

Geo-Centric Language Models for Local Business Voice Search

Amanda Stent, Ilija Zeljković, Diamantino Caseiro and Jay Wilpon

AT&T Labs – Research

180 Park Avenue Bldg. 103

Florham Park, NJ 07932, USA

stent, ilija, caseiro, jgw@research.att.com

Abstract

Voice search is increasingly popular, especially for local business directory assistance. However, speech recognition accuracy on business listing names is still low, leading to user frustration. In this paper, we present a new algorithm for geo-centric language model generation for local business voice search for mobile users. Our algorithm has several advantages: it provides a language model for any user in any location; the geographic area covered by the language model is adapted to the local business density, giving high recognition accuracy; and the language models can be pre-compiled, giving fast recognition time. In an experiment using spoken business listing name queries from a business directory assistance service, we achieve a 16.8% absolute improvement in recognition accuracy and a 3-fold speedup in recognition time with geo-centric language models when compared with a nationwide language model.

1 Introduction

Voice search is an increasingly popular application of speech recognition to telephony. In particular, in the last two years several companies have come out with systems for local business voice search (LBVS). In this type of application, the user provides a desired location (city/state) and a business name, and the system returns one or more matching business listings. The most traditional LBVS applications are commercial 411 services, which are implemented as a speech-only two-exchange dialog such as the one in Figure 1. In this approach to

LBVS, the speech recognizer (ASR) uses one grammar to recognize city/state, and then uses separate grammars for recognizing listings in each local area. This gives relatively high recognition accuracy.

Advancements in ASR and search technology have made a more information retrieval-style LBVS feasible. In this approach, the ASR typically uses a large stochastic language model that permits the user to specify location and listing name or category together in a single utterance, and then submits recognition results to a search engine (Natarajan et al., 2002). This gives the user more flexibility to “say anything at any time”. However, in recent evaluations of one-exchange LBVS we have found that locations are recognized with much higher accuracy than listing names¹. This may mean that the user has to repeat both location and listing several times (while in a traditional two-exchange interaction only one piece of information would have to be repeated). In effect, system developers have traded recognition accuracy for interaction flexibility, potentially increasing user frustration.

Advances in mobile phone technology make it possible for us to combine the advantages of two-exchange and one-exchange LBVS. The newest smart phones come with global positioning system (GPS) receivers and/or with the ability to determine location through cell tower triangulation or wi-fi. If we know the location of a LBVS user, we can use a **geo-centric language model** to achieve improved speech recognition accuracy and speed. This approach unobtrusively exploits the benefits of two-

¹The vocabulary size for listing names is larger than that for cities and states in the USA.

S City and state?
U Glendale California
S What listing?
U pizza

Figure 1: Example 411-search dialog

exchange voice search applications, while maintaining the flexibility of one-exchange systems.

In this paper, we present an efficient algorithm for constructing geo-centric language models from a business listing database and local business search logs. Our algorithm has several advantages: it provides a language model for any user in any location; the geographic area covered by the language model is adapted to the local business density, giving high recognition accuracy; and the language models can be pre-compiled, giving fast recognition time. In an experiment using LBVS queries, we achieve: a 16.8% absolute improvement in recognition accuracy and a 3-fold speedup in recognition time with geo-centric language models when compared with a nationwide language model (such as those used in one-exchange LBVS); and a 4.4% absolute increase in recognition accuracy and a 16% speedup in recognition time with geo-centric language models when compared with local area language models (such as those used in two-exchange LBVS).

The rest of this paper is structured as follows: In Section 2 we discuss related work on voice-driven local search. In Section 3 we present the motivation for and architecture of a LBVS application. In Section 4 we present our algorithm for generating geo-centric language models. In Section 5 we describe an evaluation of the performance of our geo-centric language models on business listing name queries from a deployed voice-driven search application. In Section 6 we conclude and present future work.

2 Related Work

LBVS is the most recent variation on automated directory assistance (Buntschuh et al., 1998). ASR for directory assistance is difficult for several reasons: the vocabulary is large and includes foreign words; there may be multiple possible pronunciations for many words; and the frequency distribution of words in the vocabulary is unusual, with a few words occurring very often and the rest, rarely. These difficulties are compounded by directory size.

For example, Kamm *et al.* (1995), in experiments on personal name directories, showed that ASR accuracy decreases from 82% for a 200 name directory to 16.5% for a 1.5 million name directory.

One way to reduce the directory size is to cover a smaller geographic area. For example, early LBVS covered only one city (Seide and Kellner, 1997; Collingham et al., 1997). Later, two-exchange, applications required the user to specify their desired location in the first exchange. This information was then used to select a local area grammar or language model for recognition of the listing name (Acero et al., 2008; Bacchiani et al., 2008; Yu et al., 2007; Georgila et al., 2003). In our research, we have created a novel method for constructing language models that cover a very small geographic area specific to the user's geo-location.

Another way to reduce the directory size is to drop listings that are unlikely to be requested. For example, Kamm *et al.* (1995), in their analysis of 13,000 directory assistance calls, found that a mere 245 listings covered 10% of the call volume, and 870 listings covered 20%. Chang *et al.* (2008) found that in their data sets, 19-25% of the call volume was covered by the top 200 listings. We take a different approach: we add frequent nationwide listings to our geo-centric language models to increase coverage.

Other work related to ASR in automated directory assistance has looked at ways in which users refer to locations (Gupta et al., 1998) and listings (Li et al., 2008; Scharenborg et al., 2001; Yu et al., 2007), confidence scoring for directory assistance search results (Wang et al., 2007), and ways of handling recognition errors through multimodal confirmation and correction (Acero et al., 2008; Chang et al., 2008; Paek et al., 2008). We do not address these issues here.

3 Local Business Voice Search

The current generation of smart phones contains GPS and/or can run applications that can detect the user's geo-location using cell tower triangulation or wi-fi. We hypothesize that this geo-location information can be used in mobile LBVS to improve recognition accuracy without sacrificing interaction flexibility. Our analysis of a directory assistance data set shows that in the majority of cases, users

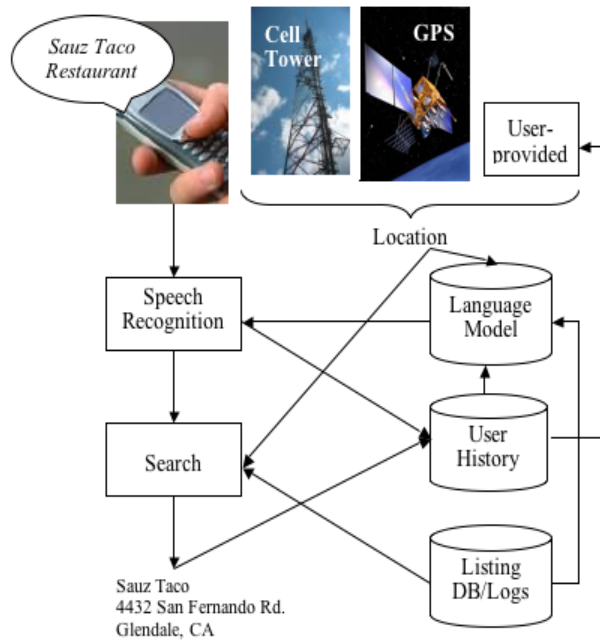


Figure 2: Architecture of a voice-driven local search application

request local listings. It is frustrating for the user of a LBVS who cannot retrieve information for a business right around the corner. So, a LBVS should maximize accuracy for local listings².

Figure 2 shows the architecture of a mobile LBVS. It includes ASR (in a speech-only or multi-modal interface), search, and presentation of results (through speech, text and/or graphics). It also includes location information from GPS, cell tower triangulation or wi-fi, or the user's query history (from previous dialogs, or previous turns in this dialog).

4 Using Location to Tailor Language Models

There are two ways to use geo-location information in ASR for LBVS. One way is to use the user's geo-location to automatically determine the nearest city. City and state can then be used to select a **local area** language model (LM) for recognizing listing names. The advantages of this approach include: human knowledge about location can be included in the design of the local areas; and local areas can be

²Of course, a LBVS should also give the user the option of specifying a different location, and/or should be able to recognize listings users are most likely to ask for that may not exist in their local area.

designed to produce a minimal number of local area LMs. However, if the user is near the edge of the pre-defined local area, the selected LM may exclude businesses close to the user and include businesses far away from the user. Also, some local area LMs contain many more directory listings than others.

Another way is to construct a **geo-centric LM** covering businesses in a given radius around the user's geo-location. This approach has the advantage that listings included in the language model will certainly be close to the user. However, on-the-fly computation of geo-centric language models for large numbers of users is too computationally demanding given current database and processing technology. It is equally impractical to pre-compile all possible geo-centric language models, since commercial GPS provides coordinates accurate to about 20 feet. Here we present an algorithm for approximating true geo-centric language modeling in a way that is computationally feasible and user relevant.

4.1 Local Area Language Models

Telecommunications companies have long understood that customers may not know the exact town in which a desired listing is, or may be interested in listings from several nearby towns. Considerable effort has been devoted to defining local service areas (LSAs) for telephone directories. In the directory service that provided the database we use, business listings are organized into about 2000 LSAs, each consisting either of several adjacent small towns or of one big city. For example, the Morristown, NJ LSA includes Morristown itself as well as 53 adjacent localities and neighborhoods spanning from Pine Brook in the north-east to Mendham in the south-west. By contrast, the New York, NY LSA contains only New York City, which includes several hundred neighborhoods. The Morristown, NJ LSA contains 50000 business listings while the New York, NY LSA contains more than 200000 listings.

We construct one LM for each LSA, giving roughly 2000 local area LMs for the whole of the USA.

4.2 Geo-Centric Language Models

To construct a a geo-centric LM for a user, we need geo-coordinates (for the center of the LM) and a search radius (to determine the extent of the LM). It



Figure 3: Geo-centric areas in New York City

is computationally infeasible to either pre-compute geo-centric LMs for each uniquely identifiable set of geo-coordinates in the USA, or to compute them on-the-fly for large numbers of users. Fortunately, the number of business geo-coordinates in the USA is much sparser than the number of possible user geo-coordinates. There are about 17 million name-address unique businesses in the USA; assuming 8-digit geo-code accuracy they are located at about 8.5 million unique geo-coordinates³. So we build LMs for business geo-coordinates rather than user geo-coordinates, and at run-time we map a user's geo-coordinates to those of their closest business.

To determine the search radius, we need a working definition of “local listing”. However, “local” varies depending on one’s location. In New York City, a local listing may be one up to ten blocks away (covering a smaller geographic area than the LSA), while in Montana a local listing may be one that one can drive to in 45 minutes (covering a larger geographic area than the LSA). Compare Figures 3 and 4. “Local” is clearly related to business density at a particular location. So we compute business density and use this to determine the radius of our geo-centric LMs.

We can do even better than this, however. Businesses are clustered geographically (in towns, shopping malls, etc.). This means that the set of listings local to one business is likely to be very similar to the set of listings local to a nearby business. So we do not need to build a separate LM for each business listing; instead, we can pre-determine the number of businesses we want to be different from one LM to another. Then we can “quantize” the business geo-

³The area of the USA with the highest business density is New York, NY, where about 270000 businesses share about 43000 geo-coordinates.



Figure 4: Geo-centric area near Vaughn, Montana

coordinates so that those that have fewer than that number of businesses different between their search radii end up sharing a single LM.

Our algorithm for constructing geo-centric LMs starts with LSAs. It proceeds in two stages: first, the business centers for the LMs are found. Second, a search radius is computed for each LM center; and third, the data for the LM is extracted.

The LM center finding algorithm uses two parameters: r_1 (radius within an LSA; should be a little smaller than average LSA radius) and N_q (number of businesses that should be different between two different geo-centric LMs). For each LSA:

1. *Find mean latitude and longitude for the LSA:* Compute mean and standard deviation for latitude (μ_{lb}, σ_{lb}) and longitude (μ_{gb}, σ_{gb}) over all businesses in the LSA.
2. *Exclude national businesses which are listed in the LSA with their out-of-LSA address and geo-coordinates:* Compute mean and standard deviation of latitude and longitude, (μ_l, σ_l) and (μ_g, σ_g) respectively, using all geo-coordinates (l, g) where: (l, g) is within a r_1 -mile radius of (μ_{lb}, μ_{gb}); l is within σ_{lb} of μ_{lb} ; and g is within σ_{gb} of μ_{gb} .
3. *Compute business density in the most business-dense region in the LSA:* find a minimum and maximum longitude (g_m, g_M) and latitude (l_m, l_M) for all businesses that are within ($\pm \frac{1}{2} \sigma_g$) and ($\pm \frac{1}{2} \sigma_l$) of μ_g and μ_l respectively. Business density per square mile (d_2) is equal to the number of businesses in the rectangle defined by the low-left (g_m, l_m) and upper-right (g_M, l_M) corner. Business density per mile is $d_1 = \sqrt{d_2}$.

4. *Compute geo-location quantization accuracy:* Choose a desired number of business listings N_q that will fall to the same geo-coordinates when the quantization is applied. This corresponds roughly to the minimum desired number of different businesses in two adjacent geo-centric LMs. Quantization accuracy, in miles, δ_{qm} , then follows from the business density d_1 : $\delta_{qm} = N_q/d_1$. Quantization accuracy for the longitude δ_g satisfies equation $distance((\mu_g, \mu_l), (\mu_g + \delta_g, \mu_l)) = \delta_{qm}$. δ_l satisfies a similar equation.
5. *Quantize geo-coordinates for each business in the LSA:* Compute quantized geo-coordinates (l_q, g_q) for each business in the LSA. $g_q = int(g/\delta_g) \times \delta_g$; $l_q = int(l/\delta_l) \times \delta_l$. Each unique (l_q, g_q) is a LM center.

The LM radius finding algorithm also uses two parameters: r_2 (maximum search radius for an LM); and N_p (minimum number of businesses within a geo-centric language model, should be smaller than average number of businesses per LSA). For each LM center:

1. Count the number of businesses at 1-mile radius increments of the LM center
2. Choose the smallest radius containing at least N_p listings (or the r_2 radius if there is no smaller radius containing at least N_p listings)
3. Extract data for all listings within the radius. Build LM from this data.

The number of geo-centric LMs can be arbitrarily small, depending on the parameter values. We believe that any number between 10K and 100K achieves good accuracy while maintaining tractability for LM building and selection. In the experiments reported here we used $r_1 = 3.5$, $N_q = 50$, $r_2 = 3$ and $N_p = 1000$, giving about 15000 LMs for the whole USA.

To summarize: we have described an algorithm for building geo-centric language models for voice-driven business search that: gives a local language model for any user anywhere in the country; uses business density determine “local” for any location in the country; can be pre-compiled; and can be

tuned (by modifying the parameters) to maximize performance for a particular application

5 Experiments

In this section we report an evaluation of geo-centric language models on spoken business listing queries from an existing directory assistance application. We compare the recognition accuracy and recognition speed for geo-centric LMs to those of local area LMs, of a national LM, and of combined LMs.

5.1 Data

Our test data comes from an existing two-exchange directory assistance application. It comprises 60,000 voice queries, each consisting of a city and state in the first exchange, followed by a business listing name in the second exchange.

We wanted to test using queries for which we know there is a matching listing in the city/state provided by the caller. So we used only the 15000 queries for which there was a match in our nationwide business listing database⁴. We categorized each query as **nationwide** or **local** by looking up the listing name in our database. We considered any listing name that occurred five or more times to be nationwide; the remaining listings were considered to be local. This method fails to distinguish between national chains and different companies that happen to have the same name. (However, from a recognition point of view any listing name that occurs in multiple locations across the country is in fact nationwide, regardless of whether the businesses to which it refers are separate businesses.) It is also quite strict because we used string equality rather than looser name matching heuristics. Example national queries include *Wal-mart* and *Domino’s Pizza*. Example local queries include *Sauz Taco* (Glendale, CA); *Dempsey’s Restaurant* (Adrian, MI); and *Concord Farmers Club* (Saint Louis, MO). Some queries contain street names, e.g. *Conoco on South Division*; *uh Wal-Mart on five thirty five*; and *Chuy’s Mesquite Broiler off of Rosedale*.

For each query in our data, we say that its local area LM is the local area LM that comes from its

⁴A query matched an entry in our database if there was a business listing in our database starting with the listing name portion of the query, in the city/state from the location portion of the query.

city and state, and that contains its listing name. Its geo-centric LM is defined similarly.

5.2 Language Model Construction

We constructed two baseline LMs. The first is a **National LM**. To take advantage of the non-uniform distribution of queries to listings (see Section 2), we also build a **Top 2000 LM** containing only information about the top 2000 most frequently requested listing names nationwide⁵. We expected this LM to perform poorly on its own but potentially quite well in combination with local LMs.

For national, top 2000, local area and geo-centric LMs, we build trigram Katz backoff language models using AT&T’s Watson language modeling toolkit (Riccardi et al., 1996). The models are built using the listing names and categories in our nationwide listing database. Listing names are converted to sentences containing the listing name, street address, neighborhood and city/state.

We predict that location-specific LMs will achieve high accuracy on local listings but will not be very robust to national listings. So we also experiment with *combination* LMs: local area combined with top 2000; geo-centric combined with top 2000; local area combined with national; and geo-centric combined with national. We use two combination strategies: *count merging* and *LM union*.

5.2.1 Count Merging

The *count merging* approach can be viewed as an instance of maximum a posteriori (MAP) adaptation. Let hw be a n -gram ending in word w and with a certain context h , and let $c_L(hw)$ and $C_T(hw)$ be its counts in the geo-centric/local area corpus L and top 2000 corpus T respectively. Then $p(w|h)$ is computed as:

$$p(w|h) = \frac{\lambda_L c_L(hw) + (1 - \lambda_L) c_T(hw)}{\lambda_L c_L(h) + (1 - \lambda_L) c_T(h)} \quad (1)$$

where λ_L is a constant that controls the contribution of each corpus to the combined model. We applied this combination strategy to local area/geo-centric and top 2000 only, not to local area/geo-centric and nationwide.

⁵We computed listing frequencies from query logs and used listings from the left-hand side of the frequency distribution curve before it flattens out; there were about 2000 of these.

5.2.2 LM Union

The *LM union* approach uses a union of language models at runtime. Let $W = w_0 w_1 \dots w_{|W|}$ be a sentence, $p_L(W)$ be the probability of W in the geo-centric/local area corpus L , and $p_T(W)$ be the probability of W in the top 2000/national corpus T . Then $p(W)$ is computed as:

$$p(W) = \max(\lambda_L p_L(W), (1 - \lambda_L) p_T(W)) \quad (2)$$

λ_L is a constant that controls the contribution of each corpus to the combined model. We applied this combination strategy to local area/geo-centric and top 2000, and to local area/geo-centric and nationwide.

Given the small size of our test set relative to the large number of local LMs it is unfeasible to train λ_L on held-out data. Instead, we selected a value for λ_L such that the adjusted frequency of the top business in the top 2000 corpus becomes similar to the frequency of the top business in the local LM. We anticipate that if we did have data for training λ_L more weight would be given to the local area/geo-centric LM.

5.3 Experimental Method

In our experiments we use AT&T’s Watson speech recognizer with a general-purpose acoustic model trained on telephone speech produced by American English speakers (Goffin et al., 2005). We ran all tests on a research server using standard settings for our speech recognizer for large vocabulary speech recognition. For each LM we report recognition accuracy (string accuracy and word accuracy) overall, on nationwide listings only, on local listings only, and on queries that contain street names only. We also report recognition time (as a fraction of real time speed).

5.4 Results

Results are given in Table 1. Comparing the baseline (National LM) to our geo-centric LMs, we see that we achieve a 16.8% absolute increase in overall sentence accuracy with a 3-fold speedup. Most of the improvement in sentence accuracy is due to better performance on local queries; however, we also achieve a 2.9% absolute increase in sentence accuracy on nationwide queries.

| LM | Recognition accuracy: String/Word [%] | | | | Real time speed |
|---|---------------------------------------|--------------------|---------------|--------------------------|-----------------|
| | Overall | Nationwide queries | Local queries | Queries with street name | |
| Nationwide language models | | | | | |
| National | 51.3/58.0 | 59.9/60.8 | 40.3/54.1 | 17.9/47.3 | 1.05 |
| Top 2000 | 23.2/31.6 | 40.6/43.3 | 9.5/25.8 | 1.3/18.3 | 0.44 |
| Local language models | | | | | |
| Local area | 63.7/69.7 | 60.8/63.2 | 69.5/77.2 | 22.4/53.4 | 0.42 |
| Geo-centric | 68.1/73.0 | 62.8/65.0 | 75.0/81.7 | 15.1/49.7 | 0.36 |
| Combined language models, LM union | | | | | |
| Local area, national | 58.9/64.5 | 61.4/62.3 | 57.9/67.1 | 21.8/50.6 | 0.84 |
| Geo-centric, national | 64.7/69.1 | 63.6/64.5 | 67.2/74.5 | 23.2/52.1 | 0.78 |
| Local area, top 2000 | 60.0/67.0 | 62.1/65.8 | 61.8/71.3 | 20.6/50.3 | 0.45 |
| Geo-centric, top 2000 | 64.7/70.7 | 63.4/66.7 | 68.8/76.5 | 14.7/48.2 | 0.42 |
| Combined language models, count merging | | | | | |
| Local area, top 2000 | 66.7/72.2 | 69.2/71.5 | 67.8/75.7 | 22.5/54.0 | 0.50 |
| Geo-centric, top 2000 | 67.7/72.6 | 68.3/70.5 | 70.4/77.7 | 13.2/46.9 | 0.44 |

Table 1: Results on mobile 411 data (total listings 14235; national listings 4679; local listings 2495; listings with street addresses 1163)

Now we look at the performance of different approaches to nationwide and local language modeling. First we compare the two nationwide LMs. As expected, we see that the overall sentence accuracy for the National LM is more than twice as high as that of the Top 2000 LM, but the recognition time is more than twice as slow. Next we compare the two local language modeling approaches. We see that geo-centric LMs achieve a 4.4% absolute increase in overall sentence accuracy compared to local area LMs and a 5.5% increase in sentence accuracy on local listings, while using less processing time.

Next we look at combination language models. When we combine local and nationwide LMs using LM union, we get small increases in sentence accuracy for nationwide queries compared to local LMs alone. However, sentence accuracy for local listings decreases. Also, these models use more processing time than the local LMs. When we combine local and national LMs using count merging, we get larger increases in sentence accuracy for nationwide queries over local LMs alone, and smaller decreases for local queries, compared to using LM union. LMs trained using count merging use more processing time than those trained using LM union, but still less than the National LM.

We conclude that: geo-centric language modeling leads to increased recognition accuracy and improvements in recognition time, compared to us-

ing a national language model; geo-centric language modeling leads to increased recognition accuracy and improvements in recognition time, compared to using local area language models; and geo-centric language models can be combined with a “most frequently asked-for” nationwide language model to get increased recognition accuracy on nationwide queries, at the cost of a small increase in recognition time and a slight decrease in recognition accuracy for local listings.

Further analysis of our results showed another interesting phenomenon. While geo-centric LMs achieve higher recognition accuracy than the National LM and local area LMs on nationwide and local queries, recognition accuracy on queries that contain a street name decreases. The likely reason is that small local LMs do not have rich street name coverage and people often do not refer to a street address precisely. A person might use a route number instead of a street name; if a single road has different names at different points they might use the wrong name; or they might use a variation on the actual name. For example, the query “Conoco on South Divison” is correctly recognized by our national LM but not with a geo-centric LM. The closest matching listing in our database for that location is “Conoco Convenience Store on South Boulevard”. We note that we did not make any attempt to generalize over the street names in our LMs, sim-

ply pulling one street name for each listing from the database. Slightly more robust handling of street names may cause this phenomenon to disappear.

6 Conclusions and Future Work

Smart phones are able to give system developers increasingly detailed information about their users. This information can and should be exploited to give improved robustness and performance in customer services. In this paper, we explored the use of location information (from GPS or cell tower triangulation) to improve ASR accuracy in LBVS. We presented an algorithm for geo-centric language model generation that: adapts to the local business density; enables good local listing coverage; and requires only a limited number of language models. We compared the performance of our geo-centric language modeling to an alternative “local” language modeling approach and to a nationwide language modeling approach, and showed that we achieve significant improvements in recognition accuracy (a 4.4% absolute increase in sentence accuracy compared to local area language modeling, and a 16.8% absolute increase compared to the use of a national language model) with significant speedup.

We are currently testing our geo-centric language models in a LBVS prototype. In future work, we will optimize the parameters in our algorithm for geo-centric LM computation and merging. We also plan to explore the impact of integrating language modeling with search, and to examine the impact of these different language modeling approaches on performance of a trainable dialog manager that takes n-best output from the speech recognizer.

References

A. Acero, N. Bernstein, R. Chambers, Y. C. Ju, X. Li, J. Odell, P. Nguyen, O. Scholz, and G. Zweig. 2008. Live search for mobile: web services by voice on the cellphone. In *Proceedings of ICASSP*, pages 5256–5259.

M. Bacchiani, F. Beaufays, J. Schalkwyk, M. Schuster, and B. Strope. 2008. Deploying GOOG-411: Early lessons in data, measurement, and testing. In *Proceedings of ICASSP*, pages 5260–5263.

B. Buntschuh, C. Kamm, G. Di Fabbrizio, A. Abella, M. Mohri, S. Narayanan, I. Zeljkovic, R. Sharp, J. Wright, S. Marcus, J. Shaffer, R. Duncan, and

J. Wilpon. 1998. VPQ: a spoken language interface to large scale directory information. In *Proceedings of ICSLP*.

S. Chang, S. Boyce, K. Hayati, I. Alphonso, and B. Buntschuh. 2008. Modalities and demographics in voice search: Learnings from three case studies. In *Proceedings of ICASSP*, pages 5252–5255.

R. Collingham, K. Johnson, D. Nettleton, G. Dempster, and R. Garigliano. 1997. The Durham telephone enquiry system. *International Journal of Speech Technology*, 2(2):113–119.

K. Georgila, K. Sgarbas, A. Tsopanoglou, N. Fakotakis, and G. Kokkinakis. 2003. A speech-based human-computer interaction system for automating directory assistance services. *International Journal of Speech Technology*, 6:145–159.

V. Goffin, C. Allauzen, E. Bocchieri, D. Hakkani-Tur, A. Ljolje, S. Parthasarathy, M. Rahim, G. Riccardi, and M. Saraclar. 2005. The AT&T Watson speech recognizer. In *Proceedings ICASSP*.

V. Gupta, S. Robillard, and C. Pelletier. 1998. Automation of locality recognition in ADAS plus. In *Proceedings of IVITA*, pages 1–4.

C. Kamm, C. Shamieh, and S. Singhal. 1995. Speech recognition issues for directory assistance applications. *Speech Communication*, 17(3–4):303–311.

X. Li, Y. C. Ju, G. Zweig, and A. Acero. 2008. Language modeling for voice search: a machine translation approach. In *Proceedings of ICASSP*, pages 4913–4916.

P. Natarajan, R. Prasad, R. Schwartz, and J. Makhoul. 2002. A scalable architecture for directory assistance automation. In *Proceedings of ICASSP*, pages 21–24.

T. Paek, B. Thiesson, Y. C. Ju, and B. Lee. 2008. Search Vox: Leveraging multimodal refinement and partial knowledge for mobile voice search. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 141–150.

G. Riccardi, R. Pieraccini, and E. Bocchieri. 1996. Stochastic automata for language modeling. *Computer Speech and Language*, 10(4):265–293.

O. Scharenborg, J. Sturm, and L. Boves. 2001. Business listings in automatic directory assistance. In *Proceedings of Eurospeech*, pages 2381–2384.

F. Seide and A. Kellner. 1997. Towards an automated directory information system. In *Proceedings of Eurospeech*, pages 1327–1330.

Y. Y. Wang, D. Yu, Y. C. Ju, G. Zweig, and A. Acero. 2007. Confidence measures for voice search applications. In *Proceedings of INTERSPEECH*.

D. Yu, Y. C. Ju, Y. Y. Wang, G. Zweig, and A. Acero. 2007. Automated directory assistance system – from theory to practice. In *Proceedings of INTERSPEECH*, pages 2709–2712.