LOCATION-AWARE QUERY PARSING FOR MOBILE VOICE SEARCH

Junlan Feng
AT&T Labs-Research, 180 Park Avenue, Florham Park, NJ 07932, USA

ABSTRACT

Mobile voice search provides users an easier way to search for information using voice from mobile devices. Most mobile search applications have access to the latitude/longitude coordinates of the device, which can be exploited to deliver location-specific search results. Geography and queries users submit are inextricably intertwined. In this paper we present the findings of a study on the spatial proximity of queried location and device location in a local search application. We then propose a location-aware query parsing model to parse queries into concepts that are necessary for high-precision search.

Index Terms—Voice search, mobile search, query parsing

1. INTRODUCTION

The focus of web search recently is moving into the mobile world. With the dramatic penetration of broadband mobile networks coupled with the global proliferation of smartphones, mobile devices are expected to overtake personal computers (PCs) as the most popular way to access the Web within five years. Search is one of the main drivers for mobile internet usage. However, mobile search is inherently different from its desktop counterpart.

First, the interface capabilities of mobile devices are very different from a typical PC. On one hand, the small screen and small keyboard of most mobile devices currently limit the effectiveness of search applications. At the same time, certain mobile devices are endowed with interface capabilities that are beyond those of a typical desktop computer, such as touch screens, accelerometers, cameras, and built-in microphones. Hence, as of today most popular mobile search applications have been voice-enabled. Users can choose to type or speak their queries.

Second, while a PC is stationary, a mobile device is moving with the bearer. It is increasingly common that mobile applications are allowed to access to the spatial location of the device. Mobile queries are contextually situated. For instance, queries from users in New York City are very different from those in Alaska. In order to behave robustly and cooperatively, a mobile voice search system needs to incorporate spatial location information of the device through all levels of processing from speech recognition, query understanding, search and user interface design.

In [1], the authors presented geocentric language models (LMs) for local business voice search in mobile context. The algorithm constructs one LM for each business center, which is chosen based on local business density. The obtained models achieved 16.8% absolute ASR word accuracy improvement and significant speedup in speech recognition time. A related geocentric language model was reported in [2]. It used city-specific LMs, where queries mentioning the same city are grouped together as the training data for the city-specific model. It showed modest gains when these models are combined with a nationwide LM.

For acoustic models, [3] proposed an approach to building local acoustic models for city-state for a business search application. The motivation was to capture dialectal variations of American English by adapting a general US HMM to speech from different regions. The authors partitioned the training data into six subsets according to six broad US dialect regions. The US HMM was then adapted to the regional data by sequentially applying maximum likelihood linear regression (MLLR) and Maximum A Posteriori (MAP) algorithms. This adaptation yielded absolute word accuracy gains of (0.2%-0.4%).

For mobile search, device location has been used as one of the key factors for ranking. A user submitting the query restaurants more likely prefers the restaurants nearby. The spatial location of the device also influences query understanding. For instance, the query Clay Center could mean the Clay Center city in Kansas or the name of a body shop in Cincinnati Ohio. Accurately disambiguating this query relies on the user location.

In this paper, we focus on two issues. First, we examined the relationship between mobile device location and queried location (location mentioned in the query) in a voice local search application. We report our findings. Second, we built location-aware query parsing models based on the analysis. We compared the performance of these models with a nationwide generic model.

The remainder of the paper is organized as follows. In Section 2, we describe our analysis of locations of mobile queries. In Section 3, we present our location-aware query parsing models to parse a voice query into concepts that are necessary for high-precision search. In Section 4, we present
an evaluation. Finally we summarize our findings and future work in Section 5.

2. LOCALITY OF MOBILE QUERIES

Speak4it is a voice-enabled local search system currently available for iPhone devices. It allows users to speak local search queries in a single utterance and returns information of relevant businesses. We analyzed the geography aspect of mobile queries using query logs of Speak4it. More specifically we randomly choose 20,000 queries with device latitude/longitude coordinates from the query log archive. We converted device coordinates into city and state using Twitter geocode APIs http://apiwiki.twitter.com/. We parsed each query into search terms(what) and location terms(where). The parser further breaks down the location terms into finer grained location entities such as state, city, street, zip, neighborhood, landmarks. We used a statistical parser introduced in [4], which we will overview and extend in Section 3.

2.1. Basic Analysis

Of the 20,000 random queries, only 18.1% of them have been parsed with locations. Table 1 provides further analysis. Among the queries with location information, 65.3% explicitly include State; 49.4% have both State and City; 24.3% contain City only, and the rest (10.4%) don’t include either City or State, such as landmarks only. We also examined the agreement between parsed query location and device location. The third column of Table 1 shows the agreement ratio. The majority (73.4%) of queried states are the same as the state that the device belongs to. This ratio decreases to 14.0%-19.6% for cities.

<table>
<thead>
<tr>
<th>Locations</th>
<th>Percentage</th>
<th>Agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>65.3%</td>
<td>73.4%</td>
</tr>
<tr>
<td>City &amp; State</td>
<td>49.4%</td>
<td>19.6%</td>
</tr>
<tr>
<td>City NoState</td>
<td>24.3%</td>
<td>14.0%</td>
</tr>
<tr>
<td>Others</td>
<td>10.4%</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 1. Queried Location Versus Device Location

In order to uncover motivations of the users who choose to include location in their queries, we compared queries from the 50 high-population U.S cities (according to the U.S. Census Bureau), mostly metropolitan areas (Metro) with the rest of the country (Rest). Table 2 illustrates the comparison. Users in the Metro areas include location information in 15.8% of their queries, while the rest of country has a higher rate, 18.4%. This matches our expectation that users in big cities have less need to specify a location for their intended business, since there are a lot more businesses available in the metropolitan cities. Similar to Table 1, we break down the location into four finer-grained location categories, namely, State, City, and State, City without State, and Others. Others are more specific addresses such as neighborhood, street address, landmark, zip codes. Comparing Metro with Rest, Table 2 shows that Metro queries tend to include more specific addresses, 17.6% as Others versus 9.8% as Others for Rest. 66.0% of Rest queries with location information include State, while only 56.5% of Metro queries with location contain State.

The fourth column of Table 2 shows the agreement rate between queried location and device location. We observe that Metro queries include the city of the device at much higher rates, 37.0% and 34.0% versus 19.6% and 14.0%. This implies that users in metropolitan areas tend to search business more often in the same city even if they explicitly include city information in the query.

<table>
<thead>
<tr>
<th>Location</th>
<th>Percentage</th>
<th>Agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metro (15.8%)</td>
<td>State</td>
<td>56.5%</td>
</tr>
<tr>
<td></td>
<td>City State</td>
<td>40.4%</td>
</tr>
<tr>
<td></td>
<td>City NoState</td>
<td>25.9%</td>
</tr>
<tr>
<td></td>
<td>Others</td>
<td>17.6%</td>
</tr>
<tr>
<td>Rest (18.4%)</td>
<td>State</td>
<td>66.0%</td>
</tr>
<tr>
<td></td>
<td>City State</td>
<td>50.1%</td>
</tr>
<tr>
<td></td>
<td>City NoState</td>
<td>24.2%</td>
</tr>
<tr>
<td></td>
<td>Others</td>
<td>9.8%</td>
</tr>
</tbody>
</table>

Table 2. Locations of Queries from Metropolitan Areas and the Rest of the Country

2.2. Distribution of the distance

We examined the distribution of the distance between queried location and device location. First we converted the parsed locations into coordinates. Second, we calculated the distance between two locations using the great circle distance [5]. In order to understand the dynamics between distance d and query frequency f, we bucket queries by intervals of 5 kilometers and count the number of queries in each bucket. Plotting on a log-log scale of d and f, we observe that the curve decreases roughly according to a power-law with exponent -1.3 for distances from 5 kilometers 800 kilometers. There is a long tail for the long-distance queries.

3. QUERY PARSING WITH MOBILE LOCATION

The task of query parsing is to segment ASR output (1-best and word lattices) into meaningful segments that contribute to high-precision search. In [4], we described a generic probabilistic query parsing approach using text index and search (PARIS), which takes only query content as input without considering the context in which the query is situated. In the following we first summarize the framework in [4] and then extend it to incorporate device location information into parsing.
We approximate the prior probability $P(C)$ using an n-gram model on the concept sequence. Training examples of concept sequences can be created from annotated queries.

We model the segment sequence generation probability $P(S|C)$ as shown in Equation 4 using independence assumptions. A corpus of instantiations of the concept $c_k$ are needed to infer conditional probabilities $P(s_k|c_k)$. As proposed in [4], we estimate the query subject probability $P_{sb}(S)$ as the likelihood of $s$ to be a complete query or an independent concept in the given application. A full description of how $P_{sb}(S)$ is derived can be found in [4].

$$P(S|C) = \prod_{k=1}^{m} P(s_k | c_k)$$ (4)

We exploited two well-established toolkits including standard text indexing and search software and general finite state machine (FSM) tools for higher efficiency of training and parsing. More specifically, we used Apache-Lucene as the indexing and search tools [6]. We modified Lucene to index the segment generation probability $P(s|c_k)$ and query subject probability $P_{sb}(s)$, and return these probabilities during parsing. For FSM, we use the AT&T FSM library [7]. We represent the four probabilities in Equation 3 as weighted finite-state acceptors (FSA) or finite state transducers (FST). The parsing task of finding $(S^*, C^*)$ is then a search for the lowest weight path of a composed FST from the four components.

### 3.2. PARIS with Device Location

We propose to extend PARIS by considering device location $l_d$. The target function defined in Equation 3 becomes the following Equation 5.

$$(S^*, C^*)|Q_{wcn}, l_d = \arg\max_{\{S,C\}Q_{wcn}} P(S|C, l_d) *$$

$$P(C|l_d)^{\lambda_c} * P_{ef}(S)^{\lambda_ef} * P_{sb}(S|l_d)^{\lambda_{sb}}$$ (5)

The challenge is that device location $l_d$ is represented as latitude and longitude coordinates varying in a huge range. It is almost impossible to build models for each $l_d$. Then the question becomes how we should factor in $l_d$ into query parsing. There are a few ideas we can borrow from previous work such as those in [1] [2] for location specific language modeling. [1] built geocentric language models which adapt to the local business density and achieved higher ASR performance. [2] proposed city-state specific language models. According to our analysis in Section 2, most mobile queries are locally situated, where over 80% of queries, which do not include location, presumably search for businesses close by. For queries including location, the likelihood of the queried location drops monotonically as a function of distance to the user according to a power-law distribution with exponent...
−1.3. A higher percentage of queries from the metropolitan areas constrain their searches using finer grained addresses.

As a first step towards location-aware parsing, we propose to build specific parsing models for each state and top high-population city, where queries from the same state or the same metropolitan city are grouped together as the training data for the corresponding model. In the next Section, we report our preliminary experimental results with this approach.

4. EXPERIMENTS

Our training data consists of 18 million web queries submitted to http://www.yellowpages.com/, where a query comprises two fields, SearchTerm and LocationTerm, 11 million unique business entries, and 15 thousand annotated voice queries. The annotated queries are used to train the n-gram model for $P(C)$ in Equation 3. Since we only have 15 thousand annotated queries, which is relatively small, we equalize $P(C|l_d)$ to $P(C)$.

In order to train the location-aware models, namely $P(S|C, l_d)$ and $P_{sb}(S|l_d)$ in Equation 5, for states and metropolitan cities, we group together queries and listing entries mentioning the same state, or the same metropolitan city. We use $D_d$ to denote the subset for location $l_d$, and $D$ for the entire dataset with a relationship $D = D_s \cup D_l$. As shown in Section 2, a small fraction of queries (less than 10%) are for out of state. To provide the parser an ability to parse non-local queries, we supplement $D_s$ with a random subset of $D_l$. The size of this random subset is one ninth the size of $D_s$.

The parsing task is to parse a voice query into three fields, namely SearchTerm, LocationTerm and Filler. Then, it further breaks down the LocationTerm into City, State, Zip, Neighborhood, Landmark, and Street address. We tested our approaches on 1000 randomly selected voice queries from a newer time period than the training data on the first level parsing: SearchTerm and LocationTerm. Table 3 reports the performance of these local models and compares with the national model trained on the entire dataset $D$. The parsing performance with the location-specific models is slightly better on parsing transcribed voice queries. It is around 1.2% improvement on extracting the exact location term. It shows less improvement on SearchTerm, 0.7%. Errors made from these two sets of models overlap at around 50% of the time. For the sake of space limit, we do not describe our ASR system, ASR performance and parsing performance on ASR output. We observed similar gains with ASR 1-best as shown in Table 3.

<table>
<thead>
<tr>
<th>Slots</th>
<th>Nation-Wide Models</th>
<th>Location-specific Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>SearchTerm</td>
<td>94.1%</td>
<td>94.8%</td>
</tr>
<tr>
<td>LocationTerm</td>
<td>96.0%</td>
<td>97.2%</td>
</tr>
</tbody>
</table>

Table 3. Parsing Performance on Transcribed Voice Queries

5. SUMMARY

This paper studies the relationship between mobile device locations and queried locations in a local voice search application. We observed that, as expected, most queries in mobile medium do not explicitly include location. For those including location, the likelihood of the queried location drops monotonically as a function of distance to the user. The distance follows a power-law distribution with exponent −1.3. Over 70% of the queries mentioning states agree with the state of the device location. Another finding is that a higher percentage of queries from metropolitan areas constrain their searches using finer grained addresses such as street names, landmark, neighborhood versus city and state. Based on this analysis, we extend a query parsing algorithm PARIS to consider device location as a contextual input along with the query text. A preliminary experiment showed location-specific models achieved slightly higher performance. For future work, we are interested in exploring more systematic ways to factor the device location information into parsing and testing on richer ASR output.

6. ACKNOWLEDGEMENTS

We are grateful to Ritesh Agrawal, Patrick Ellen, Barbara Hollister, and James Shanahan for their help in providing data and discussing ideas in this paper.

7. REFERENCES