ICMI'12 Grand Challenge – Haptic Voice Recognition

Khe Chai Sim
National University of Singapore
13 Computing Drive
Singapore 117417
simkc@comp.nus.edu.sg

Shengdong Zhao
National University of Singapore
13 Computing Drive
Singapore 117417
zhaosd@comp.nus.edu.sg

Kai Yu
Shanghai Jiao Tong University
800 Dongchuan Road
Shanghai 200240
P.R. China
kai.yu@sjtu.edu.cn

Hank Liao
Google
76 Ninth Avenue
New York, NY 10011
hankliao@google.com

ABSTRACT

This paper describes the Haptic Voice Recognition (HVR) Grand Challenge 2012 and its datasets. The HVR Grand Challenge 2012 is a research oriented competition designed to bring together researchers across multiple disciplines to work on novel multimodal text entry methods involving speech and touch inputs. Annotated datasets were collected and released for this grand challenge as well as future research purposes. A simple recipe for building an HVR system using the Hidden Markov Model Toolkit (HTK) was also provided. In this paper, detailed analyses of the datasets will be given. Experimental results obtained using these data will also be presented.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Voice I/O, Natural language, User-centered design* ; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Speech recognition and synthesis*

Keywords

mobile text input; multimodal interface; haptic voice recognition

1. INTRODUCTION

Haptic Voice Recognition (HVR) Grand Challenge 2012 is a research oriented competition designed to bring together researchers across multiple disciplines to work on Haptic Voice Recognition (HVR) [10], a novel multimodal text entry method for modern mobile devices. HVR combines both voice and touch inputs to achieve better efficiency and robustness. Since modern portable devices are now commonly equipped with both microphones and a touchscreen display, it will be interesting to explore possible ways of enhancing text entry on these devices by combining information obtained from these sensors. The purpose of this grand challenge is to define a set of common challenge tasks for researchers to work on in order to address the challenges faced and to bring the technology to the next frontier. Basic tools and setups are also provided to lower the entry barrier so that research teams can participate in this grand challenge without having to work on all aspects of the system.

The remainder of this paper is organized as follows. Section 2 gives a brief introduction to Haptic Voice Recognition (HVR). Section 3 describes the challenges to be addressed by the grand challenge. Section 4 presents the datasets and the data collection procedures. Section 5 gives a detailed account of the analyses performed on the datasets. Section 6 describe the HVR recipe provided for the challenge. Finally, Section 7 reports some experimental results on the datasets.

ICMI'12, October 22–26, 2012, Santa Monica, California, USA.

2. HAPTIC VOICE RECOGNITION

Haptic Voice Recognition (HVR) is a multimodal interface designed for efficient and robust text entry on modern portable devices. Nowadays, modern portable devices such as the smartphones and tablets are commonly equipped with microphone and touchscreen display. Typing using an on-screen keyboard is the most common way for users to enter text on these portable devices. In many situations, users can only type with one hand, while the other hand is holding the device. Furthermore, typing on smaller devices such as smartphones can be quite challenging. As a result, typing speed on portable devices is significantly slower compared to that on desktop and laptop computers with full-sized keyboard [4]. Voice input offers a hands-free solution for text entry. This is an attractive alternative to typing because voice input completely eliminates the need for typing. However, voice input relies on Automatic Speech Recognition (ASR) technology, which requires high computational resources and is susceptible to performance degradation due to acoustic interference. These are practical issues to be addressed since portable devices typically have limited computation and memory resources to accommodate state-of-the-art ASR system. Moreover, ASR systems have to cope with a wide range of acoustic conditions due to the mobility

of these portable devices. In addition, ASR systems often do not work as well for non-native speakers or speakers with a heavy accent. Users often find that voice input is like a black box that listens to the users voice and returns the recognition output without much flexibility for human intervention in case of errors. Certain applications will return multiple recognition hypotheses for the users to choose from. Any remaining unhandled errors are typically corrected manually. Instead of accepting human inputs after the recognition process, it may be more helpful to integrate additional human input into the voice recognition process. This is the basis motivated the development of Haptic Voice Recognition (HVR) [10].

Haptic Voice Recognition (HVR) is a multimodal interface designed to offer users the opportunity to add his or her '*magic touch*' in order to improve the accuracy, efficiency and robustness of voice input. HVR is designed for modern mobile devices equipped with an embedded microphone to capture speech signals and a touchscreen display to receive touch events. The HVR interface aims to combine both speech and touch modalities to enhance speech recognition. When using an HVR interface, users will input text verbally, at the same time provide additional cues in the form of *Partial Lexical Information* (PLI) [11] to guide the recognition search. PLIs are simplified lexical representation of words that should be easy to enter whilst speaking (*e.g.* the prefix and/or suffix letters). Preliminary simulated experiments conducted by [10] show that potential performance improvements both in terms of recognition speed and noise robustness can be achieved using the initial letters as PLIs. For example, to enter the text "*Henry will be in Boston next Friday*", the user will speak the sentence and enter the following letter sequence: 'H', 'W', 'B', 'I', 'B', 'N' and 'F'. These additional letter sequence is simple enough to be entered whilst speaking; and yet they provide crucial information that can significantly improve the efficiency and robustness of speech recognition. For instance, the number of letters entered can be used to constrain the number of words in the recognition output, thereby suppressing spurious insertion and deletion errors, which are commonly observed in noisy environment. Furthermore, the identity of the letters themselves can be used to guide the search process so that partial word sequences in the search graph that do not conform to the PLIs provided by the users can be pruned away.

3. THE HVR CHALLENGES

This section will present a detailed description of the HVR Grand Challenge. The main objective of the HVR Grand Challenge 2012 is to provide a common platform on which competitive research work can be performed easily by researchers across multiple disciplines. The HVR Grand Challenge is set to address two major challenges pertaining to HVR: 1) What kind of *haptic information* can be provided via touch input and how to provide them? and 2) What kind of inference models to be used and how to combine multiple inference models together?

In order to address the above two challenges, the grand challenge consists of two challenge sub-tasks, which correspond to one of the two components of the HVR system, as depicted in Figure 1. The front-end of an HVR system (HVR interface) captures the voice and touch inputs from the user using a microphone and a touchscreen display. The

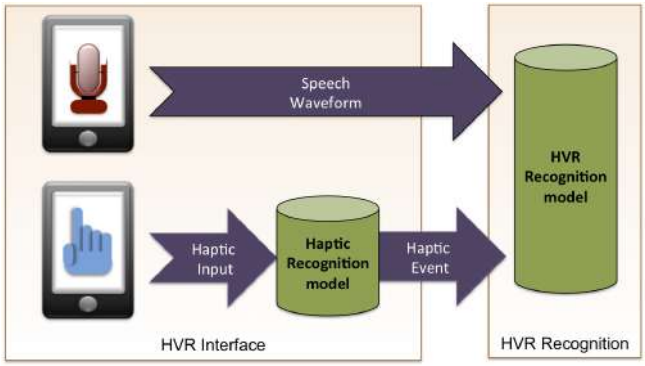


Figure 1: HVR System Architecture.

multiple streams of information captured by the front-end component are then processed by the back-end component (HVR recognition) to decipher the user intended input texts. The details of the two challenge subtasks will be described in the following.

3.1 T1 – The HVR Interface Challenge

The objective of this challenge subtask was to design innovative user interfaces for HVR. The core of this task was to design appropriate haptic events for HVR and methods for generating these events using touchscreen inputs. The complexity of the haptic event will affect the quality of the realized speech as well as the throughput using the overall HVR interface. For example, the haptic events may represent partial lexical information [11] of the words in the utterance, such as the initial and/or final letter of the words; and these letters may be generated by tapping on the appropriate keys on a soft keyboard or using more complex gesture recognition approaches. Through this challenge subtask, participants were given the freedom to propose innovative haptic events for HVR. For this challenge subtask, a list of text prompts were provided. Participants were asked to use their respective HVR interfaces to generate the corresponding speech data and haptic events. Systems were evaluated in terms of the word accuracy of the final text output from the overall HVR system. Participants in this challenge subtask may not need to build their own back-end recognition systems. A baseline HVR recognition system was provided to the participants to evaluate their HVR interfaces.

3.2 T2 – The HVR Recognition Challenge

This subtask was designed to challenge the research community to propose innovative recognition algorithms for HVR. HVR is essentially an extension to the conventional ASR, where haptic events are augmented as additional input. Participants were encouraged to discover new ways of making use of this additional information to improve the final recognition performance. Previously, haptic pruning was proposed in [10] to incorporate haptic inputs in order to constrain the decoding search space. A more generic probabilistic framework of integrating the haptic inputs based on Weighted Finite State Transducers (WFST) was introduced in [11]. Participants were invited to explore other possibilities, including but not limited to aspects such as acoustic and language model adaptation using the additional haptic events. For this subtask, participants were given a set

Entry Method	HVR Mode	Haptic Input
Method 1	Synchronous	Keyboard
Method 2	Synchronous	Keystroke
Method 3	Asynchronous	Keyboard
Method 4	Asynchronous	Keystroke

Table 2: Four different entry modes for HVR data collection.

of speech utterances along with the corresponding haptic inputs. In HVR Grand Challenge 2012, the initial letter sequences were generated using keyboard and keystroke inputs. Systems were evaluated based on the word accuracy of the final text output.

4. DATASETS

This section will describe the datasets used for the HVR Grand Challenge 2012. Three sets of data were made available to the challenge participants. A summary of these datasets in terms of the number of subjects, number of utterances and the amount of speech data is given in Table 1. The pilot dataset contains data collected from one subject. This subject has used the HVR interface for more than one year and can be regarded as an experienced user. The development and challenge datasets contains data collected from 4 and 15 subjects respectively. These subjects do not have prior experience using the interface. They were given the opportunity to practice with the HVR interface for several sentences before the data collection. These subjects were university students. Most of them were non-native English speakers.

4.1 Data Collection Procedures

The challenge datasets were collected using an HVR interface prototype implemented on iPad. The screenshots of the interface using the *keyboard* and *keystroke* input modes are depicted in Figures 2(a) and 2(b) respectively. Data collection was carried out with the HVR iPad interface operating in the *landscape* mode. For keyboard input, an onscreen soft keyboard with a standard QWERTY keyboard layout was used to enter the initial letters. The size of the keyboard is 352×1024, which is the same size as the standard English QWERTY keyboard provided by iOS. For keystroke input, subjects are required to use a predefined set of single-stroke handwriting gestures to enter the letters. These predefined gestures are given in Figure 3. Most of these letters can be represented by single-stroke gestures using the standard handwritten lowercase form, except for the letters 'F', 'I', 'L', 'T' and 'X', whose keystrokes are slightly modified to be single-stroke. Single-stroke handwriting input simplifies the recognition process since the letter boundaries are explicitly provided. Therefore, the system only needs to handle isolated handwritten letter recognition.

During data collection, each subject will enter a series of prompted texts using the HVR iPad interface. Each sentence was entered four times, each corresponds to a different HVR mode and a different haptic input method, as shown in Table 2. The synchronous HVR mode indicates that the subjects will enter the texts verbally, at the same time provide the corresponding initial letter sequence using either the keyboard or keystroke input method. On the other hand, for

(a) Keyboard Input Mode

(b) Keystoke Input Mode

Figure 2: Screenshots of HVR iPad interface used for data collection.

asynchronous HVR mode, subjects will read the prompted sentence first and then provide the initial letters afterwards.

For each text entry method, the speech utterances were recorded and stored as single channel 16 bit linear pulse code modulation (PCM) sampled at 16 kHz. For keyboard input, the HVR interface also captured the corresponding letter sequence as the subjects tap on the onscreen keyboard. The timestamps of the key presses relative to the start of the speech recording were also saved. For keystroke inputs, the HVR interface captured a series of 2-dimensional coordinates for each handwriting gesture. Likewise, the start times of the keystrokes relative to the start of the speech recording were also saved.

The data collected was conducted in a research laboratory where the recorded speech may be considered noise free. Noisy speech data were then artificially created by corrupting the clean speech with *additive* noise. The noise samples were collected from a school canteen where the primary noise type is babble noise. Three sets of noisy data were created at signal-to-noise ratios of 20dB, 15dB and 10dB.

Datasets	No. of Subjects	No. of Utterances		Amount of Speech (mins)	
		Train	Test	Train	Test
Pilot	1	164	20	12.0	1.5
Development	4	243	80	25.1	9.3
Challenge	15	977	180	94.3	20.8

Table 1: Number of subjects, number of utterances and amount of speech data in the pilot, development and challenge datasets.

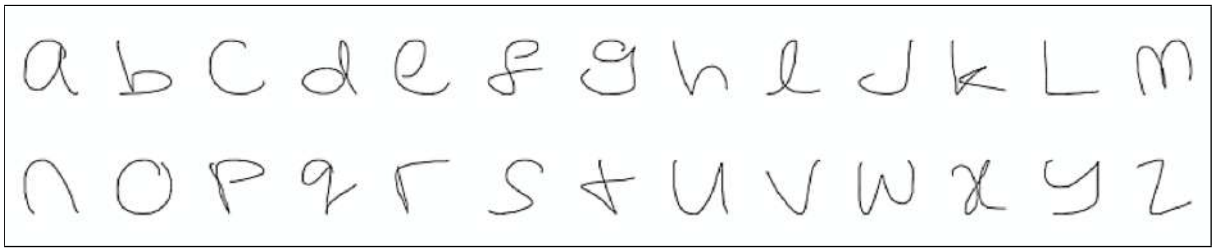


Figure 3: Single-stroke letter keystrokes used for data collection.

5. ANALYSES OF DATASETS

This section gives an account of the characteristics of the datasets in various aspects. First of all, the effects of HVR interface on the speech produced by the subjects were investigated. The durations of the speech and silence segments of the resulting speech collected using the synchronous and asynchronous modes were compared in Figure 4. Forced-alignment [13] was used to obtain the phone boundaries. The speech data produced by the subjects when using HVR in asynchronous mode were considered to be normal speech since their speech was not affected by any concurrent touch inputs. Therefore, the durations of the phones and silences for asynchronous mode were about the same for keyboard and keystroke inputs, as show in Figures 4(b), 4(d) and 4(f). Three types of silences were considered. A leading silence means the portion of silence at the beginning of each utterance. Likewise, a trailing silence denotes the portion of silence at the end of each utterance. Inter-word silences are the gaps in between successive words. These gaps are typically very small for fluent continuous speech.

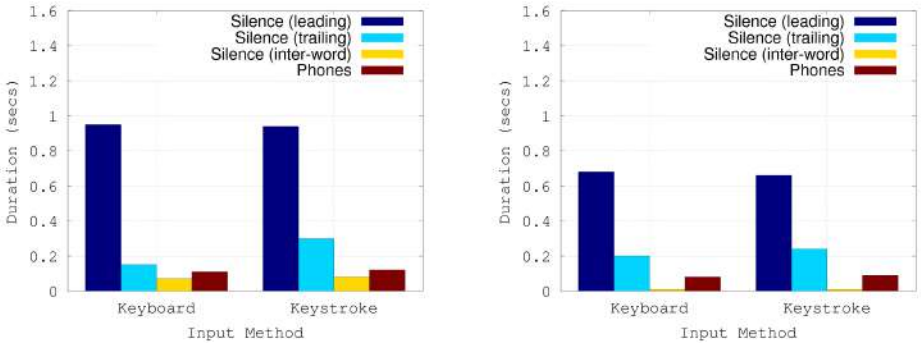
In general, the average durations of phones and various types of silences are longer for synchronous data compared to asynchronous data. The average duration of the leading silence for synchronous mode is about 1 second for all the datasets. This is consistently longer than the leading silence durations for asynchronous data, which indicates that there is a finite delay for the subjects to locate the key on the soft keyboard or determine the appropriate keystroke for the first letter of the first word of the sentence before he or she began to speak. There seems to be no difference in the leading silence durations between keyboard and keystroke inputs. On the other hand, the trailing silence for the keyboard and keystroke inputs are quite different for synchronous mode. For keyboard input, the trailing silence durations are almost the same for both synchronous and asynchronous cases. However, since the time taken to speak a word may be shorter than the time needed to complete a handwriting gesture for the corresponding initial letter, the trailing silences for synchronous keystroke mode was found to be more than 2 times longer than those for synchronous keyboard mode.

Similarly, the silence durations in between successive words were significantly longer for synchronous data. Beginners (subjects for development and challenge data) were found to spend on average 0.11s – 0.13s longer in between words to locate the right keys for synchronous keyboard input and 0.30s – 0.34s longer to complete the handwriting gestures. An experienced user, on the other hand, spent on average 0.06s and 0.07s longer in between words for keyboard and keystroke inputs. This shows that, with sufficient practice, potential speedup in HVR text entry can be achieved.

Besides, synchronous input also caused the average phone durations to be longer. The average phone duration for beginners increased by 0.02s –0.06s for synchronous keyboard input and 0.04s – 0.10s for synchronous keystroke input. On the other hand, the phones produced by an experienced user lengthened by 0.03s for both keyboard and keystroke inputs.

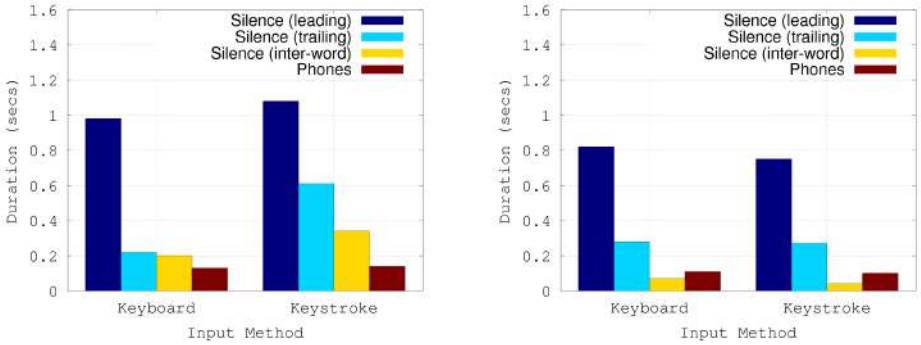
Next, the characteristics of the touch inputs were analyzed. Table 3 shows the average durations between successive haptic inputs. They were measured as the difference between the timestamps of the successive key presses or the start times of the successive handwriting gestures. The corresponding effective input speeds, measured in the number of words per minute (WPM), were also reported in the same table. For asynchronous mode, beginners' keyboard and keystroke input speeds were 69 – 79 WPM and 44 WPM respectively. An experienced user can achieve much higher input speeds, at 122 WPM and 95 WPM respectively. However, despite the additional cognitive loads, the effective haptic input speeds increased slightly for synchronous inputs. The input speeds for beginners increased to 73 – 87 WPM and 54 – 58 WPM for keyboard and keystroke inputs respectively. The keystroke input speed for an experienced user also increased to 102 WPM. This phenomenon may be due to the fact that the subjects subconsciously increase the haptic input speed to catch up with the faster speaking rate in synchronous mode.

Given the timestamps of the haptic inputs and the time boundaries of the phones obtained using forced-alignment, it will be interesting to analyze the synchrony of these two streams of inputs. Table 4 shows the average deviation of the haptic inputs from the start of the corresponding words.



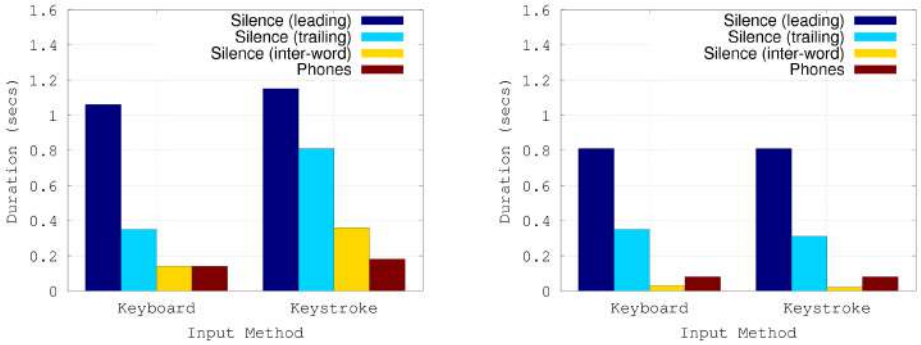
(a) Synchronous – Pilot

(b) Asynchronous – Pilot



(c) Synchronous – Development

(d) Asynchronous – Developement



(e) Synchronous – Challenge

(f) Asynchronous – Challenge

Figure 4: Durations between successive haptic events in the pilot, development and challenge datasets.

Datasets	Average Input Duration (sec)				Effective Input Speed (WPM)			
	Synchronous		Asynchronous		Synchronous		Asynchronous	
	Keyboard	Keystroke	Keyboard	Keystroke	Keyboard	Keystroke	Keyboard	Keystroke
Pilot	0.49	0.59	0.49	0.63	122	102	122	95
Development	0.82	1.04	0.87	1.36	73	58	69	44
Challenge	0.69	1.12	0.76	1.36	87	54	79	44

Table 3: Durations between successive haptic inputs and the effective input speed for the pilot, development and challenge datasets.

Datasets	Average Deviation (sec)	
	Keyboard	Keystroke
Pilot	0.10	0.36
Development	0.44	0.61
Challenge	0.22	0.62

Table 4: Deviation of haptic inputs from the start of the corresponding words for the pilot, development and challenge datasets.

Datasets	Input Method	Occurrence (%)		
		Before	Within	After
Pilot	Keyboard	4	96	0
	Keystroke	1	91	8
Development	Keyboard	8	80	12
	Keystroke	2	83	15
Challenge	Keyboard	11	84	5
	Keystroke	4	85	11

Table 5: Percentage of haptic inputs occurring before, within and after the corresponding words for the pilot, development and challenge datasets.

Only sentences whose length matches the number of corresponding haptic inputs were considered[1]. For beginners, key presses occurred about 0.22s – 0.44s after the start of the corresponding words; keystrokes happened 0.61s – 0.62s after the subjects started speaking the words. However, the deviations for an experienced user were much shorter: 0.10s and 0.36s for keyboard and keystroke inputs respectively. Sometimes, subjects may also enter the haptic inputs before they started speaking the word or after they have finished the word. Table 5 shows the percentage of haptic inputs occurring before, within and after the corresponding words. For beginners, between 80% – 85% of the haptic input occurrences fall within the corresponding words. About 2% – 11% and 5% – 15% of them happened before and after the words respectively. The haptic inputs for an experienced user were more precise. About 91% – 96% of them occurred within the words. Only 1% – 4% were before the words and 8% after the words.

6. HVR RECIPE

As part of this challenge, a simple recipe based on the Hidden Markov Model Toolkit (HTK) [13] was also provided. This recipe adopts an *offline* implementation of HVR where the recognition is performed after all the speech and haptic inputs are captured (*e.g.* at the end of an utterance). This allows the haptic inputs to be incorporated as constraints to restrict the decoding network so that the standard speech recognition algorithm can be used without modification. This implementation uses *regular expressions* to represent the Partial Lexical Information (PLI) for each word. For example, for the sentence "*My name is Peter*", the initial letter sequence 'M', 'N', 'I' and 'P' is represented as

```
^M, ^N, ^I, ^P
```

Likewise, the final letter sequence 'Y', 'E', 'S' and 'R' is represented as

```
Y$, E$, S$, R$
```

Combining the above initial and final letter information yields the following PLI representation:

```
^M.*Y$, ^N.*E$, ^I.*S$, ^P.*R$
```

Given the PLI information, a *lexically constrained* decoding network will be constructed in the form of a confusion network (see Figure 5). Each PLI is expanded into a set of word alternatives by matching its regular expression against all the words in the vocabulary. For example, the regular expression `^M.*Y$` will expand to words including *MACY*, *MANY*, *MAY*, *MY* and so on. This is a very simple implementation of HVR which does not support a tight integration of haptic inputs into the decoding process in an online manner. It also does not support the incorporation of language model scores which are typically used in speech recognition. Furthermore, this implementation also assumes that the PLI information provided are accurate since any haptic input error will lead to the correct words being excluded from the resulting lexically constrained decoding network. A more advanced probabilistic integration framework based on Weighted Finite State Transducer (WFST) has been proposed in [11], which is able to incorporate language model scores and handle uncertainties in haptic inputs.

7. EXPERIMENTAL RESULTS

This section presents the experimental results using the HVR Grand Challenge 2012 datasets described in Section 4. This section is divided into two parts. The first part describes the inference models for different haptic input methods and presents the letter recognition performance of these inference models. The second part describes the HVR recognition systems and their performances.

7.1 Haptic Input Performance

The datasets provided for the HVR Grand Challenge 2012 comprise the speech recording as well as the corresponding initial letter sequences for the words in the utterances. These

[1]There were a small number of sentences where subjects entered more or fewer letters than necessary by mistake.

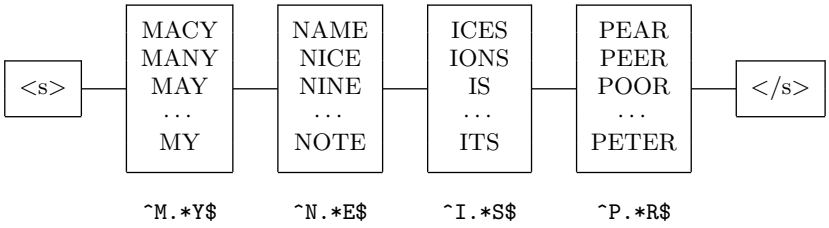


Figure 5: An example lexically-constrained decoding network based on the given initial and final letter Partial Lexical Information (PLI) for the sentence "*My name is Peter*". <s> and </s> denote the start and end of the sentence respectively.

Input Method	HVR Mode	LER (%)		
		Pilot	Dev.	Challenge
Keyboard	Sync	0.0	2.3	0.7
	Async	1.4	0.7	1.3
Keystroke	Sync	0.0	6.8	10.7
	Async	0.0	9.5	11.6

Table 6: Letter error rate performance of haptic inputs for the pilot, development and challenge datasets.

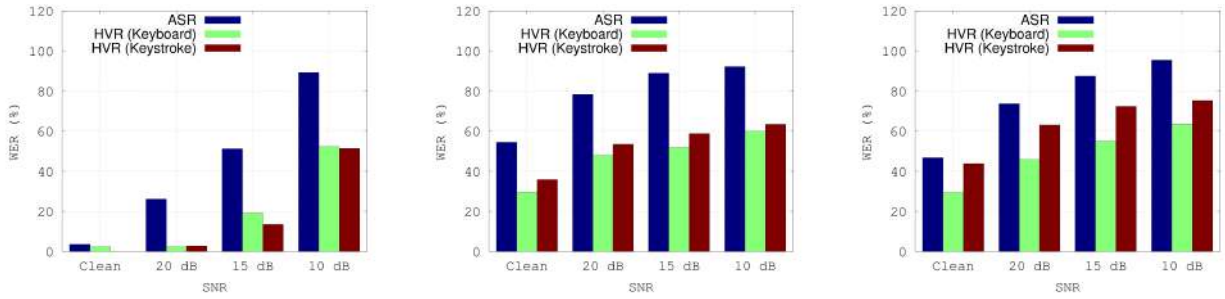
initial letters were entered by users either using an onscreen QWERTY keyboard or handwriting gestures (see Section 4.1 for more details on the data collection procedures). For the keystroke input, a 3-state left-to-right Hidden Markov Model (HMM) [9] was used to model the handwriting gesture for each letter. The emission probability of each state was represented by a Gaussian distribution with a full covariance matrix. The input features were 6-dimensional vectors given by the two-dimensional normalized coordinates of the touch points together with the first and second order differential parameters representing the instantaneous gradient and curvature of the keystroke. These differential parameters were computed using HTK [13], similar to the way the dynamic parameters were generated for speech recognition. Table 6 shows the Letter Error Rate (LER) performance of the haptic inputs provided by the users. For keyboard input, the LER indicates the error rate of the user tapping on the incorrect keys. Likewise, the LER indicates the performance of the underlying handwriting recognition system for keystroke input. One of the difficulties faced by the beginners is getting accustomed to the handwriting gestures shown in Figure 3 for keystroke input. This results in much higher LERs compared to keyboard inputs. Surprisingly, the LERs were lower for synchronous mode despite the additional cognitive loads involved. The LERs for keyboard inputs were 0.7% – 2.3% for synchronous input and 0.7% – 1.3% for asynchronous input. However, subjects in the development set made more errors for synchronous input while those in the challenge set made more errors for the asynchronous mode. So, one can only say that the error patterns are user specific. An experienced user, however, was able to provide a more consistent haptic inputs. There were no errors in inferring the letters in all cases, except for asynchronous keyboard input. They were substitution and deletion errors indicating that the user may have subconsciously replaced or skipped certain words as the sentence was being recalled after it was first spoken.

7.2 HVR Recognition Performance

Finally, we report the performance of the baseline HVR system. The baseline system was provided together with the HVR Grand Challenge 2012 datasets. In this baseline system, triphone acoustic models were represented by 3-state left-to-right Hidden Markov Model (HMM) [9]. Decision tree state clustering [14] was used to control the model complexity such that the final system comprised about 3000 distinct states. The emission probability of each HMM state is represented by a Gaussian distribution. Although more advanced configuration are used in state-of-the-art large vocabulary continuous speech recognition (LVCSR) [12] systems (e.g. Gaussian Mixture Model (GMM) state emission probability [7] and n-gram statistical language model [3]), a much simpler baseline system was chosen for HVR so that it is more practical for mobile devices with limited computation and memory resources. Mel Frequency Cepstral Coefficient (MFCC) [5] features were used for acoustic model training. 12 static coefficients together with the `C0` energy term and the first two differential parameters were used to form a 39 dimensional acoustic feature vector. Maximum likelihood Baum-Welch training [2] was used to estimate the HMM parameters. Maximum Likelihood Linear Regression (MLLR) [8] was used to adapt the Gaussian mean and variance vectors to specific users and noise conditions[2].

Figure 6 summarizes the Word Error Rate (WER) performances of *synchronous* HVR in various noise conditions for the pilot, development and challenge datasets. The ASR performances were obtained using the speech data collected in the *asynchronous* mode. In general, one observes a consistent improvement of HVR (either using keyboard or keystroke inputs) over ASR across different noise conditions. This shows the effectiveness of using additional haptic inputs to enhance the robustness of voice input in noisy environment. Further, the WER results on the pilot dataset were much better than those on the other datasets. This is because the subject in the pilot dataset has a good English proficiency while the subjects in the development and challenge datasets were mostly non-native English speakers. In general, HVR using keyboard input achieved better WER performance compared to using keystroke input. This is expected since the letter recognition error for keystroke input is much higher than keyboard input (see Table 6). Further-

[2]This work adopts MLLR as a simple approach to adapt the acoustic models to different noise conditions since it is readily supported by HTK. More advanced model-based noise compensation techniques, such as Parallel Model Combination (PMC) [6] and Vector Taylor Series (VTS) [1] can also be used.



(a) Pilot (b) Development (c) Challenge

Figure 6: Word error rate performance of *synchronous* HVR for the pilot, development and challenge datasets.

more, it was also observed that the WER performance of HVR still degrades significantly as the signal-to-noise ratio (SNR) decreases. This shows that MLLR is not very effective for noise compensation. However, it was found in [11] that the combination of VTS [1] noise compensation and HVR can greatly enhance the noise robustness.

8. CONCLUSIONS

This paper has presented a detailed description of the Haptic Voice recognition (HVR) Grand Challenge 2012 and the datasets collected for this challenge. Various analyses conducted on the datasets showed that synchronous input has the effect of increasing the durations of the phones and gaps in between words. The effect is smaller for a more experienced user. Keyboard inputs were found to be much quicker to input and had much lower inference error compared to keystroke inputs. However, since this study involved only one experienced user, more detailed studies are needed to properly understand the full potential of HVR.

9. REFERENCES

[1] A. Acero, L. Deng, T. Kristjansson, and J. Zhang. HMM adaptation using vector Taylor series for noisy speech recognition. In *Proc. of ICSLP*, volume 3, pages 869–872, 2000.

[2] L. E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bull. Amer. Math. Soc.*, 73:360–363, 1967.

[3] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, ACL '96, pages 310–318, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

[4] E. Clarkson, J. Clawson, K. Lyons, and T. Starner. An empirical study of typing rates on mini-qwerty keyboards. In *CHI '05 extended abstracts on Human factors in computing systems*, CHI EA '05, pages 1288–1291, New York, NY, USA, 2005. ACM.

[5] S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustic, Speech and Signal Processing*, 28(4):357–366, 1980.

[6] M. Gales, S. Young, and S. J. Young. Robust continuous speech recognition using parallel model combination. *IEEE Transactions on Speech and Audio Processing*, 4:352–359, 1996.

[7] X. Huang, A. Acero, H.-W. Hon, and R. Reddy. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall PTR, 1st edition, 2001.

[8] C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer speech and language*, 9(2):171, 1995.

[9] L. A. Rabiner. A tutorial on hidden Markov models and selective applications in speech recognition. In *Proc. of the IEEE*, volume 77, pages 257–286, February 1989.

[10] K. C. Sim. Haptic voice recognition: Augmenting speech modality with touch events for efficient speech recognition. In *Proc. SLT Workshop*, 2010.

[11] K. C. Sim. Probabilistic integration of partial lexical information for noise robust haptic voice recognition. In *Proceedings of the 50th annual meeting on Association for Computational Linguistics*, ACL '12. Association for Computational Linguistics, 2012.

[12] S. J. Young. Large vocabulary continuous speech recognition: A review. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 3–28, Snowbird, Utah, December 1995.

[13] S. J. Young et al. *The HTK Book (for HTK version 3.4)*. Cambridge University, December 2006.

[14] S. J. Young, J. J. Odell, and P. C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings ARPA Workshop on Human Language Technology*, pages 307–312, 1994.