DYNAMIC VOCABULARY PREDICTION FOR ISOLATED-WORD DICTATION ON EMBEDDED DEVICES

Jussi Leppänen, *Jilei Tian*

Interaction Core Technology Center, Nokia Research Center, Tampere, Finland

{jussi.ar.leppanen, jilei.tian}@nokia.com

ABSTRACT

Large-vocabulary speech recognition systems have mainly been developed for fast processors and large amounts of memory that are available on desktop computers and network servers. Much progress has been made towards running these systems on portable devices. Challenges still exist, however, when developing highly efficient algorithms for real-time speech recognition on resource-limited embedded platforms. In this paper, a dynamic vocabulary prediction approach is proposed to decrease the memory footprint of the speech recognizer decoder by keeping the decoder vocabulary small. This leads to reduced acoustic confusion as well as achieving very efficient use of computational resources. Experiments on an isolated-word SMS dictation task have shown that 40% of the vocabulary prediction errors can be eliminated compared to the baseline system.

Index Terms: speech recognition, vocabulary prediction, embedded systems, isolated-word dictation

1. INTRODUCTION

Over the past decades, speech technology has advanced substantially. It is becoming a more and more important input and output method for small embedded devices. Using a voice user interface (UI) is especially convenient when the device is being used in situations where normal input methods are restricted.

For embedded devices, low memory and computational complexity implementations of the automatic speech recognition (ASR) algorithms is crucial. Even though the computational power of embedded devices is rising constantly, cost will always be an important factor in designing mass-market products. Moreover, there will always be an increasing amount of applications competing for the same computational resources as the voice UI. There are many different aspects to look at in embedded ASR systems to reduce their computational complexity and memory requirements and increase the speed of the system. Such aspects include, for example: the acoustic model and language model complexity, vocabulary size and decoder structure. A lot of work has been done optimizing performance with respect to the above mentioned parts of ASR systems. Profile compression for language model size reductions [1], probability calculation speed-up through Gaussian selection [2] and quantized HMMs for reducing storage space required for acoustic models [3] have been proposed earlier.

Research on approaches similar to what is presented in this paper has been carried out for large-vocabulary continuous speech recognition with a different context and a different goal. Adaptive vocabularies are used for reducing the number of out-of-vocabulary words for languages that have a large number of inflections in [4]. In [5], the focus is on refining the vocabulary between recognition passes based on information gathered from the previous recognition pass.

In this paper, the focus is on lowering computational complexity by decreasing the size of the active vocabulary in the decoder of an embedded isolated-word recognition system. More specifically, improvements to the vocabulary prediction algorithm of the system are proposed. Vocabulary prediction is used for generating the active vocabulary that is used for recognizing a word. The prediction is based on the words that have already been recognized. The predicted active vocabulary is a small subset of the entire vocabulary of the dictation system.

The rest of the paper is organized as follows. First, the dictation system and general vocabulary prediction method are briefly overviewed in Section 2. In Section 3, a dynamic vocabulary prediction approach is proposed to enhance the performance of vocabulary prediction. The experimental results shown in Section 4 compare the dynamic and original vocabulary prediction. Finally, the conclusions are drawn in Section 5.

2. DICTATION SYSTEM OVERVIEW

In this section, we describe our embedded isolated-word dictation system. First, the front-end processing and acoustic modeling are briefly outlined. Then we introduce the modular architecture of the system as well as word- and sentence-level decoding approach in Section 2.1. Finally, language modeling and vocabulary prediction are overviewed in more detail in Section 2.2.

The front-end of the system extracts a set of 12 MFCC coefficients and log-energy, together with their first- and second-order time derivatives, from a continuous-time speech signal sampled at 8 kHz. A feature vector normalization scheme is then applied on the features. The log-energy and its time derivatives are mean and variance normalized, and for the rest of the coefficients only mean subtraction is applied [6].

The acoustic models are decision-tree state-tied 3-state biphone hidden Markov models. Each state consists of 16 Gaussians that have been tied across states. The total number of distinct Gaussians in the system is two thousand. The continuous density Gaussian parameters have also been quantized [3].

2.1. Modular architecture

The modular design can certainly result in more useful information for improving the system performance and drive more valid conclusions about the performance of different algorithms.

This modular approach allows meaningful comparisons and the pinpointing of problems in the algorithms used in the modules. In addition, new progress can be achieved on individual modules to

978-1-4244-1746-9/07/$25.00 ©2007 IEEE

allow identifying the best techniques in the different modules and comparing different modules.

The system is designed for voice input where a clear pause is kept between words. A voice activity detection module is used for identifying word segments. For each of these word segments, a word-level Viterbi decoding is performed. The decoder vocabulary for each word segment changes from segment to segment and is built based on the previously recognized words and the language model (see Section 2.2). The word-level decoder outputs an N-best list of candidate words which are stored in a word lattice. Finally, after the whole sentence has been recognized, a sentence-level decoder carries out a search on the word lattice and outputs the final recognized sentence.

Thus as shown in Figure 1, the isolated-word dictation system is composed of three cascaded decoders. The text decoder predicts a small subset of the full recognizer vocabulary for building a recognizer network. The word decoder then performs an acoustic word-level Viterbi search on the network. The acoustic scores output from the search are appended with language model scores and the words are inserted into a lattice. The syntactic decoder uses LM to rescoring on the lattice to generate output sentence. The performance of the system can be thought of as the product of the accuracies of these three decoders.

Recognition accuracy =
(prediction rate) * (word decoding rate) * (syntactic rate)

Looking at the different accuracies, it is easy to identify the modules of the system that need the most improvement For example, if the vocabulary prediction in the text decoder does not perform well, then the prediction algorithm has to be improved or the subset vocabulary has to be enlarged. Otherwise, the power of acoustic and LM model are strictly limited because the final recognition accuracy can't be higher than the prediction accuracy. The memory and complexity can also be optimized in the module. The paper focus on efficient approach of text decoder. In next Section 2.2, text decoder is overviewed in more detail.

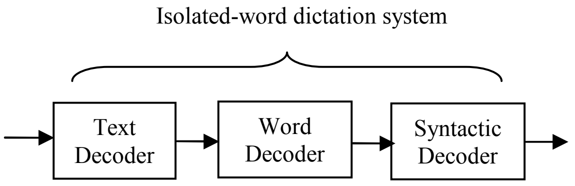

Figure 1. *Modular architecture of three cascaded decoders.*

2.2. Language modeling and vocabulary prediction

The language model used in the system consists of first- and second-order *n*-grams, i.e. unigrams and bigrams. In addition to providing probabilities for sentence modeling, the language model, or more specifically the bigram part, is used for vocabulary prediction. Vocabulary prediction is used after every recognized word to build a list of candidates used for recognizing the next word. This list of words comprises what is referred to here as the decoder or active vocabulary.

The motivation behind using a predictive scheme is to avoid taking the whole recognizer vocabulary into use during word-level decoding. This improves the speed of the decoder as well as lowers the amount of memory required. From the recognition accuracy point of view, vocabulary prediction can be seen to have a positive and a negative effect. On one hand, the reduced size of the decoder vocabulary reduces confusion during word-level decoding. On the other hand, prediction errors that exclude the correct word from the decoder vocabulary decrease performance.

Vocabulary prediction works as follows. First, a word is spoken and the decoder outputs an N-best list of possible recognized words. Then, all bigrams in the language model whose first word is one of the N-best list words are found. The predicted vocabulary for the next word is then the union of all the follower words in the previously found bigrams as shown in Figure 2.

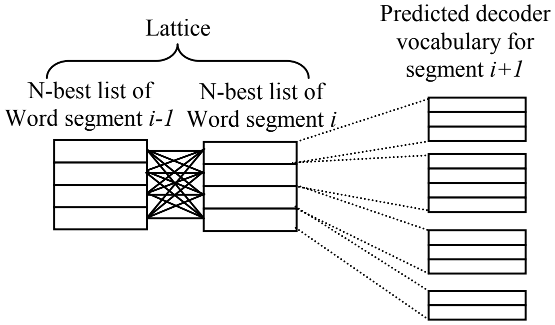
Lattice
N-best list of Word segment *i-1*
N-best list of Word segment *i*
Predicted decoder vocabulary for segment *i+1*

Figure 2: *Vocabulary prediction.*

The language model used in the current setup of the embedded system is trained on an in-house English SMS text corpus and contains approximately 32k unigrams and 400k bigrams. On average, each unigram has then about 12.5 followers in the bigram part. Thus, if the N-best list size was, for example, 8, the size of the predicted vocabulary would be on average in the order of 100 words. However, the distribution of the number of follower words is not uniform over the 32k words in the recognizer vocabulary. The words with the highest unigram probabilities tend to have the most followers. So, in fact, the size of the predicted vocabulary is much larger than the 100 words mentioned above. As will be shown in Section 4.1, the predicted vocabulary size is about one tenth of the size of the full vocabulary of the system, i.e. around 3k words. In our implementation, this corresponds to reducing the memory required for the decoder grammar to approximately one sixth of the memory required when the full vocabulary is used.

3. DYNAMIC VOCABULARY PREDICTION

The predictive scheme explained in Section 2.2 is by no means the best way of building the recognition vocabulary. Since a dictated word is not always the 1-best recognition result (or the 2^{nd} or 3^{rd} best for that matter), we have to keep N sufficiently large to have the correct word used for prediction in all cases. In most cases, though, the dictated word is actually the 1-best result. In such cases, in addition to the 1-best result, we would use the rest of the N-best list as well for prediction. This leads to larger than necessary decoder vocabularies. In addition, unwanted words are unnecessarily added to the vocabulary as they are predicted based on misrecognized words.

Based on the observations above, it would seem beneficial having a different N value for different word segments. For example, when the 1-best result is the correct one, N would be set to 1 and so on. Of course, it's not possible to know how high in the N-best list the correct result is. It is, however, possible to make an estimate based on some kind of confidence measure. This leads to the main idea presented in this paper: dynamic vocabulary prediction, where a variable number of words are chosen, based on a confidence measure, for prediction.

3.1. Confidence measures

Given a word sequence *W* and an observation sequence *X*, the posterior acoustic probability *P(W|X)* is calculated by applying the Bayes rule as shown below.

$$P(W \mid X) = \frac{P(X \mid W) \cdot P(W)}{P(X)} \qquad (1)$$

where *P(X)* is usually not considered because it is difficult to estimate reliably. This, however, introduces a problem when taking use of *P(X|W)·P(W)* as confidence measure since it no longer indicates the absolute and context independent confidence value. The problem can be solved by introducing normalization on the reference score. There are many implementations using different references such as shown in (2).

$$LLR(W \mid X) = \log\left\{\frac{P(W \mid X)}{P(\hat{W} \mid X)}\right\} = \log\left\{\frac{P(X \mid W) \cdot P(W)}{P(X \mid \hat{W}) \cdot P(\hat{W})}\right\} (2)$$
$$= \log\{P(X \mid W) \cdot P(W)\} - \log\{P(X \mid \hat{W}) \cdot P(\hat{W})\}$$

where $\hat{W}$ is the reference word sequence.

Four confidence measures have been investigated for vocabulary prediction in this paper. The first one is a simple measure which looks at the time-normalized difference of the log-likelihood of the best word hypothesis and the other word-hypotheses:

$$C_{acoustic}(i) = \frac{LL_1 - LL_i}{T} \qquad (3)$$

In the above equation*,* LL_i is the acoustic log-likelihood of the *i*th word in the N-best list of the current word segment. *T* is the length of the current word segment, which is used to have similar sized confidence values regardless of the word length. The measure is used such that all words in the N-best list whose confidence is above a certain threshold will be selected for vocabulary prediction. In this case, this means that all words whose log-likelihood score is close to the score of the best hypothesis are selected. Note that here we are thus assuming that one of the words in the N-best list is always correct and when one or more scores are close to the best score we are not sure which one of these is correct.

The next confidence measure considered is similar to the one shown in (3). The difference is that here we have the average of the log-likelihood scores in the N-best list as the reference value instead of the best log-likelihood value:

$$C_{avgacoustic}(i) = \frac{LL_{avg} - LL_i}{T} \qquad (4)$$

The above confidence measures are purely based on the acoustic log-likelihood scores of the words in the N-best list of the current word segment. We do, however, have more information available for us to base the confidence measure on. For example, token-scores, calculated from the beginning of the sentence to the current word segment, carry language model score information as well as the acoustic scores. So, the confidence measures can be calculated using the token-score, rather than the acoustic log-likelihood as shown in (5) and (6).

$$C_{token-score}(i) = \frac{TS_1 - TS_i}{t} \qquad (5)$$

$$C_{avgtoken-score}(i) = \frac{TS_{avg} - TS_i}{t} \qquad (6)$$

Notice that in (5) and (6) the normalization is still done using the length of the current word. This results in both the acoustic model and the language model probabilities being normalized. The reason for applying the normalization in this manner is that this way the ordering of words does not change whether it is done by the token-score or the confidence value. As the N-best list is obtained by ordering based on the token-scores it would otherwise be possible that words not appearing in the N-best list are used for prediction.

4. EXPERIMENTS

In this section, results of speech recognition experiments using dynamic vocabulary prediction are shown. The performance of the various test set-ups are measured in terms of both prediction accuracy and recognition accuracy. Prediction accuracy indicates how often the recognition vocabulary contains the word that is to be recognized. The test set used in the experiments contains a total of 5500 SMS messages (100,000 words) from 23 US English speakers (male and female). The speakers have been selected so that different dialect regions and age groups are well represented.

4.1. Baseline

The prediction accuracy of the baseline system is 96.44%. This is obtained by using 8-best words in each word segment for predicting the vocabulary for the next segment. This results in the average vocabulary size for each word segment to be 3240 words. Table 1 shows the prediction accuracies and average vocabulary sizes when using a different number for words for prediction. The results are intuitive; as the number of words used for prediction is lowered, the average vocabulary size is decreased. This results then in lower prediction accuracy.

Table 1: *Baseline prediction accuracy vs. average vocabulary size.*

# of words used for prediction	Average predicted vocabulary size	Prediction accuracy
8	3240	96.44%
6	3028	96.17%
4	2733	95.65%
3	2531	95.11%
2	2258	93.94%

4.2. Confidence measure comparison

The baseline results showed that the average decoder vocabulary size can be made smaller by using fewer words for prediction. This, however, introduces prediction errors. Next, we try out the four confidence measures (Equations 3-6) to see if we can improve the prediction accuracy without increasing the average vocabulary size.

Figure 3 and Figure 4 show prediction accuracy vs. vocabulary size for the different confidence measures, obtained by varying the confidence thresholds. Figure 3 shows the results for the confidence measures that use the best score for reference ($C_{acoustic}$ and $C_{token\text{-}score}$). Figure 4 shows the results for the other two confidence measures ($C_{avg\_acoustic}$ and $C_{avgtoken\text{-}score}$). For comparison, the accuracy when using a fixed number of words for prediction is also shown in both figures. This baseline figure uses the values from Table 1.

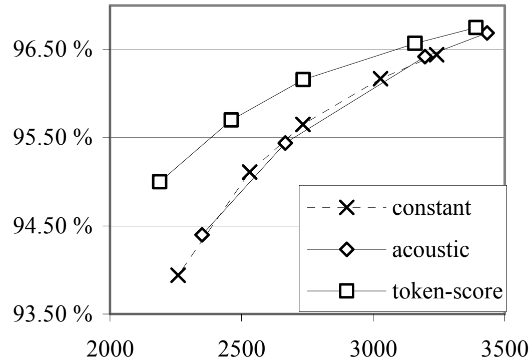

Figure 3: *Prediction accuracy vs. average vocabulary size for $C_{acoustic}$ and $C_{token\text{-}score}$.*

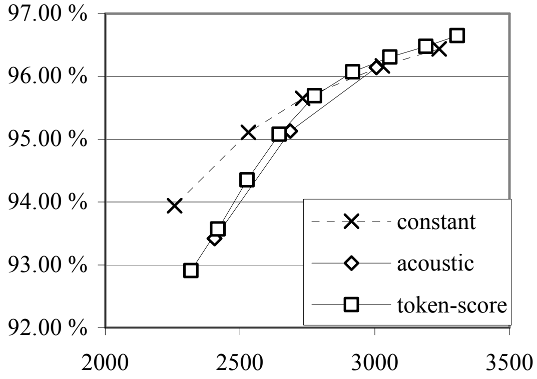

Figure 4: *Prediction accuracy vs. average vocabulary size for $C_{avgacoustic}$ and $C_{avgtoken\text{-}score}$.*

From Figure 3 and Figure 4 we can see that the best choice for confidence measure is $C_{token\text{-}score}$, shown in (3). While it performs the best over the whole range of vocabulary sizes shown here, the largest increase in performance is seen for the smallest vocabulary sizes. Prediction accuracies for $C_{acoustic}$ do not differ from the baseline prediction accuracies significantly. Both of the confidence measures that use the average of the N-best scores as the reference score have the worst performance as shown in Figure 4. Their performance is actually lower than that of the baseline. This is seen especially at the lower vocabulary sizes. Based on the above results, $C_{token\text{-}score}$ will be used as the confidence measure for the following experiments.

4.3. Adding a minimum predicted vocabulary size

The prediction errors are partially due to words that are not well represented in the bigram part of the LM. Such words do not have many continuations and thus, do not contribute much to the predicted vocabulary. If the prediction is done solely on such words, the predicted vocabulary is small and the risk of a prediction error is high. In addition, the LM training corpus can not cover all the cases in real-life use, resulting in a mismatch between training and testing.

This problem can be alleviated by setting a minimum predicted vocabulary size. If the decoder vocabulary size after prediction is below a minimum value, commonly used words (based on the unigram part of the LM) are appended until the minimum value is reached. Figure 5 shows prediction accuracy vs. average vocabulary size when using a minimum vocabulary setting and the $C_{token\text{-}score}$ as the confidence measure for vocabulary prediction. The minimum vocabulary sizes used here were 250, 500 and 1000.

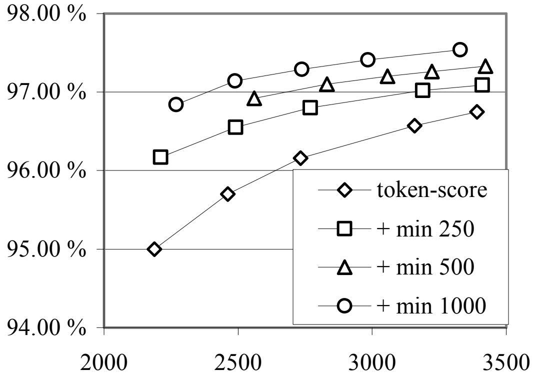

Figure 5*: Effect of having a minimum vocabulary size.*

The results shown above indicate that using dynamic vocabulary prediction with a minimum vocabulary setting outperforms just using dynamic vocabulary prediction. This is partly due to solving the problem of very small predicted vocabularies mentioned above. In addition to this, some of the gain in accuracy is due to the fact that the words appended to the predicted vocabulary are the words with the highest unigram probability. Thus they are likely be good candidates following any recognized word.

4.4. Effect on recognition accuracy

So far, only the prediction accuracy has been considered. Here we present word recognition accuracies for selected setups from the above tests. The recognition accuracies are of course lower than the prediction accuracies as mistakes are introduced in the word-level decoding as well as the sentence-level decoding. Figure 6 shows the recognition accuracies for the baseline system as well as two different dynamic vocabulary prediction setups. One of these has a minimum vocabulary setting and the other does not. Both of the dynamic vocabulary prediction setups use $C_{token\text{-}score}$ for the confidence measure. As it can be seen, the recognition accuracy figures look similar to the vocabulary prediction figures in terms of relative trend.

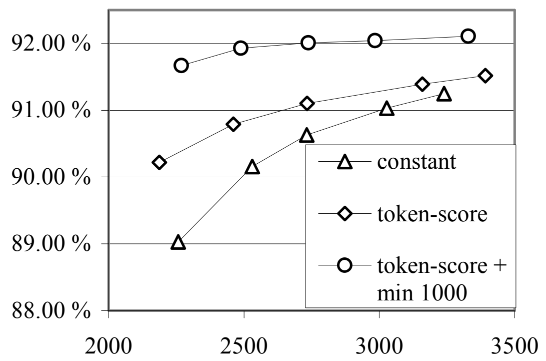

Figure 6: *Recognition accuracy vs. average vocabulary size.*

4.5. Modular accuracies

In Section 2.1 we discussed the modular architecture of our recognition system. *Table 2*, below, shows the accuracies of the three components for the same systems as shown in Figure 6. The average vocabulary size for each of the systems was set to approximately 2700. As it can be seen, the dynamic vocabulary prediction improves the vocabulary prediction rate while the word decoding and syntactic rates remain unchanged.

Table 2: Accuracies of the three different system components.

Prediction	Prediction rate	Word decoding rate	Syntactic rate	Recognition accuracy
constant	95.65%	98.69%	96.01%	90.63%
Token-score	96.16%	98.67%	96.01%	91.10%
Token-score + min 1000	97.29%	98.61%	95.91%	92.01%

4.6. Experiments with a static vocabulary

All of the experiments presented so far have been done using some form of vocabulary prediction. It is, of course, possible to recognition with a static vocabulary, i.e. by using the same vocabulary for each word segment. Table 3, below, lists the vocabulary prediction and recognition accuracies for several different sized static vocabularies. The same language model is used as in the previous experiments. The words in the vocabulary are determined by the unigram section of the language model.

Table 3: *Prediction accuracy vs. average vocabulary size for static vocabulary systems.*

Vocabulary size	Prediction accuracy	Recognition accuracy
3500	96.86%	91.53%
3000	96.37%	91.20%
2500	95.74%	90.78%
2000	94.96%	90.23%

From Table 3 we can see that the vocabulary prediction and recognition accuracies are higher than the baseline accuracies (Table 1 and Figure 3). However, when compared to the dynamic vocabulary prediction scores (token-score + minimum vocabulary size, Figure 5 and Figure 6), the scores are lower.

When comparing systems with static and dynamic vocabularies, looking at just the vocabulary size might not be the best choice for measuring complexity. This is due to the fact that in a static vocabulary system there is no need to rebuild the decoder tree for every word segment and the time needed to build the decoder tree is saved. In the current version of our embedded SMS dictation system running on a mobile phone [7], however, the time to build the decoder tree is less than one tenth of the time that is required for decoding a segment of 100 frames (1 second of speech). Thus, for the same vocabulary size, decoder building and decoding on the dynamic system is less than 10% slower than a static system. However, with the same average vocabulary size, the dynamic vocabulary based system outperforms the static vocabulary one (approx. 97.1% vs. 95.7% prediction accuracy at 2500 average vocabulary size). To get the same performance as the dynamic vocabulary prediction scheme, when the average vocabulary size is the same 2500, the static vocabulary would need to be larger than 3500. In this case the static system would be slower.

5. CONCLUSSIONS

In this paper, we proposed dynamic vocabulary prediction for improving vocabulary prediction in our embedded isolated-word dictation system. In this approach a confidence measure is used to determine the number of words used for prediction instead of using a constant number.

Compared to the baseline system, dynamic vocabulary prediction achieved improved vocabulary prediction rates. This was especially true when reducing the average predicted vocabulary size. For an average predicted vocabulary size of around 2500, for example, the proposed algorithm had a prediction accuracy of 95.70% compared to 95.11% of the baseline system. When a minimum vocabulary size setting was used in conjunction with dynamic vocabulary prediction, even better results were obtained. At the same average vocabulary size of approximately 2500, this setup reached a prediction rate of 97.00%. Compared to the baseline, this corresponds to approximately a 40% prediction error rate reduction. It was also shown that the word recognition accuracy of the system goes hand in hand with the prediction accuracy. In other words, improvements in the vocabulary prediction rate improved the word recognition rate as well.

6. REFERENCES

[1] J. Olsen and D. Oria, "Profile Based Compression of N-Gram Language Models," in Proceedings of ICASSP 2006, Toulouse, France, vol 1, pp. 1041-1044, 2006.

[2] E. Bocchieri, "Vector Quantization for Efficient computation of continuous density likelihoods," in Proceedings of ICASSP 1993, Minneapolis, MN, USA, vol. 2, pp. II-692 – II-695, 1993.

[3] M. Vasilache, "Speech Recognition Using HMMs with Quantized Parameters," in Proceedings of Eurospeech 2001, Aalborg, Denmark, vol. 2, pp. 1265-1268, 2001.

[4] P. Geutner, M. Finke, and P. Scheytt, "Adaptive vocabularies for transcribing multilingual broadcast news," in Proceedings of ICASSP, Seattle, USA, pp. 925–928, 1998.

[5] I. L. Hetherington, “A multi-pass, dynamic-vocabulary approach to real-time, large-vocabulary speech recognition,” in Proceedings of Interspeech 2005, Lisbon, Portugal, pp. 545-548, 2005.

[6] O. Viikki, D. Bye, and K. Laurila, “A Recursive Feature Vector Normalization Approach for Robust Speech Recognition in Noise,” in Proceedings of ICASSP 1998, Seattle, Washington, USA, pp. 1692-1695, 1998.

[7] E. Karpov, I. Kiss, J. Leppänen, J. Olsen, D. Oria, S. Sivadas, and J. Tian, ”Short Message Dictation on Symbian Series 60 Mobile Phones,” in Proceedings of SiMPE 2006, Espoo, Finland, 2006.