Design and Implementation of the Note-taking Style Haptic Voice Recognition for Mobile Devices

Seungwhan Moon
Franklin W. Olin College of Engineering
1000 Olin Way
Needham, MA, U.S.A.
me@shanemoon.com

Khe Chai Sim
National University of Singapore
Computing 1, 13 Computing Drive
Singapore, Singapore 117417
simkc@nus.edu.sg

ABSTRACT

This research proposes the "note-taking style" Haptic Voice Recognition (HVR) technology which incorporates speech and touch sensory inputs in a note-like form to enhance the performance of speech recognition. A *note* is taken from a user via two different haptic input methods - handwriting and a keyboard. A note consists of some of the keywords in the given utterance, either partially spelled or fully spelled. In order to facilitate fast input, the interface allows a shorthand writing system such as Gregg Shorthand. Using this *haptic note sequence* as an additional knowledge source, the algorithm re-ranks the n-best list generated by a speech engine. The simulation and experimental results show that the proposed HVR method improves the Word Error Rate (WER) and Keyword Error Rate (KER) performance in comparison to an Automatic Speech Recognition (ASR) system. Although it generates an inevitable increase in speech duration due to disfluency and occasional mistakes in haptic input, the compensation is shown to be less than conventional HVR methods. As such, this new note-taking style HVR interaction has the potential to be both natural and effective in increasing the recognition performance by choosing the most likely utterance among multiple hypotheses. This paper discusses the algorithm for the proposed system, the results from the simulation and the experiments, and the possible applications of this new technology such as aiding spoken document retrieval with haptic notes.

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces—*Haptic I/O*; I.2.7 Artificial Intelligence: Natural Language Processing—*Speech recognition and synthesis*

Author Keywords

Spoken document retrieval; haptic voice recognition; note-taking style; multi-modal Interface

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

INTRODUCTION

As an increasing number of mobile devices support speech recognition technology as a native application, automatic speech recognition (ASR) technology is becoming a popular method of interaction among smartphone users. ASR technology essentially enables verbal communication between humans and mobile devices, thus providing the most natural interaction experience to users. However, the recognition performance of ASR often does not meet the accuracy to be a primary method of interaction for mobile devices, especially in a noisy environment. As a result, most of the users still choose to use the touch interface as a main input medium.

Haptic Voice Recognition (HVR) is a multi-modal interface that combines speech data with touch sensory inputs via a mobile device to improve voice recognition performance. HVR uses haptic events such as taps or strokes as additional cues, thus filtering away the less probable sequences of words given a speech utterance. Examples of haptic inputs include the Boundary of Sentences (BoS), Boundary of Words (BoW), First Letter of Words (FLoW), etc. Previous research in this field has significantly improved the performance of speech dictation, as shown in the decreased Word Error Rate (WER) and increased Runtime Factor (xRT) performance of HVR, compared to those of conventional Automatic Speech Recognition (ASR). The prior studies however, led to an increased duration of speech and the atypical fluency [9, 10].

In this paper, a different method of combining speech and haptic input, named the *note-taking style* HVR, is examined. In the note-taking style HVR, haptic inputs in the form of a note are used as additional cues to aid in choosing the most likely candidate for a given utterance. In order to maximize the freedom of usability, the note-taking style HVR allows shorthand sequences such as shorthand letters and partially spelled words as haptic inputs. Shorthand writing is a difficult skill to acquire for beginners, but the experts of shorthand systems such as Gregg Shorthand [3], Pitman Shorthand [7], or Teeline Shorthand [2] are known to be able to write five to ten times faster than the average handwriting for the Roman alphabet [3]. In this research, the Gregg shorthand alphabet (Figure 1) is used because of its popularity among journalists and other professionals. This paper defines the rules and algorithms for the

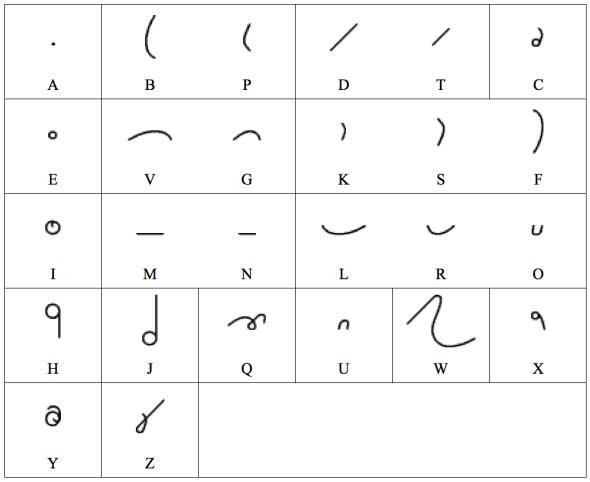


Figure 1. A list of Gregg shorthand letters used in the Note-taking style HVR grouped by their shapes

note-taking style HVR, and examines its feasibility by conducting simulations and preliminary user studies. Finally, the paper examines the possible applications of the suggested algorithm as well as the future works that may be conducted.

NOTE-TAKING STYLE HAPTIC VOICE RECOGNITION

Note-taking is a natural way of recording information that accompanies verbal communication, which aids in the recognition and retention of both human thoughts and communication. A note is typically taken synchronously with speech, and is usually composed of a series of keywords or shorthand symbols that represent the context of the given speech. As note-taking is a common process widely performed in recording lectures or daily conversations, it is empirically known as a simple task that does not require much cognitive attention.

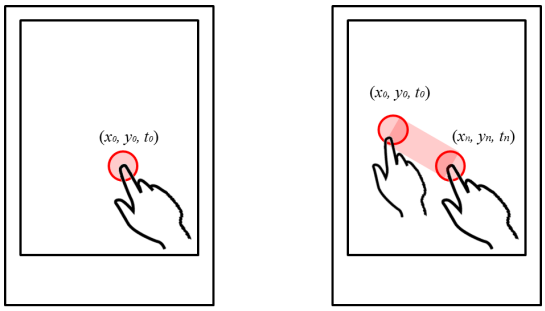
As such, we design the note-taking style HVR such that it resembles the process of natural note-taking on paper. In the note-taking style HVR, a *note* will be taken synchronously with voice input via touch screen. The optimal interface would thus include both a hand-held touch device and a touch-sensitive stylus pen. A note consists of a sequence of keywords associated with the given speech utterance. Each keyword can be either fully or partially spelled in letters, spelled as common abbreviations, or written in shorthand symbols. Partially spelled words or shorthand symbols add ambiguity, which can be considered as a trade-off for faster input. In this paper, such Partial Lexicon Information (PLI), or a series of partially or fully spelled words that are given by the haptic events is referred to as a *haptic note sequence*. In order to maximize the freedom of usability and the performance of the system , the note-taking style HVR imposes the following rules:

1. An element in a haptic note sequence refers to a partially or fully spelled word in the decoded word sequence.
2. The number and the order of keywords in a haptic note sequence do not need to match those of words in the actual word sequence.
3. The exact time at which a haptic event occurs is ignored.

The first rule ensures that an element of a haptic note sequence is relevant to the actual sequence in terms of its format as well as its context or semantics. The second rule allows users freedom of choosing the context and semantics of the haptic note sequence. The third rule ensures that users will not be concerned about synchronizing the speech and the haptic input. These rules are intended to let users interact in the most natural way possible. Using this haptic information as additional knowledge sources, the algorithm re-ranks the likelihood of the n-best utterances returned by a speech recognition engine. The following sections describe the essential components and the algorithm of the note-taking style HVR in detail.

Haptic Input Methods

In order to aid the user experience in providing the haptic input, two types of haptic events are allowed: taps and strokes (Figure 2). Tap events are triggered when a user touches the screen with his or her finger, and are defined by the time and location (x, y coordinate of the screen) at which the touch event occurred. Stroke events, on the other hand, are defined by the sequence of times and locations of touch events from the initial touch until the removal of the finger [9]. Therefore, handwritten letters or shorthand symbols can be trained as an ordered sequence of coordinates using Hidden Markov Models (HMM) [4, 5, 12]. For tap events, letters and symbols can be input via a virtual keyboard that consists of a grid layout of letters and symbols.



(a) Tap Events (b) Stroke Events

Figure 2. Illustration for a tap event and a stroke event

The series of input letters is split into groups by a white space, each of which is considered as a fully or partially spelled word in a haptic note sequence. For stroke events, words are separated using the simple white space detection algorithm by relative proximity of letters. For tap events, words are separated by an explicit input of a white space.

Gregg Shorthand Handwriting Recognition for HVR

In order to facilitate fast input, a sequence of words written in Gregg shorthand symbols is accepted as a haptic input as well. Gregg shorthand is completely based on elliptical figures and lines that bisect them, allowing the fastest input possible via handwriting [3]. However, handwriting recognition for Gregg shorthand can be challenging, because some of the letters that have phonetic similarities, such as 'b' and 'p', 'r' and 'l', also have similar shapes (Figure 1). For effective handwriting recognition for HVR, such letters are classified into groups, thus enhancing the overall recognition accuracy. The haptic note sequence written in Gregg symbols thus has to incorporate all the possible permutations of spellings of words as alternatives, which adds ambiguity to the partial lexicon information.

The original Gregg shorthand system has a set of rules that facilitates fast and effective input. For example, the traditional Gregg shorthand system encourages writers to spell words phonetically, which makes writing faster, but significantly harder to retrieve the written document. In order to avoid unnecessary ambiguity added to the haptic note sequence, the Note-taking style HVR specifies users to follow the conventional spelling of words. This paper examines the efficacy and validity of the 'borrowed' Gregg shorthand system optimized for HVR.

Algorithm Design

Sim [10] defines the haptic voice recognition problem as finding the joint optimal solution for the word sequence, $\hat{\mathcal{W}}$ and the PLI sequence, $\hat{\mathcal{L}}$, given the sequence of observed acoustic features, $\mathcal{O}$, and the sequence of haptic features, $\mathcal{H}$. This definition can be expressed as the following equation:

$$(\hat{\mathcal{W}}, \hat{\mathcal{L}}) = \arg\max_{\mathcal{W},\mathcal{L}} P(\mathcal{W}, \mathcal{L}|\mathcal{O}, \mathcal{H}) \quad (1)$$

By expanding (1), Sim [10] identifies the four knowledge sources that incorporate HVR, which are the acoustic model score, the language model score, the haptic model score, and the PLI model score. These knowledge sources can be represented as Weighted Finite State Transducers (WFSTs) [6], and the composition operator ($\circ$) can integrate them into a single WFST, of which the shortest path refers to the optimal solution of (1). In the case of the note-taking style HVR, the merged WFST can be expressed as:

$$\bar{\mathcal{F}} = \hat{\mathcal{L}} \circ \bar{\mathcal{P}} \circ \hat{\mathcal{H}} \quad (2)$$

where $\hat{\mathcal{L}}$, $\bar{\mathcal{P}}$, $\hat{\mathcal{H}}$ denote the WFST representation of the lattice that consists of multiple hypotheses of word sequences decoded by a speech engine, the PLI model, and the lattice of the permutations of the haptic note sequence. Essentially, $\bar{\mathcal{P}}$ serves as the merger that defines the alignment between $\hat{\mathcal{L}}$ and $\hat{\mathcal{H}}$, by assigning non-zero values for *substitution* and *insertion cost*. The distance between a haptic note sequence and a decoded sequence can be determined by computing the edit penalties of $\bar{\mathcal{P}}$. Since the actual order of keywords in a haptic note is ambiguous, all the possible permutations of N words ($N!$) are considered. Finally, the combined WFST ($\bar{\mathcal{F}}$) can be sorted by the computed distances, the process of which is referred to as *lattice rescoring* [10]. Essentially, the whole process can be viewed as re-ranking the n-best list by measuring how probable each decoded word sequence is given the haptic note sequence.

In this research, OpenFST[1], an open source library for performing FST operations [1], was used in order to implement the described process. The n-best list, the PLI model, and the haptic notes were compiled into the WFST representations using `fstcompile` method, which were then composed into a single WFST by executing the `fstcompose` method. Finally, `fstshortestpath` was used to get the n-shortest paths of $\bar{\mathcal{F}}$.

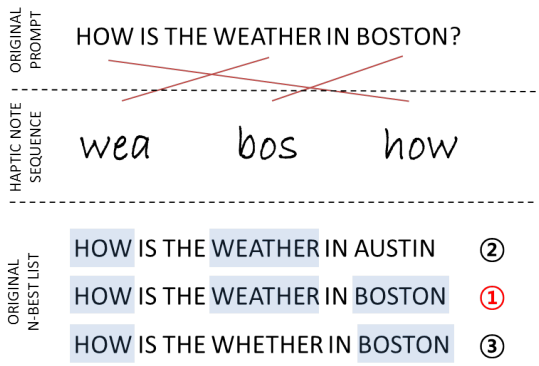


Figure 3. An example of the re-ranked word sequences, given a speech utterance "How is the weather in Boston" and a partially spelled haptic note sequence 'wea bos how.'

Figure 3 shows an example of the re-ranked n-best list given a haptic note sequence. Note that the best choice after re-ranking has the most number of words matched with the given haptic note sequence, thus having the shortest distance. Note also that if multiple hypotheses have the same distances, they are secondly sorted according to their ranks from the original n-best list.

Note that this method is different from some of the haptic methods which aid in the recognition process by pruning the search space during the decoding process, which is referred to as *haptic pruning*. Haptic events such as First Letter of Words (FLoW) via *integrated decoding*, for example, can effectively prune the search space under the rule that dth haptic input matches the dth word of the decoded sequence of words. If the letter specified by the dth haptic input does not match the first letter of dth word in the Word Link Record (WLR), such word can be pruned away, thereby improving the efficiency

[1] http://www.openfst.org

of the recognition process. When a WLR is haptically pruned, it improves the runtime factor performance as well as the recognition accuracy [9, 10].

On the other hand, the process of lattice rescoring is independent of a speech engine, and thus it cannot prune away the search space during the decoding process. This is because lattice rescoring is an additional re-validation of the given n-best list. As such, it does not improve the runtime factor of the speech engine itself. It also indicates that the performance of the note-taking style HVR is largely dependent on the quality of the ASR speech engine that it is based on.

EXPERIMENTAL RESULTS

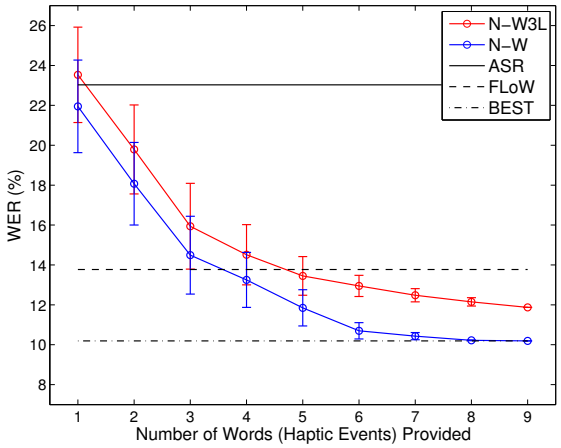
In order to investigate the feasibility of the Note-taking Style HVR, the following simulation and experiments were conducted. For each experiment, the Google Speech Recognition API was used to get a 20-best list for each speech input.

Simulation

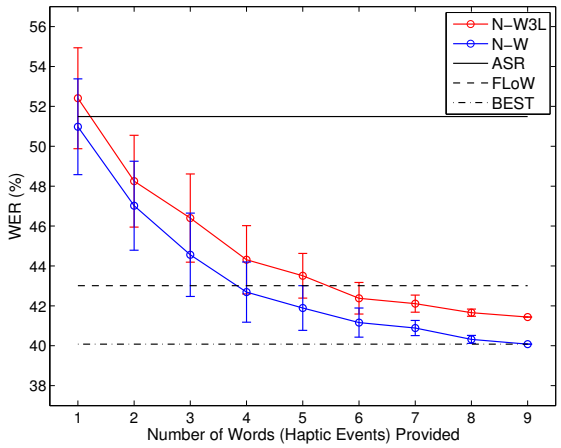
The following simulation was performed in order to investigate the theoretical performance improvement over different numbers of haptic input. For each prompt, artificial haptic events were generated by randomly choosing N words from the prompt sentence. In reality, a user would spot and choose the keywords in the sentence, and write or type those keywords by using one of the haptic input interfaces described earlier. This simulated haptic note sequence is then used to find the distance from the n-best list generated by the speech engine.

For each iteration, N words were randomly chosen from each sentence. Two cases were considered for each chosen word: when only the first three letters of the word were provided (partially spelled), and when it was fully spelled. The simulation was repeated from $N = 1$ to $N = 9$, each repeated 100 times, under two different Sound Noise Ratios (SNR) - clean and 15dB. The 15dB sound data were obtained by artificially corrupting the original speech data with the *babble* noise from the NOISEX-92 database [11]. In this simulation, 72 sentences from the Wall Street Journal Database [8] were used. In order to compare the performance with that of other HVR methods, artificial haptic events for FLoW were also generated and tested. Lastly, the worst and the best WER obtainable from the 20-best list were measured as a reference for the quality of the speech engine. Figure 4(a) and 4(b) show the simulation results.

As seen in Figure 4(a) and 4(b), the performance of the N-W method reaches its best possible WER at $N = 9$. This shows that the upper boundary for improvement in WER is bounded by the quality of the speech engine that the note-taking style HVR is based on. Note that the differential improvement of WER decreases as N increases. This indicates that the PLI information becomes redundant as the number of the given keywords increases. A user can thus determine the number of keywords to provide, with the trade-off in time duration in mind. Note



(a) Simulation results (clean)



(b) Simulation results (15dB)

Figure 4. Simulation results (a) when performed without any additional noise and (b) when performed with artificial noise at SNR = 15dB. x-axis denotes the number of randomly chosen keywords (N), whereas y-axis denotes the word error rate (WER). The red and the blue lines refer to the Note-taking-style HVR performance with the first 3 letters of N randomly chosen words (N-W3L), and the Note-taking-style HVR performance with N fully-spelled words (N-W). The error bars indicate the standard deviations of the 100 iterations. The ASR performance, FLoW HVR performance, and the best obtainable performance were represented as the black solid, the dashed, and the dash-dot lines, respectively.

also that the standard deviations of WER during the 100 simulations on the Note-taking Style HVR are significant. This indicates that the performance of the Note-taking Style HVR greatly depends on the words that a user chooses as their keywords. A user might want to give the words that are often mis-recognized by the speech engine (due to their phonetical or grammatical nature) as keywords, in order to maximize the efficiency of the system. However, this information is not given

to users most of the time. Users would rather naturally choose the words that are more important in the context of the sentence. Hence, the results of the actual experiments with more realistic haptic input are presented in the next section.

Preliminary User Studies

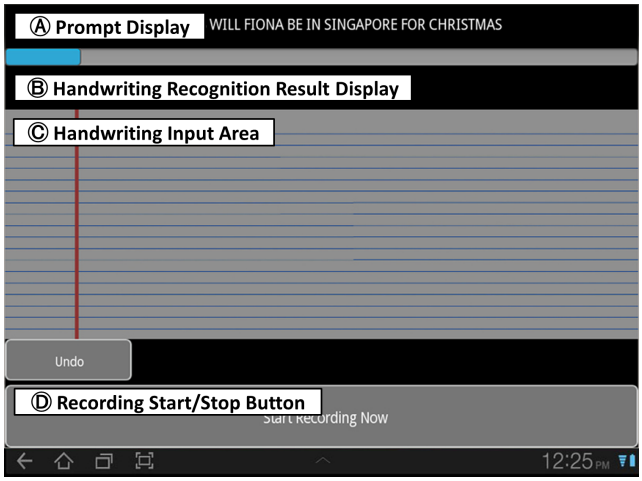


Figure 5. A prototype interface for handwriting data collection. For tap event data collection, a virtual QWRETY keyboard replaced the handwriting area seen in section C.

In order to perform preliminary user studies, a prototype interface (Figure 5) was developed for data collection. The data acquisition was conducted under the following protocol: a user provides both speech and haptic note sequence data, given a written prompt and a set of keywords. The experiment is repeated for four different haptic input methods - First Letter of Word (FLoW), haptic note sequence of the first 3 letters of the 3 chosen words (3W3L) inputted by longhand (normal) handwriting, Gregg shorthand handwriting, and touch keyboard. For Gregg shorthand writing, the interface displayed the Gregg symbol representation of the keywords so that a user without expertise in shorthand writing could trace. The 3W3L was performed in this experiment because it showed a good improvement in WER while having the minimal increase in the duration of speech. 72 sentences were randomly selected from WSJ Database, and were recorded by a single user for each haptic input method. The results were compared with the data recorded without any haptic events other than the Boundary of Sentences (BoS) which only defines the start and the end of each prompt. The best and worst obtainable WER from the given n-best list was also reported, in order to indicate the quality of the speech engine performance.

Table 1 shows the speed of input in Words Per Minute (WPM), WER, and Keyword Error Rate (KER) for each haptic method. WPM is defined as the ratio of the number of words to the total duration of input measured from the start of recording to the end of the last haptic input. KER is defined as the ratio of the number of keywords appearing in the sentence to the total number of keywords provided, where keyword is the word that a user chooses to provide a haptic event for. All the haptic methods showed similar ASR performance, with similar best and worst obtainable WER. For all the haptic input methods, the algorithm showed improvement in both WER and KER when compared to ASR, although it brought disfluency in speech. Among the haptic methods tried in this experiment, FLoW showed the greatest absolute improvement in WER performance. Although the 3W3L methods showed a smaller improvement in WER, they generally showed a greater improvement in KER than FLoW. This outcome indicates that FLoW chooses the candidate that increases the overall WER, whereas the N-Words haptic methods choose the one that matches the most number of keywords, thus maximizing the KER improvement. As for the speed of input, 3W3L by shorthand handwriting showed the smallest decrease (14.3%) from BoS , whereas FLoW resulted in the largest decrease (24.2%). However, Shorthand 3W3L showed less improvement in WER and KER performance than other 3W3L methods. This decrease is due to the added ambiguity of the shorthand haptic note sequence because of its classified letters.

CONCLUSION

The note-taking style of HVR that resembles the process of natural note-taking on paper was examined throughout this study. The simulations and preliminary experiments showed notable improvement in both the Word Error Rate (WER) and the Keyword Error Rate (KER) over ASR. Specifically, the note-taking style HVR showed greater improvement in KER than other HVR methods such as FLoW. When a user writes a note, it is assumed that he or she chooses the keywords of the given speech that best reflect the context and semantics of the speech. Thus, this improvement in KER can enhance the user experience with the speech recognition system, because users expect to see the keywords they choose become corrected in the final output. Although the note-taking style HVR did cause an increased duration of speech compared to ASR, the use of partially spelled words and Gregg shorthand symbols minimized this side-effect. Because the WPM of a haptic method may vary depending on the users' familiarity with the input medium, users can choose the medium that enables the fastest input for them. However, it was shown that users' choice of keywords can significantly affect the performance, as seen in the large standard deviation of WER in the HVR simulation. Therefore, users would have an increased chance of experiencing a better performance if they choose the words that are frequently mis-recognized for their phonetic features or grammatical properties, and vice versa.

It is worthwhile to note that the proposed algorithm can be applied to any speech recognition engine as long as it returns the n-best list for a given speech. As such, the proposed HVR method can easily be augmented to the existing Large Vocabulary ASR services such as Google

Table 1. Result of the Experiment

Haptic Method	Words Per Minute (WPM)	WER (%)				KER (%)		Improvement (%)	
		ASR	HVR	(Best)	(Worst)	ASR	HVR	WER	KER
BoS	110.7	24.9	-	-	-	22.6	-	-	-
Shorthand (3W3L)	94.9	24.8	18.3	12.7	47.5	21.2	12.2	6.6	9.0
Longhand (3W3L)	86.5	23.1	15.8	11.9	46.8	22.2	12.9	7.3	9.3
Keyboard (3W3L)	85.2	25.9	18.6	12.6	51.7	24.4	14.9	7.3	9.4
FLoW	83.9	23.6	14.4	12.0	49.9	23.2	17.5	9.2	5.7

Speech API for Android devices[2], AT&T's Watson API[3], or wami[4]. This gives developers the opportunity to easily implement applications that make use of this technology.

The preliminary user studies discussed the case where a user provides both speech and haptic note sequence data synchronously, given a written prompt. As it is shown in the previous research as well, the multi-tasking between speaking and writing is an unusual and thus challenging psychological task. Future work will investigate another application of the same technology, where a user listens to a given speech, and provides an appropriate haptic note sequence synchronously. This task is more natural for a user, of which an example can be found in everyday tasks such as listening to a lecture and taking notes. Such application can aid in spoken document retrieval when a natural note input is given.

REFERENCES

1. C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri. Openfst: A general and efficient weighted finite-state transducer libraryh. *Lecture Notes in Computer Science*, 4783(11):11–23, 2007.
2. H. Butler. *Teeline Shorthand*. Butterworth Heinemann, 1991.
3. J. R. Gregg. *The Basic Principles of Gregg Shorthand*. New York: Gregg Pub, 1923.
4. S. Gunter and H. Bunke. Hmm-based handwritten word recognition: on the optimization of the number of states, training iterations and gaussian components. *Pattern Recognition*, 37:2069–2079, 2004.
5. J. Hu, S. G. Lim, and M. K. Brown. Writer independent on-line handwriting recognition using an hmm approach. *Pattern Recognition*, 33(1):133–147, 2000.
6. M. Mohri, F. Pereira, and M. Riley. Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, 16(1):69–88, 2002.
7. G. A. Reid, E. J. Thompson, and M. Angus. *Pitman Shorterhand*. New York: Pitman Pub, 1972.
8. T. Robinson, J. Fransen, D. Pye, J. Foote, S. Renals, P. Woodland, and S. Young. *WSJCAM0 Cambridge Read News*. Linguistic Data Consortium, Philadelphia, 1995.
9. K. C. Sim. Haptic voice recognition: Augmenting speech modality with touch events for efficient speech recognition. *IEEE Spoken Language Technology Workshop (SLT)*, pages 73–78, 2010.
10. K. C. Sim. Probabilistic integration of partial lexical information for noise robust haptic voice recognition. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 31–39, July 2012.
11. A. Varga and H. J. Steeneken. Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Communication*, 12(3):247–251, 1993.
12. S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book (for HTK version 3.4)*. Cambridge University, December 2006.

[2]http://developer.android.com/reference/
[3]http://www.research.att.com/projects/WATSON/index.html
[4]http://code.google.com/p/wami/