GEOGRAPHIC LANGUAGE MODELS FOR AUTOMATIC SPEECH RECOGNITION

Xiaoqiang Xiao, Hong Chen, Mark Zylak, Daniela Sosa, Suma Desu
Mahesh Krishnamoorthy, Daben Liu, Matthias Paulik, Yuchen Zhang

Apple Inc., U.S.A

ABSTRACT

In this paper, we propose improving automatic speech recognition (ASR) accuracy for local points of interest (POI) by leveraging a geo-specific language model (Geo-LM). Geographic regions are defined according to U.S. Census Bureau Combined Statistical Areas. Depending on the user's associated geographic region, for each user a class based Geo-LM is constructed dynamically within a difference-LM based weighted finite state transducer (WFST) system. The benefits of this approach include: improved accuracy for local POI name recognition, flexibility in training, and efficient LM construction at runtime. Our experiments show that the proposed Geo-LM achieves an average of over 18% relative word error rate (WER) reduction on the tasks of local POI search, with no degradation to the general accuracy and very limited latency increase, compared to the baseline nationwide general LM. In addition to accuracy improvement, we also discuss optimization of runtime efficiency.

Index Terms— speech recognition, language model, Geo-LM, class LM, Combine Statistical Area

1. INTRODUCTION

In recent years, speech recognition accuracy has experienced phenomenal improvements due to the wide adoption of deep learning techniques. There are already claims of ASR achieving human parity in conversational speech recognition [1]. On the other hand, although many state-of-the-art ASR systems perform very well on general recognition tasks, recognition of named entities is still poor due to various challenges. For example, named entities are often so diverse that it is difficult for the regular lexicon to cover all of them. Furthermore, named entities can be unique and rarely seen in training data. As a result, named entities often have low LM probabilities, which makes them hard to recognize.

We believe the next frontier for ASR is the effective utilization of personalized and localized information. In this paper, we are specifically tackling the problem of improving local POI name recognition in mobile devices by utilizing Geo location information. Nowadays speech has become a popular user interface for mobile devices, and Geo location information is readily available. Since mobile users are more likely to search nearby local POIs, by incorporating users' Geo location information into ASR system, the accuracy of local POIs name recognition can be significantly improved.

Using Geo location information has been investigated over the years [2, 3, 4, 5, 6, 7, 8]. Generally there are two ways of defining users' Geo locations for ASR system: One way is to extract Geo information from users' speech that contains a location. For example, when a user says "direction to Helmand Restaurant near Boston," Boston becomes a strong cue to indicate the Geo location of the user. However, this often requires two-pass decoding as described in [2]. Another way is to obtain Geo coordinates directly from mobile devices. As nowadays latitude/longitude coordinates are commonly accessible in mobile devices, it becomes feasible to design location aware ASR system accordingly [4]. In this paper, we utilize Geo coordinates extracted directly from mobile devices for our location aware ASR system design.

One of the problems with using Geo location information is defining the Geo region granularities. Different Geo region granularities such as city, state, zip code, and designated marketing areas (DMA) have been proposed for location aware ASR systems design [3, 5]. In this paper, we propose using combined statistical areas (CSAs) as defined by the U.S. Census Bureau [9]. It is believed CSAs are better in representing the economic and social links within regions, thus more relevant to our applications on mobile devices.

To improve local POI name recognition by leveraging Geo location information, language models can be updated with location specific information. Our proposed Geo-LMs are compact statistical LMs represented as WFSTs, which are dynamically and efficiently spliced into our main language model via on-the-fly replacement during Viterbi search. Essentially our Geo-LMs framework is a type of class LM, which provides an efficient solution for supporting the traditional class LM [10] in a difference-LM based WFST system.

2. GEOGRAPHIC REGIONS

In our proposed system, one Geo-LM is constructed for each defined Geo region. For users within a certain region, the system will load the associated Geo-LM for ASR decoding. Therefore it is important that every defined Geo region has good coverage of local POIs that users search for within the region.

2.1. Analysis of User-POI Distance

To determine the granularity of Geo regions, we analyzed the distances between users and their searched POIs. The latitude/longitude coordinates of the users and their searched POIs are obtained from Siri logs. In order to preserve the privacy of Siri users, the precision of users' latitude/longitude is reduced by randomizing it within the local population. Since each of our targeted regions covers a big diameter, the fuzziness of the data is not an issue. In total we analyzed 4,612,624 users' queries in the United States, and Fig. 1 shows the statistics of the distances between user location and searched POI location. As one can see, nearly 70% of the distances are less than 50 miles, which means the majority of the users are close to the POIs they want to search. The distribution also has a long tail, with about 10% of the queries having larger than 1000 miles of distance between the user and the searched POI. By looking into the long distance queries, we find that many of them are referring to common POIs that may not even be in the U.S.; e.g., Abadan International Airport, which the ASR system with general LM probably

has already performed well on. On the other hand, the short distance queries often refer to local business names which can be quite unique and have low n-gram probabilities in general LM.

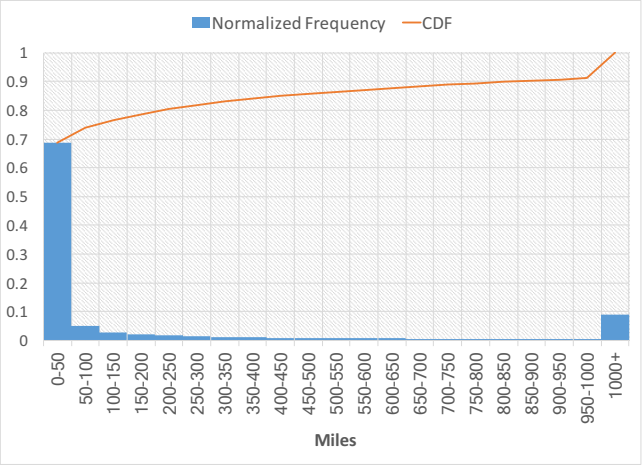

Fig. 1. The normalized frequency and cumulative distribution function (CDF) of the user-POI distances.

2.2. Geographic Regions Definition and Search

Based on the observations in section 2.1, we propose to define Geo regions based on Combined Statistical Areas, which consist of adjacent metropolitan areas that show economic and social links measured by commuting patterns. In total there are 169 CSAs covering 80% of the population in the United States. For each CSA a dedicated Geo-LM will be built. For other areas not covered by CSAs, a single "global" Geo-LM will be built. The details of how those Geo-LMs are built will be described in section 3. To efficiently search the CSA for a user, we store a latitude/longitude lookup table from rasterized Cartographic Boundary Shapefiles provided by U.S. Census Bureau [11]. At runtime, given a user's Geo location, the system can find the associated CSA in the lookup table with constant time complexity, as described in the next section.

2.3. Geographic Regions Data Format

Since all the location lookup and Geo-LM swapping must be done at runtime, processing efficiency and memory usage must be greatly optimized. This section describes the process.

A cylindrically projected world map of regions is stored in a Portable Grey Map (PGM) [12] file. A grey value of each pixel encodes the identity of the associated CSA. The PGM file is accompanied by a JSON metadata file, which defines framing of the bitmap into latitude/longitude coordinates and the mapping array between CSAs and grey values.

The process of finding location-specific models starts from linear transformation of user coordinates $c \in [-180 : 180] \times [-90 : 90]$ into texture coordinates $t \in [0 : 1]^2$. Then grey value of the pixel closest to t can be found, and the corresponding CSA can be retrieved. All steps can be done in constant time, hence the overall CSA lookup time is $O(1)$.

Our distributed cluster runs a number of worker processes on each node to handle multiple clients in parallel. Processes within a node share a single copy of the regions bitmap. Since the bitmap lookups can be done without building additional data structure, except for index of CSAs by grey value, the total RAM consumption associated with lookups is $O(n)$ per process plus one size of the bitmap file covering the U.S. (per node), where n is the number of CSAs. The total PGM file size is $17.4MB$, which can easily fit into small fraction of RAM in distributed cluster node.

3. ALGORITHM

The underlying ASR system is based on a WFST based decoder as first described in [13]. The decoder employs the difference LM principle similar to [14, 15]:

$$HCLG_{small} \circ F \tag{1}$$

where $\circ$ denotes on-the-fly composition, H contains HMM definitions, C represents the context dependency, L is the lexicon, G_{small} is a small, typically uni-gram, language model, and

$$F = G^{-}_{small} \circ G_{big} \tag{2}$$

where G^{-}_{small} is negated score version of G_{small} and G_{big} is a big language model. To provide efficiency and computational time savings, $HCLG_{small}$ is constructed prior to runtime via offline composition. At runtime, the static cascade $HCLG_{small}$ is dynamically composed with the difference grammar "$G^{-}_{small} \circ G_{big}$" on the fly.

As described in [13], the decoder includes support for user-specific vocabularies by leveraging a G_{big} class language model for which we dynamically replace class non-terminals with intra-class grammars [16]. Our Geo-LM construction leverages these principles as explained in the following sections.

3.1. Geo-LM Construction

One straightforward way of constructing Geo-LM is to build the whole LM for each individual Geo region by interpolating the general Geo independent LM and the LM trained on the Geo dependent text only. However, it would result in a large number of large sized Geo-LMs, prohibiting the server from preloading all models into system memory as required by our application of real-time ASR decoding. As a result, this solution is not practical for deployment and therefore not considered in this paper. Instead, we propose constructing Geo-LM based on class LM described as follows.

Our G_{big} class LM results from an offline interpolation of various component language models build over different types of training data sources. Two important data sources in this specific context are 1) production traffic, i.e. automatically transcribed data; and 2) artificially-created text data. The automatic transcriptions produced by our system include specific markers that allow us to identify words or phrases that stem from intra-class grammars, in the following referred to as slot LMs G_{slot}. Based on these markers, the training data can be curated to replace such words and phrases with non-terminal class labels, e.g. \CS-POI. For artificial training text data creation, we leverage simple templates as shown in Table 1. Component language models trained on artificial data can play a very important role when introducing new non-terminal class labels for the first time. Fig. 2 illustrates a toy example of G_{big} that represents the templates in Table 1.

Prior Probabilities	Templates
0.5	directions to \CS-POI
0.3	where is \CS-POI
0.2	find the nearest \CS-POI

Table 1. Examples of templates and their prior probabilities.

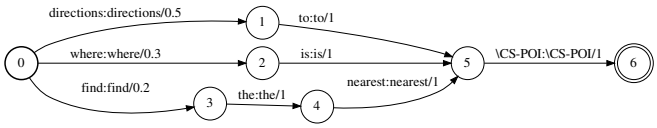

Fig. 2. A toy example of WFST LM with class non-terminals.

G_{slot} models entities of specific category, which is POI in the case of Geo-LM. In the proposed system, one G_{slot} is built for each Geo region. The training data for each G_{slot} are the names of local POIs in the corresponding region. For illustration, we provide a toy example of G_{slot} containing only three POIs with priors as shown in Fig. 3.

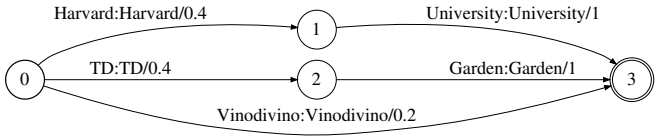

Fig. 3. A toy example of WFST representing slot LM.

Training G_{slot} as a statical n-gram LM enables it to model the variations in POI names; e.g., both "Harvard University" and "Harvard" can be modeled in the G_{slot} as long as "Harvard University" exists in the training data. In our system, priors are derived based on distributions observed in production traffic.

In this work, the lexicon L is the union of lexicons for G_{big} and all available G_{slot} LMs. We employ an in-house developed grapheme-to-phoneme (G2P) system to derive pronunciations automatically if a word in the POI name is not already in our decode lexicon.

At runtime, we dynamically replace the class non-terminals via on-the-fly replacement with the respective matching, and appropriately scaled, G_{slot} LM for the G_{big}. Fig. 4 shows an example for how the final WFST would look like if offline replacement were used to statically create the graph. Since we construct the static cascade $HCLG_{small}$ at model building time, the class non-terminals in G_{small} are replaced by a small G_{slot} LM built from the POIs of all regions. It should be noted that the scaling factors used to uniformly scale the G_{slot} LM log-likelihoods play an important role.

The described framework allows for flexible updates to the overall system. To update POIs or add new regions, one simply needs to rebuild the G_{slot} LMs and potentially also $HCLG_{small}$ and G^{-}_{small}. This can be done very quickly and efficiently due to the small sizes of the respective LMs involved. The flexibility of G_{slot} updates is essential for the sustainability of our application due to the rapid change in POIs, such as the opening/closing of businesses and continuously changing popularity. In addition, since the sizes of G_{slot} LMs are small, the proposed framework allows all models to be preloaded into system memory during server initialization.

4. EXPERIMENTS

In this section we report the benchmark evaluations of the proposed Geo-LM and general LM on the task of POI recognition in the United States. In all experiments, the same AM of hybrid convolutional neural network (CNN)-hidden Markov model (HMM) [17] is used, which is trained with filter bank features from about 3000 hours of English speech data using cross-entropy and subsequent bMMI objective functions [18].

4.1. Data

4.1.1. Training Data

For constructing the baseline general LM, the training data (D1) contains a variety of data sources of collected privacy-preserved user data. The training data for building the component LMs in the proposed Geo-LM is composed of D1 and artificial use case templates containing the POI class symbol as described in section 3.1. The use case templates are equally weighted in our setup. To build slot LMs within Geo-LM, we extract the searched POIs from the daily updated Apple Map Search logs. Based on the Geo locations of the extracted POIs, they are divided into 170 parts to build slot LMs for 169 CSAs and 1 "global" area. The "global" slot LM will be loaded for users outside the 169 CSAs. The priors of POIs are set according to the engagement frequency in the logs. The size comparison of general LM and Geo-LM is summarized in Table 2, where the master LM represents the G_{big} class LM with class non-terminals, i.e. before the replacement with slot LM G_{slot}.

LM		#n-grams (millions)	
		G_{small}	G_{big}
General LM		5.5	9.3
Geo-LM	Master LM	5.5	9.3
	Slot LM	2.0	0.7 (avg.)

Table 2. Number of n-grams in general LM and Geo-LM. The size of slot big LM is the average size over 170 regions.

4.1.2. Test Data

There are two types of test data used in our experiments:

1. Real-world user data randomly selected from Apple Siri production traffic in the United States. There are two test sets created from those real-world user data which include (**T1**) POI search test set, which consists of 17587 utterances in the local POI search domain; and (**T2**) general test set, which consists of 9955 utterances that don't include POIs.
2. Internally recorded local POI search test set (**T3**). We picked 8 major U.S. metropolitan regions and for each region the top 1000 most popular POIs (with megachains filtered out) based on Yelp reviews are selected. For each POI, three utterances are recorded with or without the carrier phrase "direction to" from three different speakers.

4.2. Geo-LM Configurations

For training the Geo-LM, the G_{small} is trained as a unigram LM, where the training data for the small G_{slot} LM includes all POIs for the 170 Geo regions described in Section 4.1.1. The G_{big} is trained as a 4-gram LM. Depending on the location of the test data, the big G_{slot} LM for the associated Geo region is used. As described in 3.1, the G_{slot} LMs are scaled before the class non-terminals replacements for Geo-LM. We tuned on some development dataset and set the scaling factors for the small G_{slot} LM and the big G_{slot} LMs to be 0.95 and 0.7 respectively.

4.3. Experimental Results

We first conducted experiments on the real-world user test set. The results in Table 3 show that using the class based Geo-LM leads to significant relative word error rate reduction (WERR) of 18.7% on the POI search test set (**T1**), with no accuracy degradation on the general test set (**T2**). Since the POI search test set (**T1**) is randomly

Fig. 4. A toy example of final WFST LM replacing class non-terminals with slot LM.

sampled from production traffic, it contains many megachains such as "Walmart" and "Home Depot" that the general LM has already covered well. To benchmark the performance of name recognition on more difficult local POIs, we then tested on the local POI search test set (**T3**), which doesn't contain any megachains. The results in Table 4 show that the general LM performs poorly on this test set (**T3**), and the proposed Geo-LM significantly improve WER by relatively over 40% for all 8 cities. We also measure the speed of the two benchmark systems, and observe the class based Geo-LM only increase the latency marginally by less than 10 milliseconds.

Test Set	General LM WER(%)	Geo-LM WER(%)	relative WERR(%)
General (**T2**)	6.2	6.2	0
POI search (**T1**)	15.5	12.6	18.7

Table 3. WER comparisons between general LM and Geo-LM on real-world user test sets.

Test Set	General LM WER(%)	Geo-LM WER(%)	relative WERR(%)
Boston	24.3	13.7	43.6
Chicago	26.3	15.2	42.2
Los Angeles	24.4	12.6	48.4
Minnesota	19.6	10.7	45.4
New York	27.3	15.7	42.5
Philadelphia	25.8	15.0	41.9
Seattle	24.8	13.8	44.4
San Francisco	26.5	15.1	43.0

Table 4. WER comparisons between general LM and Geo-LM on internally recorded local POI search test set (**T3**).

To understand the behaviors of our proposed Geo-LM, we check if G_{slot} LMs are indeed triggered during decoding on the POI search test set (**T1**). Table 5 shows 77% of the POI search utterances have G_{slot} triggered, and the WER on those utterances (9.7%) is significantly better than the WER on utterances without triggering G_{slot} (23%). These results indicate that the decoding process did successfully trigger G_{slot} in Geo-LM for the majority of POI search test utterances, and triggering G_{slot} is the key factor for accuracy improvement for the POI name recognition.

POIs search test subset	percentage of utterances	WER(%)
G_{slot} triggered	77%	9.7
G_{slot} not triggered	23%	23

Table 5. Performance of Geo-LM on the POI search test set (**T1**) divided according to whether slot LMs are triggered during decoding.

Further error analysis shows the proposed Geo-LM system is still making errors mainly due to the following causes: 1) Poor pronunciation for foreign POI names. For example, although covered by G_{slot}, the POI "Les Zygomates" is misrecognized as "Let's Zika matt" due to poor pronunciation guessing by our G2P model; 2) User searches for POIs outside their associated CSA area and master LM does not cover them well; 3) Confusability with common ngrams when users' requested POI names do not match ones in G_{slot}. For example, while "Roam Artisan Burgers" exists in G_{slot}, users often say "Roam Burgers" and the recognizer misrecognizes it as "Rome burgers". With these issues addressed, the triggering rate of G_{slot} is likely to increase, potentially leading to better accuracy of POI name recognition. To address these issues, in future works we plan to improve our G2P model by enriching the training data, and improve coverage and name variance of POIs in G_{slot} of our Geo-LM.

To justify our choice of CSA over other common Geo regions granularities, we first compared CSAs with states as Geo regions in our proposed Geo-LM framework. We trained two sets of Geo-LMs based on CSAs and states respectively, and tested on a subset (3280 utterances) of the POI search test set (**T1**) where the users' associated CSA and state boundaries are different. The results in Table 6 show that the CSA regions based Geo-LM performs better than the state regions based Geo-LM with relative WERR of about 3%, and they both outperform General LM baseline by over 20%. The reason state regions perform worse than CSA regions for Geo-LM could be that state regions often go beyond the normal size of frequent commuter routes. For example, users from San Francisco in Northern California seldom search and drive 8+ hours directly to the POIs in Los Angeles in Southern California, except for in less-frequent scenarios such as a road trip. It is too small of a use case to justify expanding the scope of Geo-LM regions to cover areas of the size of the whole state of California, which would introduce ambiguities and deteriorate results for more important local searches. On the other hand, smaller Geo regions such as zip codes and cities would not be sufficient because users often commute beyond the scope of these boundaries as shown in the statistics in Fig. 1.

LM	Geo region granularity	WER(%)	relative WERR(%)
General LM	—	14.4	—
Geo-LM	state	11.2	22.2
	CSA	10.8	25

Table 6. WER comparisions: general LM and Geo-LMs with state and CSA Geo region granularities on POI search task.

5. CONCLUSIONS

In this work, we present a Geo-LM framework that provides the benefits of flexible training, efficient LM construction at run-time, and significant ASR accuracy improvement over the general LM on the task of local POI recognition. Our experiments prove that localized information can benefit ASR significantly, leading to over 18% WERR on the tasks of local POI search. Due to the limited impact to system speed, regional coverage can be continuously improved. But it is still essential to provide a "global" Geo-LM in addition to regional LMs so that ASR can handle long distance queries and cases when users are located outside any supported regions.

Finally, we'd like to note that the method and system proposed here is language independent. The expansion of Geo-LM support for other locales besides US English should be straightforward.

6. REFERENCES

[1] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," in *arXiv preprint arXiv:1610.05256*, 2016.

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

[3] E. Bocchieri and D. Caseiro, "Use of geographical meta-data in ASR language and acoustic models," in *Proceedings of ICASSP*, 2010, pp. 5118–5121.

[4] A. Stent, I. Zeljković, D. Caseiro, and J. Wilpon, "Geo-centric language models for local business voice search," in *Proceedings of NAACL*, 2009, pp. 389–396.

[5] C. Chelba, X. Zhang, and K. Hall, "Geo-location for voice search language modeling.," in *Interspeech*, 2015, pp. 1438–1442.

[6] J. Feng, "Location-aware query parsing for mobile voice search," in *Proceedings of ICASSP*, 2011, pp. 5728–5731.

[7] C. Van Heerden, J. Schalkwyk, and B. Strope, "Language modeling for what-with-where on GOOG-411," in *Interspeech*, 2009, pp. 991–994.

[8] G. Ye, C. Liu, and Y. Gong, "Geo-location dependent deep neural network acoustic model for speech recognition," in *Proceedings of ICASSP*, 2016, pp. 5870–5874.

[9] U.S. Census Bureau, "Combined Statistical Areas of the United States and Puerto Rico," 2015.

[10] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[11] U.S. Census Bureau, "Cartographic Boundary Shapefiles," 2015.

[12] J. Poskanzer, "Netpbm grayscale image format, Available: http://netpbm.sourceforge.net/doc/pgm.html," 2016.

[13] M. Paulik, "Improvements to the pruning behavior of DNN acoustic models," in *Interspeech*, 2015, pp. 1463–1467.

[14] H. Dolfing and I. Hetherington, "Incremental language models for speech recognition using finite-state transducers," in *Proceedings of ASRU*, 2001, pp. 194–197.

[15] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al., "The Kaldi speech recognition toolkit," in *Proceedings of ASRU*, 2011, pp. 1–4.

[16] M. Paulik and R. Huang, "Method for supporting dynamic grammars in WFST-based ASR," Nov. 22 2016, US Patent 9,502,031.

[17] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[18] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks.," in *Interspeech*, 2013, pp. 2345–2349.