Fixed-Point Implementation of Isolated Sub-Word Level Speech Recognition Using Hidden Markov Models

Venkatesh N
Innovation Lab
Tata Consultancy Services
Bangalore, India
venk.n@tcs.com

Ruchir Gulati
Innovation Lab
Tata Consultancy Services
Kolkata, India
ruchir.gulati@tcs.com

Rajeev Bhujade
Innovation Lab
Tata Consultancy Services
Bangalore, India
rajeev.b@tcs.com

Girish Chandra M
Innovation Lab
Tata Consultancy Services
Bangalore, India
m.gchandra@tcs.com

ABSTRACT
This paper presents a limited vocabulary isolated-word speech recognition system based on Hidden Markov Model (HMM) that involves two stage classification and is implemented on Texas Instruments' (TI) DaVinci embedded platform for a home infotainment system. A methodology using simple metric has been proposed for segmenting the words into sub-word units and these sub-words are used in the second stage to improve recognition accuracy. Also, a simple and efficient way of handling the out-of-vocabulary words using an additional HMM model is presented. We have achieved recognition accuracy of around 89% for a fixed point implementation on the TI DaVinci platform, demonstrating its suitability for embedded systems.

Keywords
Speech recognition, Sub-word, HMM, TI DaVinci, Set-Top box.

1. INTRODUCTION
Serving a consumer market hungry for high-definition digital content, set-top box manufacturers are really challenged to provide the end-user with a high-end entertainment experience. Simplistic and intuitive ways to interact with such systems go a long way towards fulfilling this goal. Voice is a natural medium for humans to interact and thus voice commands can be explored in this context to achieve the above mentioned goal. In addition, they also prove to be very useful for visually challenged people. While speech recognition has reached fairly usable levels on the PC platform, efforts are still ongoing to implement it on cheaper platforms found in appliances/equipments of day-to-day use.

Early speech recognition systems tried to apply a set of syntactical grammatical rules to recognize speech. If the words spoken fitted into a certain set of rules, the system could determine what the words were. However, human language has numerous exceptions to its own rules, even when it is spoken consistently. Accents, voice modulation and gestures can greatly influence the effectiveness of verbal communication.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SAC'11, March 21-25, 2011, TaiChung, Taiwan.
Copyright 2011 ACM 978-1-4503-0113-8/11/03…$10.00.

Isolated-word recognition refers to the task of recognizing a single spoken word where the choice of words is not constrained to task syntax or semantics [1][2][3]. Early works have reported isolated-word speech recognition using primitive methods like Dynamic Time Warping (DTW)[1][4]. The main drawback of using this system is that its computational time complexity increases with the increase in dictionary size, which makes it less suitable for real world applications.

Modern speech recognition systems use powerful and complicated statistical modeling. These systems use probability and mathematical functions to determine the most likely outcome. The two models that dominate the field of speech recognition today are neural networks and Hidden Markov Model (HMM). Briefly, the HMM approach is a well-known and widely used statistical method of characterising the spectral properties of the frames of a pattern and is particularly suitable for speech recognition [1][2][5].

We aim at developing a real-time voice command-and-control application focusing on isolated-word speech recognition on the TI DaVinci platform. The latter has a low component price, making it well-suited for embedded systems. However, it must be noted that applying speech recognition for the set-top boxes poses some unique challenges. Firstly, the recognition has to be carried out amidst highly varied TV environment as background - from news to action movies to music and more. Secondly, the words may be uttered under different contexts from the user, for instance while drinking, eating etc. This calls for robust detection techniques. Further, to provide enhanced user experience the words chosen in the dictionary may be more prone to classification error. In consideration of these issues, the proposed system uses an HMM based two-level speech recognition method, involving word level recognition at the first stage, and sub-word level recognition at the second stage for improving the recognition accuracy. Second stage classification is performed only for the pre-identified set of confusion words present in our dictionary. For segmenting the words into sub-word units, we propose an algorithm based on energy peaks of the speech signal. A technique has also been suggested to handle out-of-vocabulary (OOV) words which play a crucial role in limited vocabulary speech recognition applications. The techniques described are designed to be simple and efficient to run on embedded platforms which make them suitable for real-time applications.

In efforts to explain the proposed system in detail the organization

of the paper is as follows. Section II describes the system overview. Sections III and IV cover pre-processing, feature extraction and segmentation modules respectively. HMM based classifier is explained in section V followed by methodology of its working in section VI. Fixed-point implementation of this system is explained in section VII. The results are described in section VIII. Finally conclusion and future work is discussed in section IX.

2. OVERVIEW OF THE SYSTEM

We envisage the use of speech (isolated word) recognition for a command-and-control system to access the functions of a Home Infotainment Platform (HIP) box. HIP is built around the Texas Instruments' DaVinci platform. It is essentially a Set-Top Box with extended features like Internet browsing, Short Messaging Service, media playback, etc. The speech recognition algorithm runs on a fixed-point Digital Signal Processor (DSP). As a first cut implementation we have selected the following commands: 'Menu', 'Play', 'Louder', 'Softer', 'Mute', 'Stop', 'Pause', 'Movie', 'Music' and 'TV', the utterance of which, the algorithm recognizes. This vocabulary is selected based on a survey conducted at our lab to control media playback and volume functions, Video, TV, and Music Player features. It focuses on selecting simple words to represent the action that the end users want to take place on HIP. Going forward, the vocabulary shall be extended to cover the complete features and functions of HIP. It must also be noted that if none of the above words are spoken, and anything else is spoken or is audible, it must be rejected. This is one of the important requirements [6].

The proposed system has two phases as shown in Figure 1. During the training phase, the input speech data is pre-processed and features are extracted. Using this feature set, two set of HMM models are generated, one for the complete set of words including out-of-vocabulary input and second for sub-words. Generally out-of-vocabulary words are rejected using likelihood score (LLS) [7] but in our case we have build a separate HMM model (OOV MODEL) using the commonly spoken words in the home environment and this is seen to perform better than the heuristic methods.

During the testing phase, the test speech sample is preprocessed and the extracted features are fed to the first stage classification which involves the complete set of word-level HMM models. If the output of the first stage classification falls into the first group of words then it is treated as the final output or else if it falls into the second group then the test sample is segmented into sub-words using the segmentation algorithm. Features are extracted from these sub-words and fed to the second stage classifier whose output is treated as the final output.

One simple solution to the set of confusion words problem is to substitute them with another set of distinct words. However, we did not proceed with this approach because the voice commands selected are user friendly and were chosen based on the end user survey, as remarked earlier. Hence we decided to handle the set of confusion words by adopting the sub-word level recognition concept.

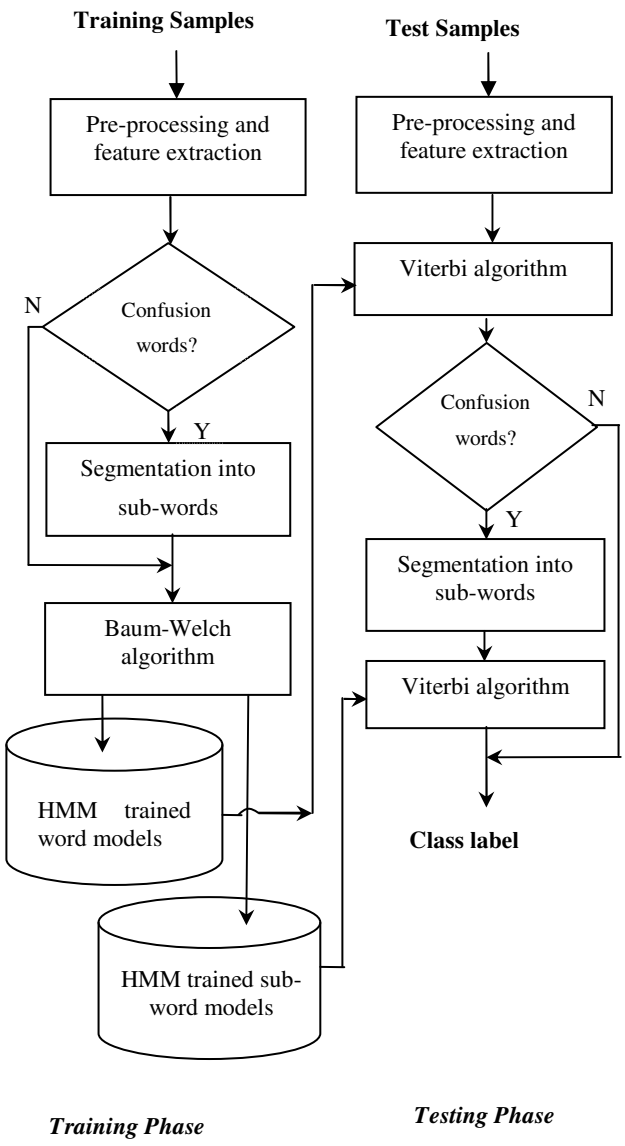

Figure 1. Overview of the proposed system.

3. PRE-PROCESSING AND FEATURE EXTRACTION

During the preprocessing stage, the speech signal is blocked into frames using Hanning window [8]. We incorporated a frame duration of 20 ms with a 50% overlap.

3.1 Mel-Frequency Cepstral Coefficients

Mel-frequency Cepstral coefficients (MFCCs) are a compact, perceptual based representation of speech frames which are capable of discriminating the spoken words and are robust towards speaker change/ambient noise. Hence we have chosen MFCCs as features [5].

Given an audio signal divided into short time windows, the Discrete Fourier transform (DFT) of each time window is computed as:

$$X(k) = \sum_{n=0}^{N-1} w(n)\, x(n) \exp(-j2\pi kn / N) \qquad (1)$$

for k=0 ,1,2…N-1 where N is the length of speech segment in samples, k corresponds to frequency $f(k) = \mathrm{k}\, f_s / N$, f_s being the sampling frequency in hertz and $w(n)$ is a time window typically Hanning window. The magnitude spectrum $|X(k)|$ is scaled thus in frequency and magnitude. First, the frequency is scaled logarithmically using the Mel filter bank $H(k,m)$ and then log is taken, giving

$$X'(m) = \ln\left(\sum_{k=0}^{N-1} |X(k)| \cdot H(k,m)\right) \qquad (2)$$

for $m = 1,2,3...M$ where M is the number of filter banks with $M << N$. The Mel filter bank is a collection of triangular filters defined by center frequencies $f_c(m)$ and can be put as

$$H(k,m) = \begin{cases} 0 & for\ f(k) < f_c(m-1) \\ \dfrac{f(k) - f_c(m-1)}{f_c(m) - f_c(m-1)} & for\ f_c(m-1) \le f(k) < f_c(m) \\ \dfrac{f(k) - f_c(m+1)}{f_c(m) - f_c(m+1)} & for\ f_c(m) \le f(k) < f_c(m+1) \\ 0 & for\ f(k) \ge f_c(m+1) \end{cases} \qquad (3)$$

The center frequencies are computed by approximating the Mel scale with $\phi = 2595 \log_{10}\left(\frac{f}{700} + 1\right)$. This equation is non linear for all frequencies. A fixed frequency resolution Mel scale is computed corresponding to the logarithmic scaling of the repetition frequency using $\Delta\Phi = (\Phi_{\max} - \Phi_{\min}) / (M+1)$ where $\Phi_{\max}$ and $\Phi_{\min}$ are the highest and lowest frequency of the filter bank on the Mel scale, computed from $f_{\max}$ and $f_{\min}$ respectively and *M* is the number of filter banks.

Finally the MFCCs are obtained by computing the DCT of $X'(m)$ as:

$$c(l) = \sum_{m=1}^{M} X'(m) \cos\left(l \frac{\pi}{M}(m - \frac{1}{2})\right) \qquad (4)$$

for $l = 1,2...,M$ where $c(l)$ is the l^{th} MFCC.

A window length of 20ms with 10 MFCC coefficients has been used. Using this feature we can utilize the HMM framework to design the speech recognition system. However, since our dictionary consists of a set of confusion words, in order to better detect among this set of confusion words, we have incorporated sub-word level recognition. The segmentation of words into sub-words is carried out by proposing a simple and efficient technique presented in the following section.

4. SEGMENTATION

To the best our knowledge systems involving sub-word level recognition create sub-word units like phoneme/syllable and involve computationally expensive Viterbi alignment algorithm for accurate segmentation thus making them unsuitable for embedded applications. To facilitate easy implementation on embedded platforms we propose a simple segmentation algorithm based on energy peaks as follows:

Let 'x' be the speech signal representing one complete word. We compute short-term energy per frame as follows:

$$E_i = \frac{1}{N} \sum_{n=0}^{N-1} x_i(n)^2 \qquad (5)$$

i is the frame index running till F which is total number of frames and *N* is the frame length. Once we obtain energy values for all the frames of the utterance, we compute energy peaks E_p using E_i and further we find the average of first maximum energy peak time instance t_1 and second maximum energy peak time instance t_2 as given in:

$$T = \frac{(t_1 + t_2)}{2} \qquad (6)$$

Further, words are segmented into sub-words using time instance '*T*' as follows:

$$x_1 = x(1:T)$$
$$x_2 = x(T:end)$$

where, x_1 is a vector consists of samples values from 1 to *T*, forming the first part of the word, and x_2 is a vector containing the sample values from *T* to end of the signal forming the second part of the word. We obtain the sub-word units by splitting the word into two halves for example; 'MENU' is segmented into two sub-word units 'ME' and 'NU' as shown in the Figure 2.

5. HIDDEN MARKOV MODEL (HMM) FOR CLASSIFICATION

An HMM is a composition of two stochastic processes, a hidden Markov chain, which accounts for temporal variability, and an observable process, which accounts for spectral variability. This combination has proven to be powerful enough to cope with the most important sources of speech ambiguity, and flexible enough to allow the realization of recognition systems with dictionaries of tens of thousands of words. In our experiment, continuous left to right HMMs were used to model the words/sub-words. To estimate the HMM model parameters we train the model using well known Baum-Welch algorithm [10].

The number of Gaussians mixtures and the number of states in the HMM were determined empirically. The samples for training are chosen to account for a good mix of male and female speakers along with different accents and pronunciations in different conditions.

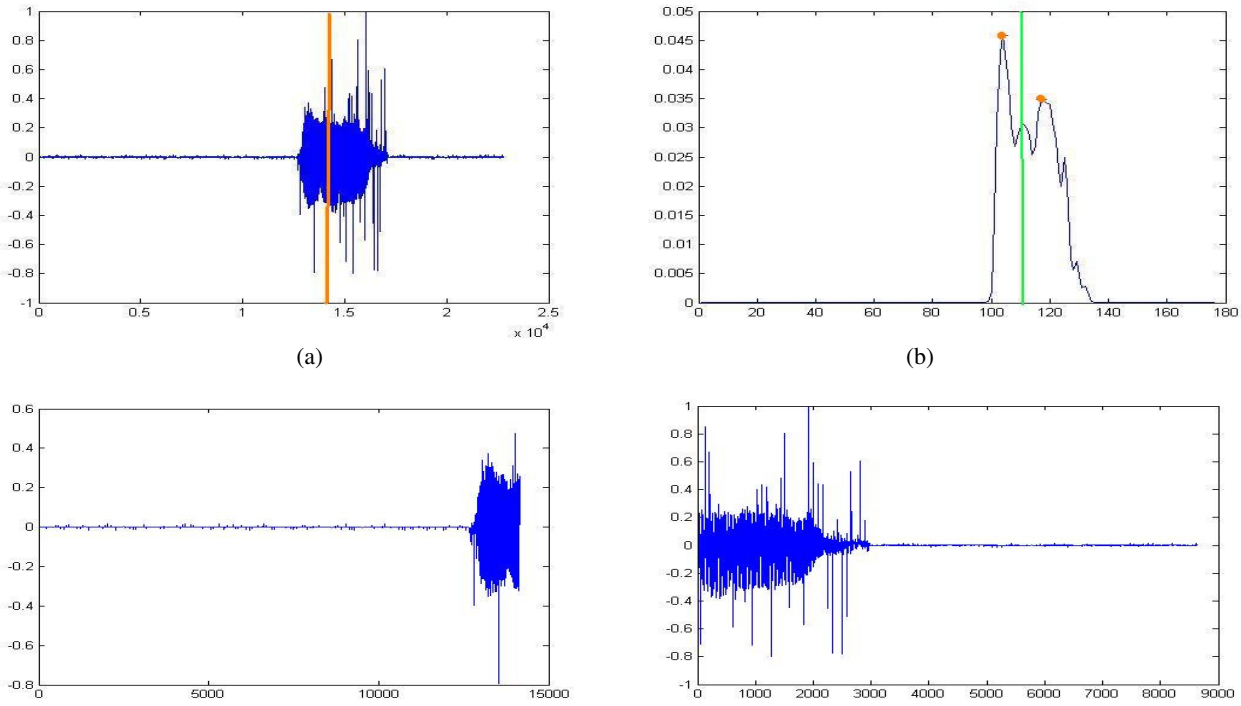

Figure 2. Segmentation of "menu" into "me"+"nu" (a) "MENU" (b) Energy plot of word "MENU" (c) sub-word "ME" (d) sub-word "NU"

Given a test utterance, by applying the Viterbi algorithm [10][11], we calculate the maximum posterior probability associated with the word/sub-word level, conditioned on each of the HMM models. The model resulting with the maximum posterior probability is declared as the classification result.

6. METHODOLOGY

6.1 Training

We collected a modest set of utterances confirming to the above considerations. We split the words into two groups.

The first group consists of words 'Play', 'Louder', 'Softer', 'Stop', 'Pause' , 'TV' and 'OOV' and the second group consists of words which are prone to confusion namely 'Menu', 'Mute', 'Movie' and 'Music'. Two set of HMM models are trained. The first set consisting of 11 HMM models was developed at word level for entire set of words present in the vocabulary also including 'OOV' model (out-of-vocabulary words). The second set of 7 HMM models was developed at sub-word level for segmented words in the second group (the sub-words are 'me', 'mo', 'mu', 'sic', 'nu' and 'te').

6.2 Testing

The test vector is input to the system. At the first stage, it is passed through all the word level HMM models. If the first stage classification output lies in the first group of words then it is declared as the final output. On the other hand if the first classification output is one among the second group of words then the test vector is segmented into sub-words followed by pre-processing and feature extraction. The extracted features are fed to the second stage classifier which makes use of only sub-word level HMM models for computing the maximum posterior probability. The outputs from the second stage classification are combined using a simple set of rules to yield the corresponding class label. For the chosen dictionary, the rule simply tries to give more weightage to the second part of the word. As depicted in Figure 3, for the set of words chosen, this strategy will lead to better classification of the words which are more prone to confusion.

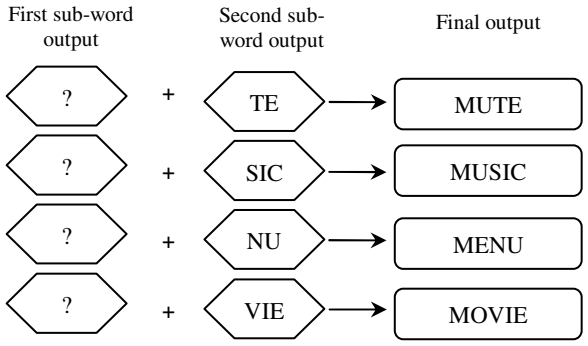

Figure 3. Sub-words combining rules to yield final output

7. FIXED POINT IMPLEMENTATION

The speech recognition algorithm is developed in floating-point, and it is ported on TI's DaVinci platform, which is a dual-core architecture consisting of an ARM CPU along with a fixed-point DSP clocked at 600MHz. Hence the algorithm should first be converted to fixed-point so that it can run in real-time on the DSP core of DaVinci. The analysis in [12] shows that using different Q-formats for different quantities of the algorithm gives an overall reduction of 29.7% in the circuit size.

$$Q[QI].[QF] \tag{7}$$

where QI = number of integer bits including one sign bit, QF = number of fractional bits with $QI+QF$ = word length.

Accordingly, we have used an approach where the algorithmic resolution and dynamic range of each quantity is determined and the conversion is carried out, thereby picking the optimal Q-format. For example, the speech input is pre-processed in the Q15 format, the Hanning window is in Q24, FFT input in Q29, and IFFT input in Q24 formats. To perform the conversion, we have used the "IQ Math" library provided by Texas Instruments. This library provides for representing a floating-point number in any Q-format with a 32-bit word length in all. Additionally, it handles arithmetic and mathematical operations of floating-point numbers represented in Q-formats. As with any other floating-to-fixed-point conversion procedure, there is a loss of accuracy due to quantization here. Hence care is taken that the resultant loss in accuracy is within an acceptable range of error, thus preserving the overall trend of scores produced by the algorithm.

In addition to this, to improve run-time performance, the FFT and IFFT implementations provided by Texas Instruments that are optimized for the said platform have been used in the second processing stage. Before incorporating the platform optimized implementations, the input to the algorithms was converted to fixed-point and evaluated for different Q-formats in MATLAB. A typical example of this can be seen for one set of IFFT output points in Figure 4. Here, the lowest plot is of the floating-point output, the middle plot is of the output corresponding to the Q24 formatted input, and the topmost one is the output of the integrated platform optimized IFFT library and Q24 formatted input.

The input Q-format was finally selected such that it produces the same trend for the output at all points, with a fixed scale factor, as the floating-point input.

All conversions and operations are done focusing on preserving the overall trend of probability scores produced by the floating-point algorithm that we started with. This can be seen from the trend of scores produced by the floating-point algorithm in Figure 5, and that produced by the ported fixed-point algorithm in Figure 6 respectively.

8. RESULTS AND DISCUSSION

The speech recognition system was trained and tested for limited vocabulary under various TV scenarios. The training set contained a total of 6,000 samples collected from 110 users with 5 trials each for all the ten classes present in the vocabulary and approximately 500 samples were collected which consists of words commonly spoken in home environment for developing OOV HMM model during training process. The test set contained approximately 150 samples per class present in the vocabulary and around 200 samples for out-of-vocabulary words amounting to 1,700 samples collected from a different set of 30 users.

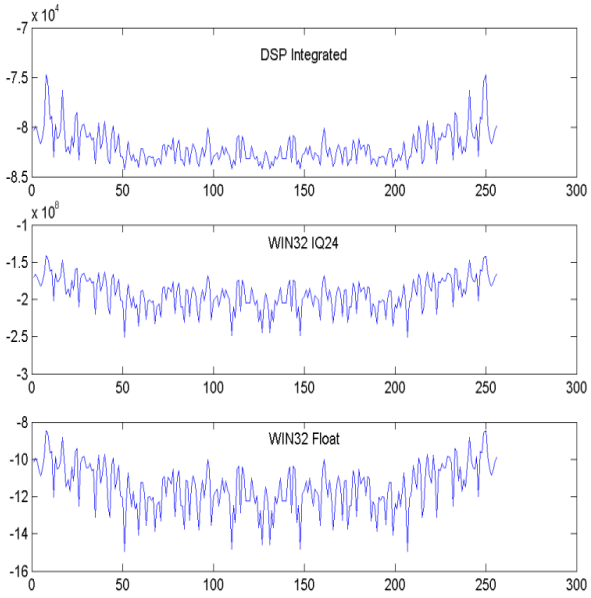

Figure 4. IFFT Output Plots

Empirically we found that 15 number of states per HMM and the 12 Gaussians mixtures yielded better results. From Figure 7 we observe that the recognition accuracy has significantly increased by around 8-10% at sub-word level recognition when compared to complete word-level recognition. This is because the words from second group sound similar, for example, 'Music' and 'Mute' are same in the beginning but only differs at endings. This causes the confusion and hence we have adopted a sub-word level recognition for yielding better recognition accuracy.

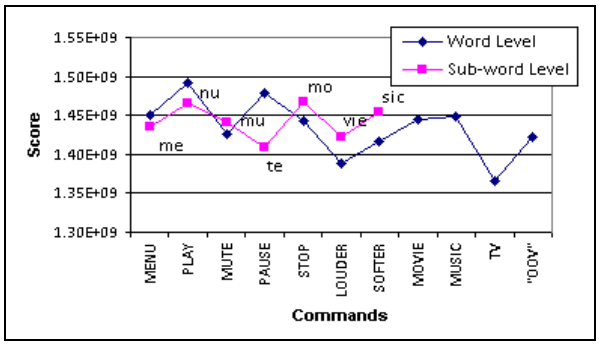

Figure 5. Output scores for floating point implementation on x86.

Table 1 summarizes the results obtained for both floating-point and fixed-point implementations and we observe that the average recognition accuracy for floating-point implementation is around 93% and for fixed-point implementation it is around 89%. It is clearly observed that the floating-point implementation has higher recognition accuracy as compared to the fixed-point implementation for each and every word because of the limited resolution that can be achieved by any fixed-point representation of a real number. We also observe that for out-of-vocabulary words case, the system has yielded a recognition accuracy of

around 90% which clearly indicates that the system has performed well in rejecting the out-of-vocabulary words making it suitable for the real-world applications.

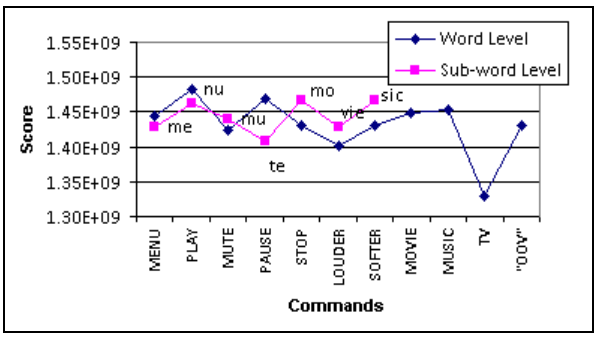

Figure 6. Output scores for fixed-point implementation ported on DSP.

9. CONCLUSION AND FUTURE WORK

A limited vocabulary isolated-word speech recognition system based on HMM has been designed. The use of sub-word level recognition for set of confusion words has yielded better results when compared to word-level recognition. We have proposed a new technique for handling out-of-vocabulary words which serves as one of the important aspect of limited vocabulary speech recognition. The fixed-point implementation of the proposed system has been executed on TI DaVinci platform, demonstrating the suitability for real-word embedded applications.

Table 1. Recognition Accuracy results

Voice commands	Floating point accuracy (%)	Fixed point accuracy (%)
'Menu'	89.6	86.8
'Play'	96.4	92.7
'Louder'	93.3	90.1
'Softer'	93.8	89.6
'Mute'	92.2	88.1
'Stop'	97.1	93.4
'Pause'	95.7	91.9
'Movie'	90.3	87.7
'Music'	91.3	86.4
'TV'	95.6	91.3
Out-of-vocabulary (OOV)	90.4	87.6
Average	**93.2**	**89.6**

In the future work we would be concentrating on increasing the database so that it covers the samples of all age groups and a good mixture of samples from males and females. With our proposed system, we would like to increase the vocabulary size and try to build a voice command system independent of language. The accuracy of the system is expected to increase further by incorporating noise reduction algorithms.

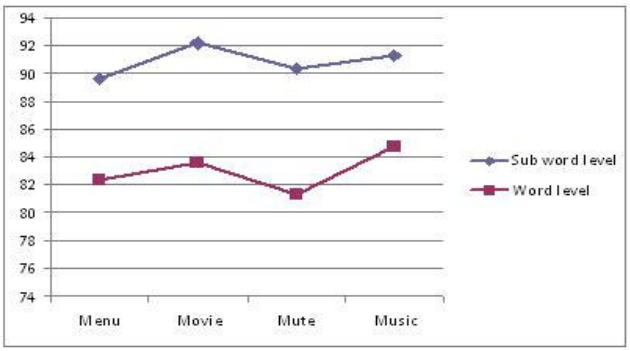

Figure 7. Recognition accuracy for word level and sub-word level recognition.

10. REFERENCES

[1] Sakoe, H. & S. Chiba. (1978), Dynamic programming algorithm optimization for spoken word recognition. IEEE, Trans. Acoustics, Speech, and Signal Proc., Vol. ASSP-26.

[2] Titus Felix FURTUNĂ, "Dynamic Programming Algorithms in Speech Recognition", Revista Informatica Economică nr. 2(46)/2008.

[3] Dong Wang , Liang Zhang, Jia Liu and Runsheng Liu , "Embedded speech recognition system on 8-bit MCU core", Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04). , 2004.

[4] L. Deng and H. Strik, "Structure-Based and Template-Based Automatic Speech Recognition --- Comparing parametric and non-parametric approaches", Microsoft Research, One Microsoft Way, Redmond, WA, USA, CLST, Department of Linguistics, Radboud University, Nijmegen, Netherlands.

[5] Psutka, J., M¨uller, L., Psutka, J. V., "Comparison of MFCC and PLP Parameterization in the Speaker Independent Continuous Speech Recognition Task", Proceedings of Eurospeeech 2001, A° lborg, 2001, pp. 1813–1816.

[6] C. White, G. Zweig, L. Burget, P. Schwarz, H. Hermansky, "Confidence Estimation, OOV Detection and Language ID using Phone-to-Word Transduction and Phone-Level Alignments", proc. IEEE International Conference on Acoustics, Speech and Signal Processing, April 2008.

[7] L.R.Rabiner, S.E.Levinson and M.M.Sondhi, "On the application of Vector Quantization and Hidden Markov Models to speaker independent, isolated word recognition," The Bell System Technical Journal, Vol.62, No.4, April 1983.

[8] L. R. Rabiner and B. H. Juang, Fundamentals of Speech Recognition, Prentice Hall (Signal Processing Series), 1993.

[9] Mayukh Bhaowal and Kunal Chawla, "Isolated Word Recognition for English Language Using LPC, VQ and HMM," IFIP WCC '04, pp. 343-352, August 2004.

[10] L. R. Rabiner, Fellow IEEE, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", proc. IEEE, Volume 77, Issue 2, Feb. 1989, pp. 257-286.

[11] Phil Blunsom, "Hidden Markov Models" retrieved from http://ww2.cs.mu.oz.au/460/2004/materials/hmm-tutorial.pdf, 2004.

[12] Y.M. Lam, M.W. Mak and P.H.W. Leong, ``Fixed point implementations of Speech Recognition Systems,'' Proceedings of the International Signal Processing Conference, Dallas, 2003.