Geo-location for Voice Search Language Modeling

Ciprian Chelba, Xuedong Zhang, Keith Hall

Google
1600 Amphitheatre Parkway,
Mountain View, CA 94043, USA

{ciprianchelba,xuedong,kbhall}@google.com

Abstract

We investigate the benefit of augmenting with geo-location information the language model used in speech recognition for voice-search.

We observe reductions in perplexity of up to 15% relative on test sets obtained from both typed query data, as well as transcribed voice search data; on a subset of the test data consisting of “local” queries — search results displaying a restaurant, some address, or similar — the reduction in perplexity is even higher, up to 30% relative.

Automatic speech recognition experiments confirm the utility of geo-location information for improved language modeling. Significant reductions in word error rate are observed both on general voice search traffic, as well as “local” traffic, up to 2% and 8% relative, respectively.

Index Terms: language modeling, geo-location, query stream, voice search

1. Introduction

Mobile is poised to become the predominant platform over which people are accessing the World Wide Web. Recent developments in speech recognition, backed by high quality speech signal acquisition on smartphones and tablets are presenting the users with the choice of speaking their web search queries instead of typing them. A critical component of an automatic speech recognition (ASR) system targeting web search is the language model (LM). Previous work [1]-[2] described the usefulness of the typed query stream for voice-search language modeling.

Another important signal that accompanies most mobile search queries is the geo-location, available at various levels of granularity: zip code, locality/city, state, country, or sometimes none at all.

We find that a simple way of integrating a fairly coarse geo-location signal, namely identifying one of about 200 designated marketing areas (DMA, [3]), provides a significant quality boost to the LM, as measured in both perplexity (PPL) and word-error-rate (WER); the impact is even higher if we restrict measurements to the “local” subset of the query traffic.

1.1. Related Work

Geo-location for ASR language models has been found useful for various applications over the years: [4], [5], [6], [7]. The most significant difference between such prior work and ours lies in the data being used for training and evaluating the language model: we make use of query logs annotated with geo-location information, as provided by mobile clients.

1.2. Privacy Considerations

Before delving into the technical details, we wish to clarify the privacy aspects of our work with respect to handling user data.

All of the query data used for training and testing models is strictly anonymous; the queries bear no user-identifying information. The only data saved after training are vocabularies and n-gram counts.

2. Geo-location for Language Modeling

Generally speaking, the geo-location signal can contain information at various levels of resolution: gps, zip code, locality/city, state, country, or sometimes none at all.

As a first approximation, one can assume a hierarchical structure on the various geo-location resolution levels and build a tree that partitions the training and test data into disjoint subsets at each level in the tree[1]; the root partition of the training data consists of the data used for building the baseline LM. Data sources that do not have geo-location information are used to augment the baseline/root LM.

We build a separate LM at each node in the geo-location clustering tree. N-gram counts are collected using the vocabulary $\mathcal{V}$ of the root LM. Since not all words are present in the training data at a given node g and we wish to interpolate language models up the geo-location clustering tree, we need to account for the words W that are in the set $\mathcal{V} \setminus \mathcal{V}_g$. A simple way to do that is to take away probability mass from the unknown word (UNK) and spread it uniformly over the $\mathcal{V} \setminus \mathcal{V}_g$ set:

$$P(w|h,g) = \begin{cases} \tilde{P}(w|h,g), w \in \mathcal{V}_g \setminus \{UNK\} \\ \alpha \cdot \tilde{P}(UNK|h,g), w \in \{UNK\} \\ (1-\alpha) \cdot \tilde{P}(UNK|h,g) \cdot \frac{1}{|\mathcal{V} \setminus \mathcal{V}_g|}, w \in \mathcal{V} \setminus \mathcal{V}_g \end{cases}$$

This probability assignment ensures that the LM at each node g in the geo-location clustering tree is properly normalized over the root vocabulary $\mathcal{V}$. Whenever measuring PPL values, we check empirically that the probabilities sum up to 1.0 for a few contexts chosen at random positions in the test data.

With this in place, we can now interpolate from fine to coarse geo-locations $g_1 \prec g_2 \prec \ldots \prec g_{root}$, as available with a given test query, typed or spoken:

$$P(w|h, g_1 \prec g_2 \prec \ldots \prec g_{root}) = \sum_k \lambda_k \cdot P(w|h, g_k)$$

[1] When this is not true, e.g. a zip code straddling the state line, some heuristic assignment into the higher order location is made.

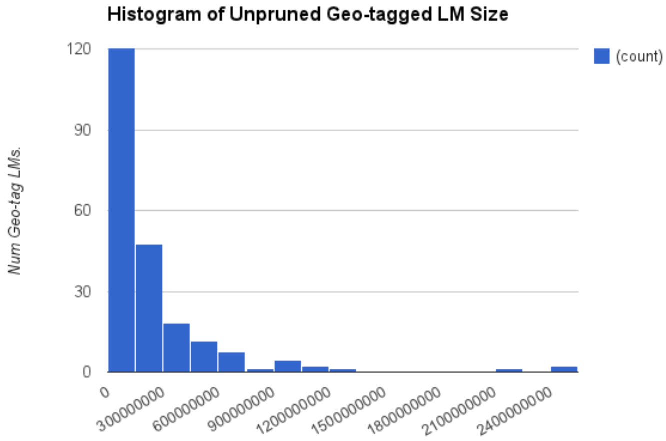

Figure 1: DMA LM histogram before pruning.

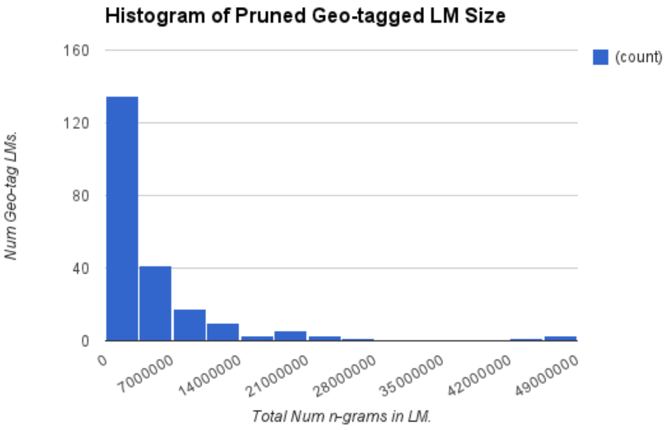

Figure 2: DMA LM histogram after pruning to about 1 billion n-grams.

In early experiments we compared various types of geo-location information and settled on using a simple clustering tree consisting of the root and the designated marketing areas (DMA, [3]) for the leaves:

$$\begin{aligned} P(w|h, g_{DMA} \prec g_{root}) &= \lambda \cdot P(w|h, g_{root}) + \\ & \quad (1-\lambda) \cdot P(w|h, g_{DMA}) \end{aligned} \tag{1}$$

The perplexity improvement from using more detailed geo-location and a deeper tree (zip code, city/locality, state, US/root) did not justify the increased complexity.

2.1. Pruning

For the on-the-fly rescoring experiments reported in Section 3.2.3 we need to prune the geo-LM in Eq. (1). The direct use of Stolcke entropy pruning [8] becomes far from straightforward, especially given that we would like to keep the root and DMA LMs separate, such that the g_{root} n-grams are reused across all g_{DMA} geo-locations.

A very simple approach is to prune each model (g_{root} and g_{DMA} for all DMA values) by specifying a relative decrease from its unpruned size. This does not take into account the potential overlap between g_{root} and one or more g_{DMA} tags.

In an attempt to deal better with such n-grams we also explored an alternative whereby we first prune the g_{root} LM down to a desired size, after which we test each n-gram in a given g_{DMA} LM using the simplified pruning statistic $D(w|h, g_{DMA} \prec g_{root})$ motivated by the inner term in the Kullback-Leibler divergence summation:

$$\begin{aligned} D(w|h, g_{DMA} \prec g_{root}) &= P(h|g_{DMA} \prec g_{root}) \cdot \\ & \quad P(w|h, g_{DMA} \prec g_{root}) \cdot \\ & \quad |\log P(w|h, g_{DMA} \prec g_{root}) - \\ & \quad \log P'(w|h, g_{DMA} \prec g_{root})| \\ P'(w|h, g_{DMA} \prec g_{root}) &= \lambda \cdot P(w|h, g_{root}) + \\ & \quad (1-\lambda) \cdot P(w|h', g_{DMA} \prec g_{root}) \\ P(w|h', g_{DMA} \prec g_{root}) &= \alpha(h, g_{DMA}) \cdot P(w|h', g_{DMA}) \end{aligned}$$

where h' denotes the back-off context for h, $\alpha(h, g_{DMA})$ is the back-off weight in the g_{DMA} LM and $P(h|g_{DMA} \prec g_{root})$ is computed by the chain rule using Eq. (1) and the pruned g_{root} LM.

In our PPL and WER experiments using root and DMA specific LMs pruned aggressively down to 1 billion n-grams, respectively, we did not observe any significant difference between the two pruning methods, so we opted for using the first, simpler approach.

3. Experiments

For training the language model we use a variety of sources, the largest one by far consisting of typed queries arriving at the `google.com` front-end from mobile devices; these are also annotated with geo-location and we use them for building DMA-specific LM.

The language models are trained using an extension of the large LM tools [9] that builds a set of language models in a single pass over the training data and stores them in a single distributed data structure, later used for serving them at run-time.

The vocabulary we use for the root LM consists of 3.4 million words; the root LM is a Katz [10] 5-gram trained on about 695 billion words of training data from a diverse set of sources and pruned down to 15 billion n-grams (3.4/1969.6/7425.5/4357.8/1115.1 million 1/2/3/4/5-grams, respectively). The 211 DMA 5-gram LMs are trained on 287 billion words; their size varies from 2.8 million n-grams to 2.4 billion n-grams; the vocabulary size varies between 99 thousand and 2.2 million words; the total number of n-grams across all DMA LMs is 48 billion n-grams.

Figs. 1-2 show histograms for the unpruned and pruned DMA LM sizes, respectively. The two histograms look very similar, suggesting a power-law distribution for the amount of data and LM size across DMAs: a couple of DMAs containing major cities (Los Angeles, New York) have a lot of data, and correspondingly large DMA LMs; about half of the DMAs fall in the left-most bin, with the smallest amounts of data and LMs, either before of after pruning.

3.1. Perplexity Experiments

We evaluate PPL improvements on mobile typed query data (US TYPED) from the entire US and on manual transcriptions for

Test Set	OoV (%)	Perplexity		
		US	US+DMA	rel. reduction (%)
US TYPED	0.7	86	78	9
US TYPED/local	0.3	85	63	26
SF BAY	0.2	88	75	15
SF BAY/local	0.08	94	66	30
US	0.2	115	89	23
Pruned (1 billion n-grams) US and DMA LMs				
SF BAY	0.2	98	85	13
SF BAY/local	0.08	108	77	29
US	0.2	131	103	21

Table 1: Out-of-Vocabulary (OoV) rates and Perplexity values for baseline (root) LM built on the entire training data for the US LM, as well as geo-LM interpolating the US LM with a DMA specific one (US+DMA); the interpolation weight λ in Eq. (1) is fixed to 0.5. ; relative reductions in PPL are measured between the US and the US+DMA LMs on each line.

ASR test sets consisting of spoken queries from the San Francisco Bay Area (SF BAY) and the entire United States (US), respectively. For both US TYPED and SF BAY test sets we also evaluate on the "local" subset of the query stream: queries whose search results display a restaurant, some address, or similar; the US test set is somewhat locally biased, as described in Section 3.2.3.

The US TYPED, SF BAY and US test sets consists of about 1.8 billion, 217 thousand, 133 thousand words, respectively. The "local" subset consists of about 9% of the entire test set in both US TYPED, SF BAY cases. About 95% of the queries encountered in training, or the US TYPED test set[2] have a geo-location annotation; all queries in the SF BAY, US test sets have geo-location annotation.

Table 1 shows the results. EM estimation for the interpolation weight λ in Eq. (1) showed marginal improvements in PPL over the initial value of 0.5, so we fixed that throughout our experiments. The geo-location enhanced LM provides significant reductions in PPL on the query stream, up to 15% relative; the "local" subset benefits even more from the geo-location, up to 30% relative reduction in PPL. We do not list PPL values for the DMA only configuration, since it always exceeds the US (root) value. Since for on-the-fly WER experiments we use pruned LMs, we also evaluated the PPL of both US and US+DMA LMs after pruning both the US and the set of DMA LMs down to a total of approximatively 1 billion n-grams, respectively. Each of the DMA LMs was pruned individually by setting a target size relative to unpruned.

3.2. ASR Experiments

3.2.1. Lattice Rescoring Experiments

As a first attempt at using the geo-location LM in voice-search ASR we tried rescoring lattices generated with a first pass LM that is geo-location agnostic. We could observe only modest improvements in accuracy: no gain on SF BAY and only 0.1% abs reduction in WER on SF BAY/local, even when using search and lattice beams that were well above real-time. This was surprising, given the large perplexity reductions observed, in particular on the "local" subset.

To diagnose this unexpected behavior, we built a first pass LM targeted at the SF BAY DMA by only using query data originating in the SF BAY DMA in our mix of training data sources; lattices produced by this LM are then rescored using either a US or US+DMA large LM.

Such a system produced significant reductions in WER of -0.2% abs on the SF BAY test set and -1.2% or -1.4% abs on the "local" subset, respectively, showing the large potential improvements attainable when using geo-location in the LM.

To rule out any possible bug in our lattice rescoring setup, we also partitioned each of the SF BAY, SF BAY/local test sets into two subsets:

- an "*almost there*" subset containing the utterances where the correct transcription was present in the top 10-best hypotheses output by the 1-st pass LM
- a "*not yet*" subset containing the remaining utterances, where the correct transcription was not present in the top 10-best hypotheses output by the 1-st pass LM.

Decoding each of these subsets using the SF BAY DMA system we observe that all the improvements come from the "*not yet*" subsets and there are virtually no improvements on the "*almost there*" subsets of both SF BAY and SF BAY/local.

We believe that these two experiments show conclusively that the geo-location LM needs to be integrated closer to the 1-st pass of an FST-based ASR system [11] using a geo-location agnostic CLG, instead of relying on lattice rescoring.

3.2.2. On-the-fly Lattice Rescoring and Generation

Deploying a system where traffic is routed to decoders running DMA specific 1-st pass LMs is not an appealing solution, so we continued investigating the use of a LM that switches DMA context on a per-recognition request basis. To bring the geo-location LM closer to the 1-st pass, we used the on-the-fly lattice rescoring architecture suggested by [12]. This allows us to use the geo-location agnostic LM to drive the Viterbi search while applying the geo-location LM as the search space is expanded. Our decoder follows the general form of the fast dynamic decoding approach described in [13]. We adapt the rescoring approach of [12] to incorporate the hashed path-histories used in the [13] decoding algorithm. The path-histories are used to allow for dynamic determinization when generating the output lattice. The resulting algorithm requires keeping a meta-state which encapsulates both the unique path-histories the relevant geo-location LM states.

[2] The PPL calculation includes queries that do not have a geo-location, in which case $\lambda = 1$ in Eq. (1).

Language Model			Test Set			
			SF BAY		**SF BAY/local**	
1-st pass	on-the-fly	lattice rescoring	WER (%)	rel. reduction (%)	WER (%)	rel. reduction (%)
US	—	US	9.3	—	13.0	—
US	US	US	9.2	1	12.5	4
US	US+DMA	US	9.1	2	12.3	5
US	DMA	US	9.2	1	12.2	6
US	US+DMA	US+DMA	9.1	2	12.1	7
US	DMA	US+DMA	9.2	1	12.0	8

Table 2: WER results on SF BAY and SF BAY/local test sets in various configurations; for the on-the-fly and lattice rescoring LMs, we denote with US LMs where $\lambda = 1.0$, US+DMA where $\lambda = 0.5$ and DMA where $\lambda = 0.0$ in Eq. (1), respectively.

Language Model			Test Set	
			US	
1-st pass	on-the-fly	lattice rescoring	WER (%)	rel. reduction (%)
US	—	US	10.4	—
US	DMA	US	10.1	3
US	DMA	US+DMA	10.1	3

Table 3: WER results on locally-biased US-wide test set in various configurations; for the on-the-fly and lattice rescoring LMs, we denote with US LMs where $\lambda = 1.0$, US+DMA where $\lambda = 0.5$ and DMA where $\lambda = 0.0$ in Eq. (1), respectively.

3.2.3. *On-the-fly Rescoring Experiments*

We now use three LMs:

- a relatively small (100M n-grams) geo-location agnostic first pass LM used for CLG compilation;
- a medium size LM (1B n-grams) used for on-the-fly rescoring and lattice generation; could be either a DMA specific LM, $\lambda = 0.0$ in Eq. (1), or a US-wide LM, $\lambda = 1.0$ in Eq. (1);
- a large distributed geo-location LM used for lattice rescoring.

The scores from the three LMs are combined using log-linear interpolation in two stages:

- the LM cost on CLG arcs is mixed with the on-the-fly LM using equal weights (0.5) and saved as the LM cost on lattice arcs
- the LM cost on lattice arcs is mixed with the distributed LM cost when doing lattice rescoring, again using equal weights (0.5)

In either stage the state space is computed by taking the cross product of the state spaces of the LMs being combined.

Table 2 shows the WER results in various configurations; for the on-the-fly and lattice rescoring LMs, we denote with US LMs where $\lambda = 1.0$, US+DMA where $\lambda = 0.5$ and DMA where $\lambda = 0.0$ in Eq. (1), respectively.

We observe significant reductions in WER on both the SF BAY and the "local" subsets, respectively; the relative improvement on the "local" subset is quite large, close to 8% relative.

It is worth highlighting the large gain on the "local" subset obtained by simply adding the 1-billion n-gram LM in the on-the-fly rescoring pass; simply doing this would take us half-way to the best geo-location augmented system.

As a last set of experiments we transcribed a set of utterances from general US-wide traffic, after asking human raters to filter out voice-search queries that were *definitely not of "local" nature*. The resulting test set was transcribed 3-way and utterances on which two or more annotators agreed were kept, resulting in a test set with 24618 utterances and 111935 words. Table 3 presents the results, showing significant improvements in overall WER. We have also compared the system on the last row of Table 3 against the baseline by sampling from the difference set and asking human raters which system performs better. With high confidence (p-value below 0.1%) the raters preferred the geo-location augmented one; the same was true for the system on the second row of Table 2.

4. Conclusions and Future Work

Geo-location is a very useful signal for improving the quality of the LM used for voice search. Significant reductions in word error rate are observed both on general voice search traffic, as well as "local" traffic, up to 2% and 8% relative, respectively.

Better partitioning of the geo-annotated data, as well as pruning of geo-location LMs jointly with the US root LM should be investigated more thoroughly. Various LM adaptation techniques can be employed for combining LMs at various geo-location resolution levels.

An interesting finding is that lattices generated with a geo-location agnostic 1-st pass LM do not contain the paths that a LM augmented with geo-location would be able to improve. To be able to realize the potential of such an LM we need to use it closer to the 1-st pass, in our case using on-the-fly rescoring. It is quite likely that being able to bring the large (distributed) LMs used in lattice rescoring to the on-the-fly rescoring pass will yield even larger gains in accuracy and simplify the current architecture using two rescoring passes. This is another important direction worth exploring in the future.

5. Acknowledgments

We would like to thank Lukasz Heldt and Krzysztof Duleba for help in understanding the variety of geo-location signals available, and their reliability; Thorsten Brants and Colin Young for help on LM training infrastructure; Trevor Strohman, Babak Damavandi for providing the geo-location ASR test sets.

6. References

[1] C. Chelba, J. Schalkwyk, T. Brants, V. Ha, B. Harb, W. Neveitt, C. Parada, and P. Xu, "Query language modeling for voice search," in *Spoken Language Technology Workshop (SLT), 2010 IEEE*, 2010, pp. 127–132.

[2] Ciprian Chelba and Johan Schalkwyk, *Empirical Exploration of Language Modeling for the google.com Query Stream as Applied to Mobile Voice Search*, pp. 197–229, Springer, New York, 2013.

[3] Wikipedia, "Media market," Wikimedia Foundation.

[4] Sheng Chang, Susan J. Boyce, Katia Hayati, Issac Alphonso, and Bruce Buntschuh, "Modalities and demographics in voice search: Learnings from three case studies.," in *ICASSP*. 2008, pp. 5252–5255, IEEE.

[5] Bruce Buntschuh, Candace A. Kamm, Giuseppe Di Fabbrizio, Alicia Abella, Mehryar Mohri, Shrikanth Narayanan, Ilija Zeljkovic, R. D. Sharp, Jeremy H. Wright, S. Marcus, J. Shaffer, R. Duncan, and Jay G. Wilpon, "Vpq: a spoken language interface to large scale directory information.," in *ICSLP*. 1998, ISCA.

[6] M. Bacchiani, F. Beaufays, J. Schalkwyk, M. Schuster, and B. Strope, "Deploying GOOG-411: Early lessons in data, measurement, and testing," in *Proceedings of ICASSP*, April, pp. 5260–5263.

[7] Alex Acero, Neal Bernstein, Rob Chambers, Yun-Cheng Ju, Xiao Li, Julian Odell, Patrick Nguyen, Oliver Scholtz, and Geoff Zweig, "Live search for mobile: Web services by voice on the cellphone," in *Proc. of the Int. Conf. on Acoustics, Speech, and Signal Processing*. April 2008, Institute of Electrical and Electronics Engineers, Inc.

[8] Andreas Stolcke, "Entropy-based pruning of back-off language models," in *Proceedings of News Transcription and Understanding Workshop*, Lansdowne, VA, 1998, pp. 270–274, DARPA.

[9] Thorsten Brants and Peng Xu, "Distributed language models.," in *HLT-NAACL Tutorial Abstracts*, 2009, pp. 3–4.

[10] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," in *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1987, vol. 35, pp. 400–01.

[11] Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri, "OpenFst: A general and efficient weighted finite-state transducer library," in *Proceedings of the Ninth International Conference on Implementation and Application of Automata, (CIAA 2007)*. 2007, vol. 4783 of *Lecture Notes in Computer Science*, pp. 11–23, Springer, `http://www.openfst.org`.

[12] Takaaki Hori, Chiori Hori, Yasuhiro Minami, and Atsushi Nakamura, "Efficient WFST-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 4, pp. 1352–1365, 2007.

[13] George Saon, Daniel Povey, and Geoffrey Zweig, "Anatomy of an extremely fast LVCSR decoder," in *in Proc. Interspeech*, 2005, pp. 549–552.