Garbage Modeling for On-device Speech Recognition

Christophe Van Gysel[1]*, Leonid Velikovich*[2]*, Ian McGraw*[3]*, Françoise Beaufays*[3]

[1]University of Amsterdam, Amsterdam, The Netherlands
[2]Google, Inc., New York, NY USA
[3]Google, Inc., Mountain View, CA USA

cvangysel@uva.nl, leonidv@google.com, imcgraw@google.com, fsb@google.com

Abstract

User interactions with mobile devices increasingly depend on voice as a primary input modality. Due to the disadvantages of sending audio across potentially spotty network connections for speech recognition, in recent years there has been growing attention to performing recognition on-device. The limited computational resources, however, typically require additional model constraints. In this work, we explore the task of on-device utterance verification, wherein the recognizer must transcribe an utterance if it is in a target set or reject it as being out of domain. We present a data-driven methodology for mining tens of thousands of target phrases from an existing corpus. We then compare two common garbage-modeling approaches to utterance verification: a sub-word rejection model and a white-listed n-gram model. We examine a deficiency of the sub-word modeling approach and introduce a novel modification that makes use of common prefixes between targeted phrases and non-targeted phrases. We show good performance in the trade-off between recall and word error rate using both the prefix and white-listed n-gram approaches. Finally, we evaluate the prefix-based approach in a hybrid setting where rejected instances are sent to a server-side recognizer.

Index Terms: automatic speech recognition, language modeling, utterance verification, OOV rejection, garbage modeling

1. Introduction

Word error rates of modern speech recognizers have improved so dramatically in recent years that, for some, the main impediment to the everyday use of speech technology on their mobile devices lies not in the quality of the transcription, but in the latency and the reliability of the network connection used to transmit the audio and receive results. Still, due to the limited computing resources available on these devices and the complexities of recognizing continuous speech, large vocabulary speech recognition is typically performed on a powerful servers in data centers [1, 2].

In this work, we explore the ramifications of shifting some of the recognition processing to the device. We examine two use cases in particular. The first is a hybrid recognizer, along the lines of [3, 4, 5], where a subset of voice search queries are handled on-device and the remainder are passed along to the more powerful server-side engine. The second is an offline-only voice actions recognizer, containing command-and-control style queries which could be executed without the aide of a network connection.

The work described in this paper was performed while the first author was an intern at Google, New York.

We frame both problems in terms of utterance verification (UV) [6], and transcription. In this context, the verification task is to determine whether or not an utterance contains a phrase or sentence that is among the (possibly quite large) set expected to be recognized on-device. The verification process may be performed implicitly during recognition or in a second step, but in both cases the result of the overall recognition process is either a transcript of the utterance or an indication that the utterance has been *rejected* as being out-of-domain.

The problem of identifying extraneous speech has received attention from many areas of the research community. The explicit detection of out-of-vocabulary (OOV) words in large-scale continuous speech recognition is known to improve accuracy [7, 8]. Methods used for this task include the introduction of *garbage* models [9, 8, 10] or the use of word confidence models [11, 9, 12, 13]. Keyword spotting often uses related techniques, at times incorporating the entire recognition lattice to effectively ignore large swaths of non-keyword speech [14, 15, 16]. In dialogue systems, on the other hand, the utterance verification problem is occasionally attacked with multiple domain-specific recognizers evaluated concurrently to classify utterances based on a comparison and scoring of hypotheses from different systems [17, 18, 19].

In this paper, we draw inspiration from the techniques described above to scale the on-device utterance verification and transcription tasks to tens of thousands of phrases. We compare two possible solutions to this problem. We first explore the use of explicit garbage models (e.g., a phone loop) to reject utterances not in our target set. We describe the limitations of the naive implementation of such sub-word models even when higher-order phonotactics are employed. We then propose a fix to this approach that makes use of cross-over arcs from the acceptance subgraph into the garbage model to drastically improve performance. The second method we explore consists of a standard n-gram language model, followed by utterance verification through whitelisting of allowed phrases after recognition has ended. Both techniques are evaluated by sweeping sentence-level posteriors to illustrate the trade-off between accuracy and recall. To perform these tasks we leverage previous work in our lab regarding on-device speech recognition which focused on the development of an accurate, small-footprint, large vocabulary speech recognizer targeted at dictation [20].

The remainder of this paper is organized as follows. In Section 2, we discuss our data-driven methods for the mining of target phrases. Section 3 presents various techniques for on-device utterance verification and transcription. Section 4 discusses our experimental results in two domains: frequent voice search queries and offline-compatible voice actions. Finally, Section 5 concludes this paper with ideas for future work.

Targets	Targeted queries (Q)	Non-targeted queries ($\overline{Q}$)
FREQUENT	820K (70K unique; 28K words)	2.5M (2.3M unique)
OFFLINE	865K (67K unique; 23K words)	32M (20M unique)

Table 1: *Overview of the training sets used in the experiments of Section 4. Each training set was gathered from a separate sample of voice search query logs.*

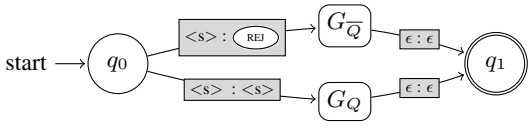

Figure 1: *Grammar with a generic garbage model for detecting out-of-grammar phrases.*

2. Mining Phrases

This work operates on the assumption that the (possibly infinite) set of phrases under consideration can be broken down into a finite targeted set of queries, $\mathbb{Q}$, and the remainder $\overline{\mathbb{Q}}$. The recognition task is then to transcribe $\mathbb{Q}$ accurately, while merely identifying and *rejecting* utterances in which a phrase from $\overline{\mathbb{Q}}$ was spoken. The manner in which rejected utterances are handled is likely to be application-specific. For instance, one might ignore a rejected utterance entirely if a device is offline, or, in the connected scenario, pass these utterances along to a more capable ASR system. The focus of this work is on the resource-constrained recognition of $\mathbb{Q}$.

In this paper, we consider the set of voice search queries received through various mobile devices. We instantiate $\mathbb{Q}$ and $\overline{\mathbb{Q}}$ using two different strategies. Note that we use $\mathbb{Q}$ to denote a set of phrases independent of any data, while its instantiation Q is associated with a set of empirical counts that can be gathered and used when training a model. In the first strategy, we exploit the fact that the most frequently occurring queries cover the largest relative portion of voice search traffic. We therefore take Q_{FREQUENT} to be the top-N distinct search queries and the counts thereof. The goal might then be to deploy an on-device recognizer that targets the head of the voice search query distribution in hopes of robustly handling some portion of the voice traffic.

While most of the top-N voice search queries require an external service (such as a search engine) to execute, some queries can be executed locally on device. A watch, for example, should not need an internet connection to carry out the "Set an alarm" command. Thus, our second strategy explicitly selects those queries, Q_{OFFLINE} that can be processed offline (e.g. command and control).

For our experiments (Section 4) we utilize two training sets as described in Table 1. To construct an estimation of $\overline{Q}_{\text{FREQUENT}}$, we use the tail of the voice search distribution. $\overline{Q}_{\text{OFFLINE}}$, on the other hand, is constructed from the sampled queries that could *not* be carried out offline. Since queries compatible with offline execution are relatively rare, we must sample significantly more data to acquire a similar number of targeted queries. Note, however, that it is only the properties of Q that will come to dominate the size of our models.

3. Rejection Modeling

The resource constraints of on-device recognition have led researchers to explore alternate ways of modeling out-of-domain speech. The authors of [3] and [5] describe on-device recognizers supplemented with confusable words or phrases to act as decoys, causing some utterances to be rejected. In the case of [3], rejected phrases are passed along to a second, more powerful network recognizer. With similar goals in mind, our work opts for somewhat more scalable approaches that can make use of training sets containing millions of utterances.

3.1. NAIVE Rejection Model

We first consider a naive reapplication of a garbage model commonly used to handle out-of-vocabulary words [8, 9, 10]. One simple approach to adapting this model to the utterance verification task is to create a graph with two paths out of the start state, where one path goes through an acceptance grammar and the other (rejection) path goes through a phonotactic garbage model. A special token is added to the output of the rejection paths to indicate that an utterance is out of domain. A schematic representation of the FST is depicted in Figure 1.

In this work, G_Q, is constructed from phrases in Q and weighted according to their frequencies. Subsequently, a phonotactic garbage model grammar, $G_{\overline{Q}}$, is constructed from $\overline{Q}$ as follows. We map every $z \in \overline{Q}$ to the set of its possible phoneme representations using the pronunciation lexicon L. We proceed by using these phonetic transcripts and their counts as a corpus for constructing the n-gram phonotactic garbage submodel $G_{\overline{Q}}$. Note that if we choose $n = 1$ for construction of $G_{\overline{Q}}$, our garbage sub-model becomes a phone loop that incorporates phoneme frequency statistics. In order to support the use of phoneme tokens in grammar G we extend the pronunciation lexicon L with an identity lookup rule for each phone. G_Q and $G_{\overline{Q}}$ are constructed separately and afterwards merged using the FST union operation [21]. Construction of the CLG transducer [22] then follows from the standard composition of the context-dependency transducer C, the augmented pronunciation lexicon L and the specially-crafted grammar G as described above.

3.2. PREFIX-based Rejection Model

We now describe an extension of the baseline garbage model detailed above that integrates the acceptance and rejection subgraphs in a way that incorporates statistical information regarding extraneous speech.

Given a training set consisting of the set of target phrases Q, its complement $\overline{Q}$, and their counts within the corpus from which they were sampled, we construct G in two parts. A deterministic grammar FST, G_Q, is constructed just as in the NAIVE case. Subsequently, the transitions into the phonotactic garbage model, $G_{\overline{Q}}$, are constructed from $\overline{Q}$ as follows. For every utterance $z \in \overline{Q}$, we find its longest common prefix with the target set Q, $lcp(z, Q) = w_1^{(z)} \cdots w_{k_z}^{(z)}$. This identifies the location of the transition from G_Q into the rejection model, $G_{\overline{Q}}$. In particular, we traverse FST G_Q following the path of consecutive transitions $\pi = t_1 \cdots t_{k_z}$ whose labels match $w_1^{(z)} \cdots w_{k_z}^{(z)}$. Letting $n[t]$ denote the destination state of transition t in the FST, we add a transition from the end of our longest common prefix, state $n[t_{k_z}]$, to the initial state of our phonotactic garbage sub-model, $G_{\overline{Q}}$. The input label of this transition is ϵ and the output label is a rejection token, indicating that the path is out-of-domain. The cross-over transition is weighted proportional to the count of the prefix in $\overline{Q}$; taking into account the already established probability mass of the transitions leaving that state.

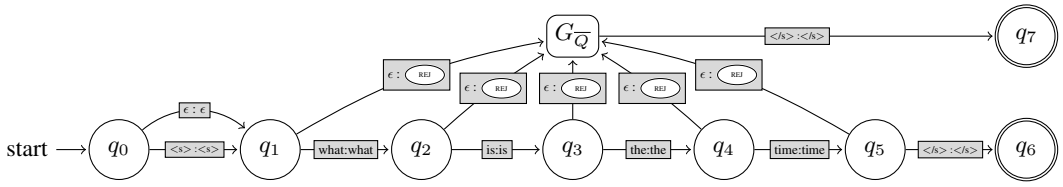

Figure 2: *Finite-State Transducer of grammar G for the example that accepts only a single query* "what is the time?"*. Transition weights are omitted for brevity.*

Actual utterance	Recognizer hypothesis
what is the time?	<s> what is the time </s>
<hello>!	<s> REJ
what <can you tell me>?	<s> what REJ
what is <this>?	<s> what is REJ
what is the time <in Madrid>?	<s> what is the time REJ

Table 2: *Example scenarios for different utterances and how these are interpreted by the recognizer.*

Finally, we use the remainder suffix $s_z = w_{k_z+1}^{(z)} \cdots w_{|z|}^{(z)}$ of z, and the frequency statistics of z, as training data for the construction of the phoneme-based n-gram garbage model in a similar fashion as described above (Section 3.1).

It may be the case that Q and $\overline{Q}$ do not have a high degree of overlap in terms of their common prefixes, causing sparsity in the cross-over transitions learned in this fashion. This motivates us to use a type of additive smoothing. At every in-grammar state in G_Q we assume an unobserved count α crossing-over to the rejection model, as described above.

The ideal behavior of this model is illustrated in the following toy example. Suppose we are given a large corpus C split into the following training phrases: a single target sentence, $Q = \{$<s> what is the time </s> $\}$ and all the remaining phrases $\overline{Q} = C \setminus Q$. Our goal is to construct a system that correctly recognizes the query "*what is the time?*" and rejects anything else (see Figure 2). Table 2 depicts some scenarios for different utterances and how the recognizer might see them. The first example generates a valid hypothesis, while the remaining ones contain the rejection token. Note that in the case of the last example, its prefix *what is the time* overlaps with a targeted query.

As recognition proceeds, the decoder keeps track of candidate paths through the CLG transducer. Unlike in the NAIVE model, where candidate paths are forced to remain in either the acceptance or rejection subgraphs for the duration of the decoding process, the PREFIX approach allows a path to cross over into the garbage model when an unexpected phoneme sequence occurs.

3.3. N-Gram WHITELIST Rejection

We can also adapt a regular n-gram model to the rejection task. If we employ an n-gram model without rejection capabilities, then the recognizer will generate a hypothesis even for non-target phrases. We can adjust this behavior by maintaining a white-list of target phrases. If the top hypothesis is a target phrase, the transcript is passed along, otherwise the utterance is rejected.

More precisely, we construct an n-gram model over the phrases in Q and reduce its size by relative entropy pruning [23]. Once recognition finished we apply white-list filtering. This approach is somewhat similar to evaluating the n-gram model and grammar G_Q separately and rejecting the utterance if their hypothesis differ [17, 18, 19].

3.4. Tuning & Posteriors

All of the methods described above require some degree of tuning. In the case of the NAIVE and PREFIX methods, we found it helpful to add a cost-multiplier β to the acceptance subgraph. More precisely, every negative log-probability on the arcs of the acceptance graph can be multiplied by β to decrease the relative weight of acceptance over rejection. The n-gram approach, on the other hand, requires decisions regarding the characteristics of the training set and the pruning of the resulting model.

In all of the models described, an N-best list can be generated with a score for each hypothesis. Normalizing these scores yields an approximate sentence-level posterior which can be used as a proxy for confidence. A confidence threshold can then be swept and WER can be plotted as a function of recall to demonstrate the tradeoff between the accuracy and the coverage of a model.

4. Experimental results

We evaluate the approaches described in the previous section on two tasks. First, we use these techniques to model the head of the voice search query distribution. For this experiment, we train our models with data from Q_{FREQUENT} and $\overline{Q}_{\text{FREQUENT}}$ and test on a 22K sample of voice search queries made through phones. Second, we examine a scenario in which voice actions from a smartwatch are recognized without the benefit of a network connection. For this experiment, we train our models with data from Q_{OFFLINE} and $\overline{Q}_{\text{OFFLINE}}$ and evaluate on a WATCH test set containing a 12K utterance sample of voice search queries made through smartwatches. For all experiments we use the 2.7 million parameter DNN acoustic model described in [20] and a consistent set of decoding parameters known to allow for on-device recognition in real-time.

Figure 3 plots word error rate on the PHONE test set against recall of the frequent voice search queries in FREQUENT. We evaluated the NAIVE approach using different phonotactic garbage model complexities ranging from unigrams to trigrams induced by the phoneme transcripts of $\overline{Q}_{\text{FREQUENT}}$. Interestingly, while the PREFIX technique performed significantly better overall, using the same technique of applying higher-order garbage models did not yield gains; we therefore present only the unigram garbage model with phone frequencies trained from suffixes of $\overline{Q}_{\text{FREQUENT}}$. Finally, the 4-gram was trained on Q_{FREQUENT} and pruned for the WHITELIST approach.

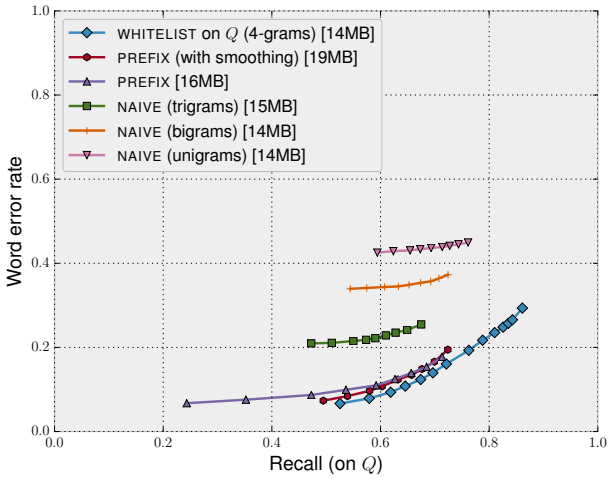

Figure 3: *Trained on* FREQUENT *evaluated on the* PHONE *test set.*

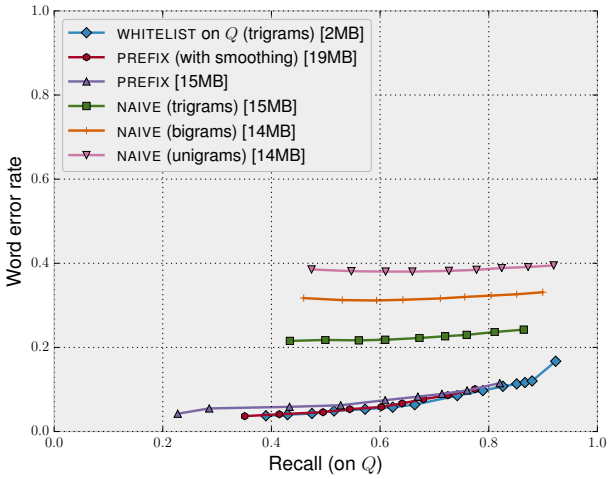

Figure 4: *Trained on* OFFLINE *evaluated on the* WATCH *test set.*

Table 3: *Performance metrics for hybrid system (*FREQUENT*) on* PHONE *and* WATCH *test sets. The first three rows show statistics for the resource-constrained* PREFIX *model, the utterances handed-off to the* NETWORK *recognizer and the combined recognizer respectively. The last row depicts performance in a non-hybrid setting.*

Recognizer	PHONE			WATCH		
	%	WER	SACC	%	WER	SACC
PREFIX	14.6%	9.7	88.36%	16.7%	11.1	82.37%
NETWORK	85.4%	11.2	70.89%	83.3%	21.2	53.82%
Combined	100%	11.1	73.41%	100%	20.1	58.56%
NETWORK	100%	10.9	73.76%	100%	20.1	58.31%

Figure 4 shows the analogous curves for models trained on OFFLINE data and applied to the WATCH test set. As in Figure 3 the sweeps are generated by varying a confidence threshold. For the NAIVE and PREFIX models, β is fixed at a value of 1.46 determined by running a preliminary sweep on a dev set. Similar trends occur here. The WHITELIST and PREFIX-based approaches perform similarly well, while the naive approach falls short. In this case, however, a trigram was used for the n-gram approach, resulting in a significantly smaller model.

We now turn our attention to the hybrid recognition scenario, where rejected utterances are sent to a server-side recognizer. Such a setup might be particularly beneficial in slow or spotty connections, where recognizing some utterances on-device would significantly improve user experience. The hope is that the server-side will still be able to transcribe $\overline{Q}_{\text{FREQUENT}}$ and $\overline{Q}_{\text{OFFLINE}}$, albeit at higher latency. Table 3 depicts the results of a simulated evaluation of hybrid systems for both of our domains. Impressively, the impoverished on-device models don't seem to be detrimental to the overall accuracy of a hybrid system. Indeed, in each case it seems that we can recognize around 15% of the queries while barely effecting overall WER.

5. Conclusions

In this paper, we described multiple techniques for resource-constrained, on-device utterance verification. An unsupervised data-driven strategy was employed for selecting targeted phrases from recognition logs. We compared a sub-word garbage modeling approach and introduced a novel method for modeling extraneous speech based on common prefixes between the targeted and non-targeted phrases. We also explored the use of a white-listed n-gram for the same task.

Evaluation was performed using transcribed anonymized voice search traffic originating from smartphones and smartwatches. On both sets we observe low word error rate at varying amounts of traffic. We showed that, even with significantly smaller on-device models, it's possible to build a hybrid system that robustly handles some amount of traffic without the latency or reliability concerns that accompany the use of a network connection.

There are a number of clear paths to extending this work. First, the addition of support for class-based language models would expand the applicability of these techniques to domains with generic concepts (e.g. time or numbers). Second, these basic techniques could be extended with more sophisticated smoothing techniques and confidence models to maximize the recall at a particular WER. Finally, one interesting application we have considered exploring is the on-device expansion of the target set, perhaps leading to the implementation of a *cache* recognizer which handles a particular user's common queries on device.

6. Acknowledgements

The authors would like to thank David Rybach, for providing the useful insights about the decoder and CLG composition, Michael Riley and Cyril Allauzen, for helpful discussions and their work on OpenFst, and Rohit Prabhavalkar for conversations regarding the whitelisted N-best approach.

During writing of this article, the first author was supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 312827 (VOX-Pol).

7. References

[1] D. Zaykovskiy, "Survey of the speech recognition techniques for mobile devices," in *SPECOM*, 2006.

[2] J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, and B. Strope, "Google search by voice: A case study," in *Advances in Speech Recognition: Mobile Environments, Call Centers and Clinics*. Springer, 2010, ch. 4, pp. 61–90.

[3] M. Levit, S. Chang, and B. Buntschuh, "Garbage modeling with decoys for a sequential recognition scenario," in *ASRU*. IEEE, 2009, pp. 468–473.

[4] T. Kuhn, A. Jameel, M. Stumpfle, and A. Haddadi, "Hybrid in-car speech recognition for mobile multimedia applications," in *VTC*, vol. 3. IEEE, 1999, pp. 2009–2013.

[5] M. Alessandrini, G. Biagetti, A. Curzi, and C. Turchetti, "A garbage model generation technique for embedded speech recognisers," in *SPA*. IEEE, 2013, pp. 318–322.

[6] H. Jiang, F. K. Soong, and C.-H. Lee, "A data selection strategy for utterance verification in continuous speech recognition," in *Interspeech*, 2001, pp. 2573–2576.

[7] T. Kawahara, C.-H. Lee, and B.-H. Juang, "Key-phrase detection and verification for flexible speech understanding," in *ICSLP*, vol. 2. IEEE, 1996, pp. 861–864.

[8] I. Bazzi, "Modelling out-of-vocabulary words for robust speech recognition," Ph.D. dissertation, Massachusetts Institute of Technology, 2002.

[9] T. J. Hazen and I. Bazzi, "A comparison and combination of methods for oov word detection and word confidence scoring," in *ICASSP*, vol. 1. IEEE, 2001, pp. 397–400.

[10] C. Parada, M. Dredze, D. Filimonov, and F. Jelinek, "Contextual information improves oov detection in speech," in *HLT '10*. ACL, 2010, pp. 216–224.

[11] J. Caminero, D. De La Torre, L. Villarrubia, C. Martin, and L. Hernandez, "On-line garbage modeling with discriminant analysis for utterance verification," in *ICSLP*, vol. 4. IEEE, 1996, pp. 2111–2114.

[12] M.-W. Koo, C.-H. Lee, and B.-H. Juang, "Speech recognition and utterance verification based on a generalized confidence score," *Speech and Audio Processing*, vol. 9, pp. 821–832, 2001.

[13] M. Creutz, T. Hirsimäki, M. Kurimo, A. Puurula, J. Pylkkönen, V. Siivola, M. Varjokallio, E. Arisoy, M. Saraçlar, and A. Stolcke, "Morph-based speech recognition and modeling of out-of-vocabulary words across languages," *Transactions on Speech and Language Processing*, vol. 5, no. 1, dec 2007.

[14] J. G. Wilpon, L. R. Rabiner, C. H. Lee, and E. R. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden markov models," *Acoustics, Speech, and Signal Processing*, vol. 38, no. 11, pp. 1870–1878, 1990.

[15] J. Rohlicek, P. Jeanrenaud, K. Ng, H. Gish, B. Musicus, and M. Siu, "Phonetic training and language modeling for word spotting," in *ICASSP*, vol. 2. IEEE, 1993, pp. 459–462.

[16] I. Szöke, P. Schwarz, P. Matjka, and M. Karafit, "Comparison of keyword spotting approaches for informal continuous speech," in *Eurospeech*, 2005.

[17] B.-s. Lin, H.-m. Wang, and L.-s. Lee, "A distributed architecture for cooperative spoken dialogue agents with coherent dialogue state and history," in *ASRU*, 1999.

[18] S. Ikeda, K. Komatani, T. Ogata, and H. G. Okuno, "Extensibility verification of robust domain selection against out-of-grammar utterances in multi-domain spoken dialogue system," in *Interspeech*, 2008.

[19] M. Takaoka, H. Nishizaki, and Y. Sekiguchi, "Utterance verification using garbage words for a hospital appointment system with speech interface," in *ASRU*, 2011, pp. 336–341.

[20] X. Lei, A. Senior, A. Gruenstein, and J. Sorensen, "Accurate and compact large vocabulary speech recognition on mobile devices," in *Interspeech*. ISCA, 2013, pp. 662–665.

[21] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "Openfst: A general and efficient weighted finite-state transducer library," in *Implementation and Application of Automata*. Springer, 2007, pp. 11–23.

[22] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.

[23] A. Stolcke, "Entropy-based pruning of backoff language models," in *DARPA Broadcast News Transcription and Understanding Workshop*, 2000, pp. 8–11.