Combine TF-IDF and LDA

$$P(p \mid s) = \lambda(\text{TF-IDF}) + (1-\lambda)P_{LDA}(p \mid s)$$

$$P_{LDA}(p \mid s) = P(p \mid \Theta_s, \Psi_k) = \sum_z P(p \mid z, \Psi_k)P(z \mid \Theta_s)$$

p: POI, ***s***: sentence, $\boldsymbol{\lambda}$: parameter, ***z***: topic

tf-idf: term-frequency(*p*) × inverse-document-frequency(*p*)

$\boldsymbol{\theta_s}$: distribution of topics, $\boldsymbol{\psi_k}$: distribution of words

2. POI Search Method

- Concern about **search scope** and **ranking criteria**
  - Local search expanded its scope until results found or global search (Cheng et al. 2015)
  - Ranking by POI relevance and distance
    - If relevance of POIs are the same, they are ranked by distance

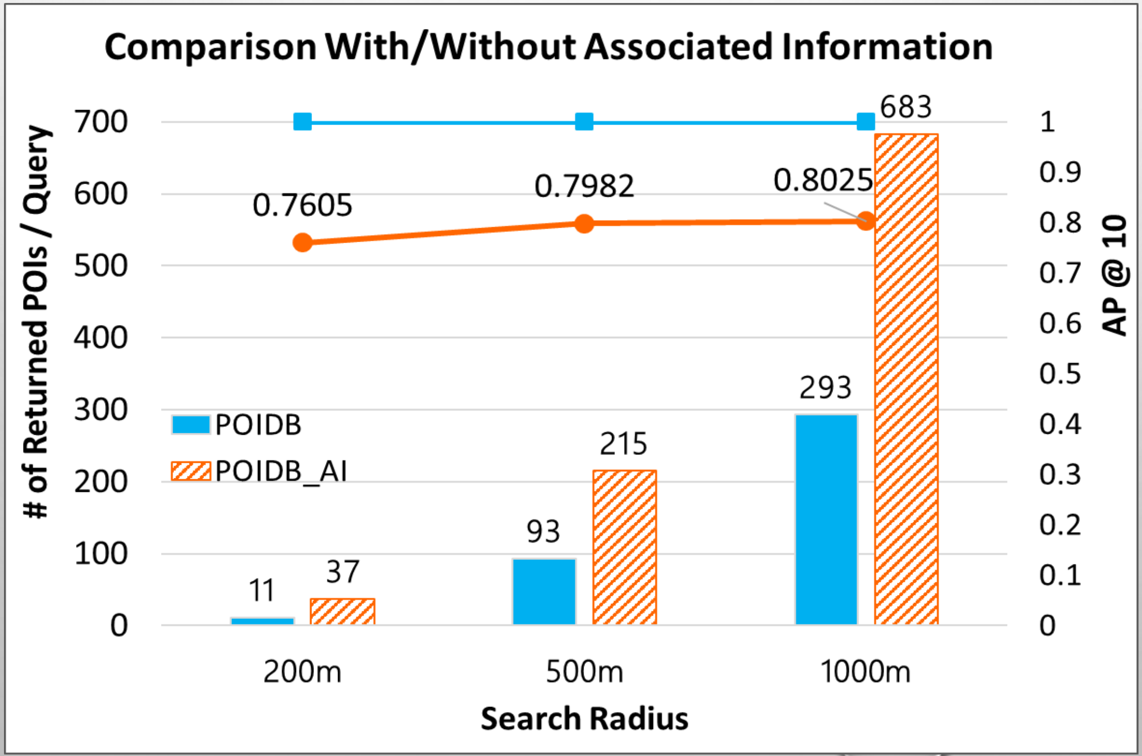
```
Algorithm Search (q, r, GPS, i)
Input: user query q, user's GPS, search scope r
Output: POI list
Iteration i is constant, i>0; confidence δ=0.5
If (i = 0) EXIT
IS = Solr(r, q, GPS)  Google Place API(q, GPS)  Online search(q)
C = Ranking(IS, δ)
If (C = null)
    Search(q, r×3, GPS, i-1)
Else
    C order by the relevance and distance
```

2. POI Ranking Model

- POI ranking model is designed to identify whether POI is relevant to user's query.
  - Be considered a classification or learning-to-rank problem
- Features extraction
  - # of match-word & match-positions for query and POI
  - Cosine similarity for query and POI name
  - Longest common sequence for query and POI name
  - # of click-through of the POI pair
- Training data: label 2,000 POIs by 200 queries
- Methods: Use libSVM and RankSVM to rank POIs

Comparison of IR with/without Associated Information

- Location: 9 centers, 3 types of radius: 200m, 500m, 1,000m
- Query: 18 (e.g., restaurant, hotel, clinic, parking lot)
- Retrieval more relevant POIs when IR enriches the associated information of POIs, but the accuracy is decreased

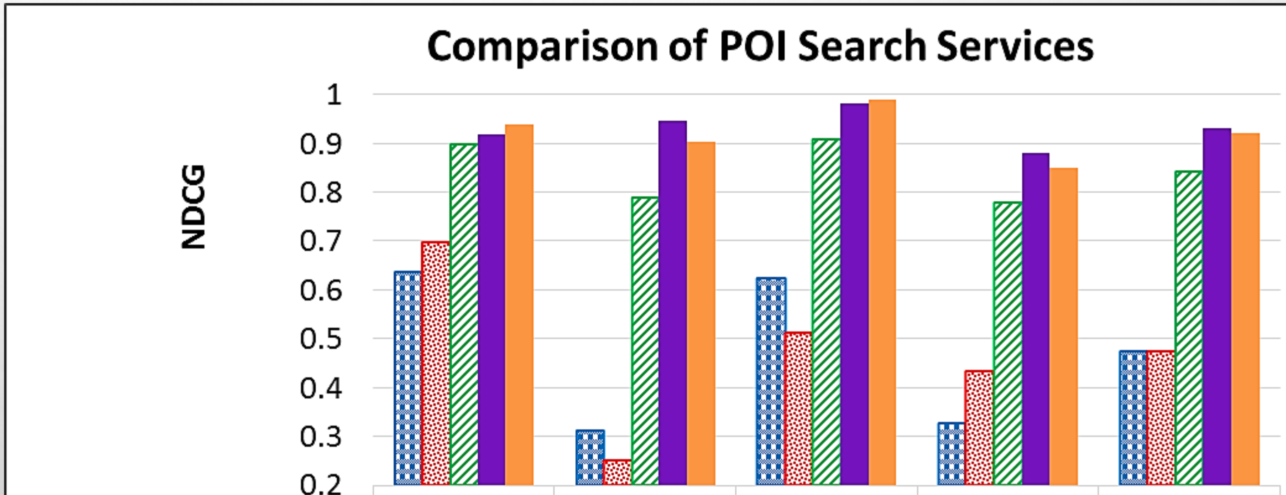


Comparison of POI Search Services

- ▶ To evaluate the constructed POI-database and the POI-search method, we compared the performance by NDCG
- ▶ Use 40 queries (general keywords and specific POIs) for urban and rural areas (8 locations) to evaluate top 10 results, respectively

$$NDCG_p = \frac{DCG_p}{IDCG_p}$$

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2(i)}$$



	urban	rural	general	specific	Avg.
wikimapia	0.638	0.313	0.624	0.327	0.476
what's the number	0.698	0.251	0.513	0.435	0.474
Google Map	0.898	0.790	0.908	0.779	0.844
IS+LibSVM	0.917	0.946	0.982	0.881	0.932
IS+RankSVM	0.937	0.904	0.990	0.851	0.920

Conclusion

- Automatically construct the POI database from the Web
- Consider different ranking strategies for POI search
  - Obtain a good performance for POI search (NDCG 0.932)
- Future work
  - POI database maintenance
    - Incremental crawling for new POIs
    - Early detection for outdated POI pairs
    - Other POI associated information extraction
  - POI recommendation

Thank You for Your Listening