ACCURATE CLIENT-SERVER BASED SPEECH RECOGNITION KEEPING PERSONAL DATA ON THE CLIENT

Munir Georges[1,2], Stephan Kanthak[1], Dietrich Klakow[2]

[1]Nuance Communications, Automotive Speech R&D, Aachen, Germany
[2]Spoken Language Systems, Saarland University, Saarbrücken, Germany
{Munir.Georges, Dietrich.Klakow}@LSV.Uni-Saarland.De, Stephan.Kanthak@Nuance.Com

ABSTRACT

In this paper, a novel technique is proposed that recognizes speech on a server but all private knowledge is processed on the client. Private knowledge could be address book entries, calendar entries or medical patient data.

The technique combines the advantage of a powerful server with almost unlimited memory and the advantage using locally available user dependent knowledge. A dynamic language model is used to recognize speech with the help of content dependent acoustic fillers on a server. The result is then recognized including user dependent knowledge on a client, e.g., a smart phone. We achieved a word error rate reduction of 17% on the Wall Street Journal Corpus.

Index Terms— Dynamic Language Model, Acoustic Filler, Client-Server Speech Recognition, Data Privacy.

1. INTRODUCTION

It would be beneficial for various speech applications to use local data such as address book entries, calendar entries or other private data. These private data are often not available on a server. This may be because of legal reasons, e.g., for medical patient data. Smaragdis et al. proposed a framework for secure speech recognition [1]. Using local data can also reduce the required server storage capacity and its software complexity for high demand speech applications. Recognition on an embedded device is often limited due to restricted computational power and memory.

We propose a novel technique that combines client and server based speech recognition through dynamic language models and acoustic fillers. There is no need to synchronize user dependent private data to achieve accurate speech recognition. All private data is recognized on the client. It enhances the recognition hypotheses from the server with suitable locally available data. This allows the use of models that are highly optimized for the use on embedded devices on the one hand. On the other hand, the server recognizer can use precise acoustic models and language models estimated on the crowd. Our novel technique can take advantage of private data that is only locally available on the client.

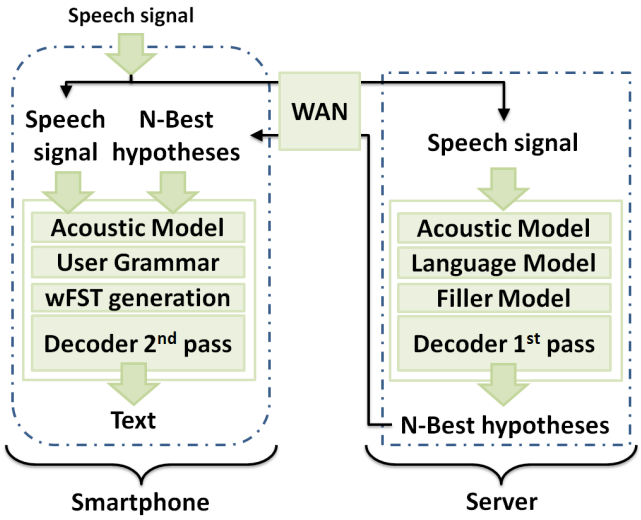

Fig. 1. Speech recognition hypotheses from a server are enhanced on the smart phone with user knowledge. The recognizers are connected through a Wide Area Network (WAN).

Our novel speech recognition technique uses several language models, simultaneously. Murveit et al. described a technique that uses different levels of detail between recognition passes [2]. Multiple pass search strategies were described in detail by Schwartz et al [3]. We combine statistical language models and grammars. A combining of linguistic and statistical knowledge was proposed by Moore et al [4]. Bruganara et al [5] proposed hierarchical language models. Linking several local models together according to a general one was proposed by Nasr et al [6]. The idea was further pursued using weighted transducers, e.g., by Schalkwyk et al [7] or Mohri [8]. An on–the–fly transducer nesting was proposed by Georges et al [9]. The technique described in this paper combines different language models in multiple passes on different devices. The used dynamic language model is described in Section 2. Our technique is not related to distributed language models for estimating N-grams on a grid computer using a wide Storage–Area–Network as described by Mnih et al [10] and Brants et al [11].

Figure 1 gives an overview of the novel technique. The speech signal is captured, without loss of generality, on a smart phone and passed through a wide area network to a server. A generalized language model is used for recognition along with an acoustic model and acoustic fillers. This is described in detail in Section 3. The recognition hypotheses are passed to the smart phone. The hypotheses were enhanced on the smart phone with user grammars and assembled to a weighted finite state transducer and finally recognized as described in Section 4. We evaluated the technique on the Wall Street Journal Corpus [12] which is summarized in Section 5. A word error reduction of 17% has shown that a significant accuracy improvement can be achieved. We observed a delay of 15% compared to real time speech recognition. This delay can be used to provide preliminary recognition results. The client is typically equipped with embedded processors and advances battery-saving modes. Our novel technique can take advance of these modes because not the full computational power is required over the entire processing.

2. DYNAMIC LANGUAGE MODEL

There are various language models used in our proposed technique which were dynamically combined on multiple devices. An overview of the language models is given in Figure 2.

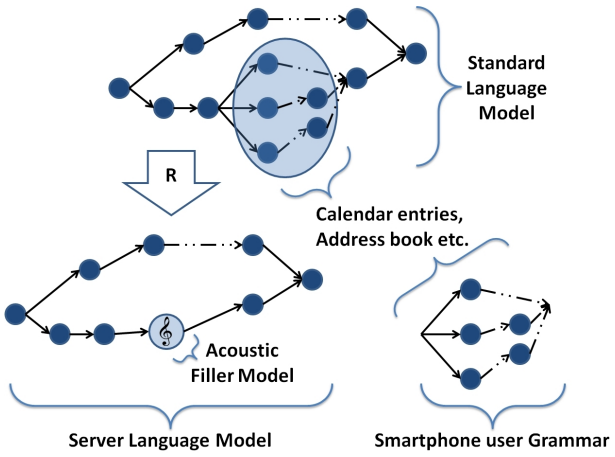

Fig. 2. The standard language model is divided into an user dependent grammar and a generalized language model. Both were represented as weighted finite state transducers.

The starting point is a corpus $\mathbb{C}$ for language model training. We also define K sets of user-dependent word sequences $S_k \subset W^*$ with $k \leq K$ over vocabulary W. User dependent word sequences could be terms from a calendar, proper names from an address book or credit card numbers, etc. Let S_K be the set of all user dependent word sequences:

$$S_K = \bigcup_k^K S_k \subseteq W^*.$$

Each occurrence of a term from S_k in $\mathbb{C}$ is substituted with a marker $t_k \in T$ with $|T| = K$ and $T \cap W = \emptyset$. The result is a generalized corpus $\mathbb{C}'$. We used the transducer replacement operator R proposed by [13], which is described in [9] for dynamic language model estimation. Regular expressions can be used, too. The definition of R is given as:

$$R : W^* \rightarrow ((W \cup T)^* \backslash S_K)^*.$$

A generalized N-gram Markov language model is then estimated on $\mathbb{C}'$. The probability of a word sequence $\underline{w}$ is given by the sequence computed by $R(\underline{w})$. The generalized language model can be formulated as:

$$P(R(\underline{w})) = \prod_i^{|R(\underline{w})|} P(R(\underline{w})_i | R(\underline{w})_{i-N+1:i-1}) \cdot \begin{cases} P_k(\underline{w}_{m:n}) & \exists m,n : R(\underline{w}_{m:n}) = R(\underline{w})_i = t_k \\ 1 & else. \end{cases}$$

P_k is a conditional probability for a replaced word sequence by R where the probability is 1 if no word was replaced. The model is normalized if each word sequence in S_K is uniquely associated with one marker. Let $P(\underline{x}|\underline{w})$ be the acoustic model. The most probable word sequence $\underline{\hat{w}}$ is given by a sequence of speech features $\underline{x}$ [14], [15]. Here, the fundamental formula of speech recognition becomes:

$$\underline{\hat{w}} = \arg\max_{\underline{w} \in W^*} P(\underline{x}|\underline{w}) P(R(\underline{w})).$$

The server uses the generalized language model where each P_k is replaced on–the–fly with a corresponding acoustic filler as described in Section 3. The acoustic fillers are based on phoneme loop models estimated on the replaced word sequences. There are various filler alternatives described in the literature. Asadi et al. proposed fillers that where used to obtain phonetic transcriptions for modelling out of vocabulary words [16]. Jiang et al [17] described fillers based on sub–word features for a vocabulary–independent word confidence measure. Fillers based on word fragments were proposed by Klakow et al [18] and various models were described by Bazzi et al [19]. We analysed the scope for improvement with oracle fillers as described in Section 5.

The server recognition result is used along with user grammars on the client, e.g., a smart phone to assemble an user dependent transducer. This transducer is recognized on the client as described in Section 4.

3. RECOGNITION ON THE SERVER

The speech recognizer on the server uses a generalized language model where user dependent word sequences were recognized with acoustic fillers. We use weighted finite state transducers, so that an on–the–fly nesting technique could be used to embed the acoustical filler models.

The generalized N-gram Markov language model can be represented by a weighted automaton G_1. The relations between phoneme sequences $\mathbb{P}$ and words W is described by a lexicon transducer $L \subseteq (\mathbb{P} \times W)^*$. Further on, the context dependency between phonemes is given by the transducer C. A static search network can be assembled as follows:

$$M_1' = min(det(C \circ L \circ G_1)),$$

where $\circ$ denote the composition operator [20]. min, det denote the transducer operator for minimization and determinization [21]. Finally, M_1' is composed on–the–fly with a hidden Markov model H along with the cross-word computation. This was initially described by Hori et al [22], [23] and further improved by McDonough et al [24] and Allaucen et al [25], [26]. The probability for a phoneme sequence given a sequence of speech features can be computed [27] using a token passing time synchronous Viterbi beam search [28]. Each acoustic filler is represented as a weighted transducer, sharing the same set of hidden states. There is no need to extend the acoustic model. The transducer replacing operator [27] can be used to nest the filler model into the M_1' transducer. We nest the filler on–the–fly [9] each time when a marker from the generalized language model was reached.

The recognition result is an N-best list of sentence hypotheses [3]. The acoustic filler location is tagged and will be later used to include user dependent knowledge. Using N-best sentence hypotheses ensures backwards compatibility for other speech applications using the same server infrastructure. Alternatively, a lattice could be passed to the client.

4. RECOGNITION ON THE CLIENT

The smart phone receives recognition hypotheses from the server. The user dependent language portion is marked. This could be proper names, dates or other private data. The user data is locally available as grammar, e.g., an address book, calendar or medical recordings. These data are used for speech recognition on the client, e.g., a smart phone.

In this paper, the received N-best sentence hypotheses from the server were summarized in one grammar G_2, where each sentence ends up in one grammar rule. This is comparable to an output voting error reduction system [29] where different hypotheses from various recognizers were combined to improve the overall accuracy. Schwenk et al [30] proposed to include language model weights which may also be used for the proposed technique in this paper. Alternatively, a word lattice could be delivered by the server. Each marker in G_2 points to an user grammar. Similar to the recognition on the server, a transducer M_2' can be assembled as:

$$M_2' = min(det(C \circ L \circ G_2)).$$

The phoneme dependency model is C and L is the lexicon transducer. M_2' is composed on–the–fly with a hidden Markov model H along with the cross-word computation. A token passing time synchronous Viterbi beam search is used similar to the server recognition system. In addition, histogram pruning [14] is applied to fulfil the embedded memory requirement.

5. EVALUATION

The proposed system is evaluated using the Wall Street Journal Corpus [12]. We use the same decoder set-up on the server and on the client for comparable reasons. An integer value based acoustic model evaluation was used. We did not take advantage of any acoustical adaptation techniques such as MLLR etc. The SRI Language Model tool–kit [31] was used to estimate the 5k word language models with Kneser-Ney discounting [32]. We prepared the Wall Street Journal Corpus $\mathbb{C}$ for language model training in a way that it becomes comparable to real world applications.

Table 1. *Used grammar for evaluation in Backus–Naur Form*

⟨*DAY*⟩	:=	'Monday' | 'Tuesday' | ...
⟨*MONTH*⟩	:=	'January' | 'February' | ...
⟨*NUM*⟩	::=	⟨*num*⟩ ['.' ⟨*num*⟩]['**,**' ⟨*num*⟩]
⟨*num*⟩	::=	'one' ⟨*num*⟩ | 'two' ⟨*num*⟩ | ... | ⟨*empty*⟩
⟨*MONETARY*⟩	::=	⟨*NUM*⟩'dollar'|⟨*NUM*⟩'cent'|...
⟨*PERCENT*⟩	::=	⟨*NUM*⟩ 'percent'
⟨*ABBREVIATION*⟩	::=	⟨*Letter*⟩ | 'SVOX' | ...
⟨*Letter*⟩	::=	'A.' ⟨*Letter*⟩ | ... | ⟨*empty*⟩

Imagine a short message dictation application where the local available address book, the music title collection and the calendar should be included in the recognition. Here, the user dependent knowledge S is a subset of all weekdays, names of months, various number terms and abbreviations according to Table 1. We exclude all user dependent knowledge $S' \subset S$ from the corpus that occur in the set of test sentences to reduce the coverage of S' from 31% down to 7%:

$$\mathbb{C}' = \{\underline{w} \in \mathbb{C} | \nexists m, n : w_{m:n} \in S'\}.$$

This coverage seems realistic for real world applications when we analyse N-gram cut-off data. $\mathbb{C}'$ is used to estimate the language models for the server only system. Each term of S is replaced in the corpus $\mathbb{C}'$ with a corresponding marker symbol from T. We used the grammar in Table 1 and the transducer replacement operator R to build $\mathbb{C}''$ as follows:

$$\mathbb{C}'' = \{\underline{w} \in ((W \cup T)^* \backslash S)^* | \exists \underline{w}' \in \mathbb{C}' : \underline{w} = R(\underline{w}')\}.$$

The generalized corpus $\mathbb{C}''$ is used to estimate the dynamic language model. This language model is used by the server along with the acoustic fillers. Those fillers are based on phoneme loop models. Every substituted word sequence from $\mathbb{C}''$ is used to estimate the 1-gram phoneme loop filler.

Initially, we analyzed the impact of the number of N-best hypotheses which were passed from the server (1^{st} decoder) to the client (2^{nd} decoder). We expected that this influences the recognition accuracy significantly. This has been confirmed by the experiments. Figure 3 shows the influence using 3-gram dynamic language models. We examined

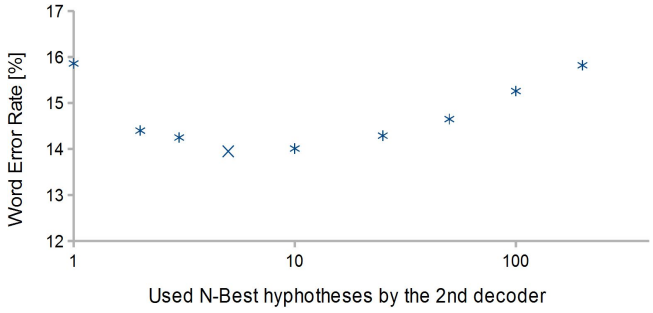

Fig. 3. The minimal word error rate was achieved when the 2^{nd} decoder was recognizing on 5-best hypotheses.

this behaviour also for the 2-gram and 4-gram models where we observed similar behaviour. A minimal word error rate along with tenable recognition time for the 2^{nd} decoder was achieved using 5-best hypotheses. This experiment is denoted with an "x" in Figure 3. The pruning behavior on the 2^{nd} decoder has a significant influence on the recognition time but nearly no influence on the accuracy for small N.

In this paper, phoneme loop fillers were used. Even when each model was estimated on representative data, the difference between each filler is minimal. The accuracy can be further improved using fillers which are strongly user adapted. We used oracle full word fillers estimated on the test data in the following experiment. Figure 4 illustrates the potential of improvement for dynamic 2-gram language models on representative hardware. The oracle filler outperforms the proposed phoneme loop model as expected. Further, we compared the performance with a grammar 2-gram language model where user dependent grammars were nested during decoding. This is only possible when the user data is available on the server. Our novel technique could achieve nearly the same recognition accuracy with user dependent fillers although it took a certain delay. We observed similar behaviour using the 3 and 4-gram language model set-up.

Finally, we compared our novel system with server only speech recognition. A faster recognition was achieved with 4-gram language models whereas no further accuracy improvement was observable. In summary, the recognition accuracy of the novel technique outperforms the server only system as summarized in Figure 5. No private data has to be synchro-

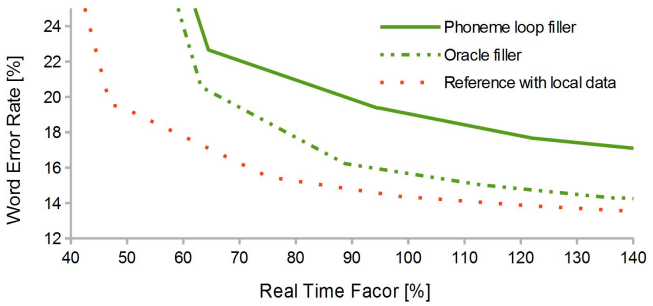

Fig. 4. The oracle filler gives an impression of the potential of improvement when user adapted fillers can be used compared to the proposed phoneme loop fillers.

nized with the server. All private data such as the address book, calendar or medical data remains on the client. The latency of the proposed technique requires a user feedback mechanism for certain applications. Here, the latency was on average 15% of the real time. The processor can stay in a battery-saving mode for the most time. A word error rate reduction of 17% was achieved for the 3-gram dynamic language model set-up.

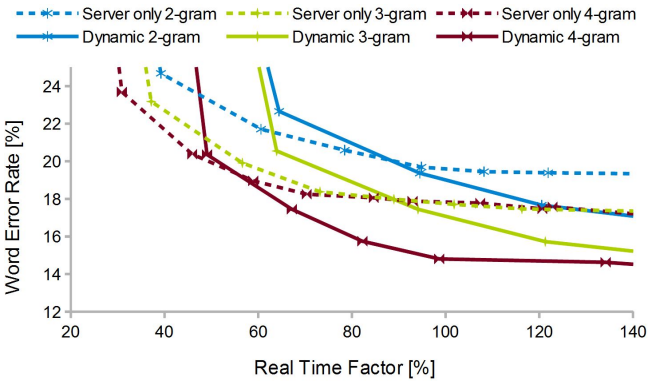

Fig. 5. The novel client-server speech recognition technique is beneficial if a short delay is acceptable.

6. SUMMARY

We propose a technique for speech recognition on a client and server where no private data has to be available on the server. Private date could be an address book, a private calendar or some medical patient data. A dynamic language model is used on the server along with acoustic fillers. The recognition result is then combined with user dependent knowledge on the client, e.g., on a smart phone.

We have shown that the proposed technique can improve speech recognition on the Wall Street Journal corpus. An average latency of 15% was observed compared to real time recognition and, in the same time, a word error rate reduction of 17% was achieved.

7. REFERENCES

[1] P. Smaragdis and M. V. S. Shashanka, "A framework for secure speech recognition." *IEEE Transactions on Audio, Speech & Language Processing*, vol. 15, no. 4, pp. 1404–1413, 2007.

[2] H. Murveit, J. Butzberger, V. Digalakis, and M. Weintraub, "Large-vocabulary dictation using SRI's DECIPHER speech recognition system: Progressive search techniques," in *Proceedings of ICASSP*, 1993.

[3] R. Schwartz, L. Nguyen, and J. Makhoul, "Multiple-pass search strategies," in *Automatic Speech and Speaker Recognition*, ser. The Kluwer International Series in Engineering and Computer Science, C.-H. Lee, F. Soong, and K. Paliwal, Eds. Springer US, 1996, vol. 355, pp. 429–456.

[4] R. Moore, D. Appelt, J. Dowding, M. Gawron, and D. Moran, "Combining linguistic and statistical knowledge sources in natural language processing for atis," in *ARPA Spoken Language Technology Workshop*, 1995.

[5] F. Brugnara and M. Federico, "Dynamic language models for interactive speech applications." in *EUROSPEECH*, G. Kokkinakis, N. Fakotakis, and E. Dermatas, Eds. ISCA, 1997.

[6] A. Nasr, Y. Esteve, F. Bechet, T. Spriet, and R. D. Mori, "A language model combining n-grams and stochastic finite automata," in *Proceedings of Eurospeech*, 1999, pp. 2175–2178.

[7] J. Schalkwyk, I. L. Hetherington, and E. Story, "Speech recognition with dynamic grammars using finite-state transducers," in *Processing of INTERSPEECH*, 2003.

[8] M. Mohri, "Local grammar algorithms," in *Inquiries into Words, Constraints, and Contexts.*, A. Arppe, L. Carlson, K. Lindèn, J. Piitulainen, M. Suominen, M. Vainio, H. Westerlund, and A. Yli-Jyrä, Eds. CSLI Publications, 2005.

[9] M. Georges, S. Kanthak, and D. Klakow, "Transducer-based speech recognition with dynamic language models," in *Proceedings of INTERSPEECH*, 2013, pp. 642–646.

[10] A. Mnih and G. E. Hinton, "A scalable hierarchical distributed language model." in *NIPS*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2008, pp. 1081–1088.

[11] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean, "Large language models in machine translation." in *EMNLP-CoNLL*. ACL, 2007, pp. 858–867.

[12] D. B. Paul and J. M. Baker, "The design for the wall street journal-based csr corpus." in *ICSLP*. ISCA, 1992.

[13] L. Karttunen, "The replace operator," 1994.

[14] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall PTR, May 2001.

[15] E. G. Schukat-Talamazzini, *Automatische Spracherkennung - Grundlagen, statistische Modelle und effiziente Algorithmen*, ser. Künstliche Intelligenz. Vieweg, 1995.

[16] A. Asadi, R. Schwartz, and J. Makhoul, "Automatic modeling for adding new words to a large-vocabulary continuous speech recognition system," in *Processing of ICASSP*, 1991.

[17] L. Jiang and X. Huang, "Vocabulary-independent word confidence measure using subword features." in *ICSLP*, 1998.

[18] D. Klakow, G. Rose, and X. L. Aubert, "Oov-detection in large vocabulary system using automatically defined word-fragments as fillers." in *EUROSPEECH*. ISCA, 1999.

[19] I. Bazzi, J. Glass, and A. C. Smith, "Modeling out–of–vocabulary words for robust speech recognition," 2000.

[20] F. C. N. Pereira and M. D. Riley, "Speech recognition by composition of weighted finite automata," in *Finite-State Language Processing*. MIT Press, 1996, pp. 431–453.

[21] M. Mohri, *Weighted Finite State Transducer Algorithms: An Overview*. Physica-Verlag, 2004.

[22] T. Hori, C. Hori, and Y. Minami, "Fast on-the-fly composition for weighted finite-state transducers in 1.8 million-word vocabulary continuous speech recognition," in *INTERSPEECH*. ISCA, 2004, pp. 289–292.

[23] T. Hori and A. Nakamura, "Generalized fast on-the-fly composition algorithm for wfst-based speech recognition." in *INTERSPEECH*. ISCA, 2005, pp. 557–560.

[24] J. W. McDonough, E. Stoimenov, and D. Klakow, "An algorithm for fast composition of weighted finite-state transducers." in *ASRU*, 2007.

[25] C. Allauzen and M. Mohri, "3-way composition of weighted finite-state transducers," in *Implementation and Applications of Automata, 13th International Conference, CIAA 2008, San Francisco, California, USA, July 21-24, 2008. Proceedings*, ser. Lecture Notes in Computer Science, O. H. Ibarra and B. Ravikumar, Eds., vol. 5148. Springer, 2008, pp. 262–273.

[26] C. Allauzen and M. Mohri, "N-way composition of weighted finite-state transducers," *Int. J. Found. Comput. Sci.*, vol. 20, no. 4, pp. 613–627, 2009.

[27] 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.

[28] S. J. Young, N. H. Russell, and Thornton, "Token Passing: A Simple Conceptual Model for Connected Speech Recognition Systems," Cambridge University Engineering Department, Tech. Rep., 1989. [Online]. Available: citeseer.ist.psu.edu/young89token.html

[29] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Proceedings ASRU*, 1997.

[30] H. Schwenk and J.-L. Gauvain, "Combining multiple speech recognizers using voting and language model information." in *INTERSPEECH*. ISCA, 2000, pp. 915–918.

[31] A. Stolcke, "Srilm – an extensible language modeling toolkit," Jun. 06 2002.

[32] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Processing of ICASSP*, 1995.