AUTOMATIC-SUBTITLING: COMPARISON ON THE PERFORMANCE OF FORCED ALIGNMENT AND AUTOMATIC SPEECH RECOGNITION

Mino Lee Sasse[1], *Stefan Schaffer*[2], *Aaron Ruß*[3]

[1]*DFKI/Uni-Potsdam,* [2]*DFKI,* [3]*DFKI*
misa02@dfki.de

Abstract: This work is focusing on the automatic generation of subtitles using different tools that can be categorized as Forced Aligners (FAs) or Automatic Speech Recognizers (ASRs). A comparison of the performance of FA and ASR for the task of generating same-language subtitles was conducted. The prime motivation was a previous task, which was the extraction of sentence-utterances in different audio files using word-timestamps. Three different tools were used for this work: *aeneas* [1] which is an FA, *Cerence* [2], which is an ASR and *Sonix* [3], which is also an ASR. We conducted a technical evaluation and a subjective evaluation based on a case study. In this study people were presented with different stimuli, each stimulus using generated subtitles based on the time-information given by the different tools mentioned above. The resulting data of a case study confirmed a rise in performance of *Cerence* compared to *aeneas*.

1 Introduction

Automatic subtitling is generally the process of generating subtitles given a medium that includes an audio source containing speech. It is used in transcription and dictation tools or in more recent media platforms like YouTube (discontinued). As to support people with a hearing impairment automatic subtitling is a necessity, all the more so due to the excess of growing media and the high cost of human resources for creating subtitles [22].

There are different methods to generate subtitles automatically. In this work, the FA tool *aeneas* used a "classic" method using the Mel-frequency cepstral coefficients of the original wave file and of a synthesized wave file to determine the correct timestamps [18]. *Cerence* and *Sonix* are commercial state-of-the-art speech recognition providers and use more modern methods which are not mentioned by name. Regardless, Section 2.1 will show some theoretical background on how a possible modern approach could be realized.

The goal is to find out if the approach of using classic FA methods is better suited to generating subtitles automatically, or if a more modern approach using ASR methods is more viable for the task at hand.

2 Fundamental Concepts

In this section we outline the theoretical background to the more recent methods used in ASR and the subtitling guidelines. Both are used in the system that is described in Section 3.

2.1 Theoretical Background

An FA needs two arguments, an audio file containing speech and a text file with the transcript of the audio file's speech. It then aligns the transcript in respect to the speech. It is quite similar to speech recognition on the outside; both of these tools try to align audio to a sequence of phonemes and then try to match this sequence with a word [4]. A lot of the available forced aligners are based on Hidden Markov Models (HMMs), and it is common to use the Viterbi Algorithm for aligning the text and speech, namely "Viterbi forced alignment" [5,6,7,8]. A forced-aligner can be trained on "audio-sequence to phoneme"-pairs. To do so, the original transcripts of the audio files are translated (via a lexicon) into a phonetic representation. Then the results of an "audio-sequence to phoneme"-speech-model is matched with the phonetic representation of the original transcript. Although the FA *aeneas* does not use this kind of "audio-

sequence to phoneme"-speech-model, as mentioned in the introduction. [18] (more details in Section 3.2.1)

An ASR needs only one argument, an audio file containing speech. In fact, many speech recognizers use the same concept that are used in forced aligners. The major difference is the final matching step, the forced-aligner uses an original transcript (translated to a phonetic representation) to match the phonetic representations with each other, whereas ASR uses the results of an „audio-sequence to phoneme"-speech-model and translates these directly via a lexicon of phonetic representation of words to the actual words [8,9]. The transitions between phonemes are represented as an HMM.

In a publication about a system that used image recognition to recognize keywords in an audio file [10] the model itself did not match audio sequences with phonemes but took a one-second-long audio-sequence as an argument to determine if a given keyword was uttered within this second. An „audio-sequence to phoneme"-speech-model can work similarly: For training such a speech-model, a Convolutional Neural Network [11] was set up to recognize formants in a spectrogram. The spectrogram was generated via the Fourier Transformation [9,12] and the Convolutional Neural Network used image recognition to distinguish phonemes. Because the input layer of a CNN has a fixed number of input-neurons, the spectrogram needed to be split into multiple smaller matrices, each with a fixed shape to match the input-neurons. Furthermore, transitions of e.g., the HMM which point from one state to its original state can be used to filter out multiple consecutive phonetic symbols. E.g., fricatives have longer pronunciation, „audio-sequence to phoneme"-speech-model recognizes a small section of the spectrogram as „ ∫ ", it is probable that for the next section, the speech-model will recognize another „ ∫ ", because there might not have been passed enough time between the two sections.

Sequence Alignment, which closes the gap between an FA and an ASR regarding the usage of a transcript, is a term from Bio Informatics. It is used to align two DNA strings. It is based on the Levenshtein-/edit-distance and can be differentiated between global, semi-global and local versions. It determines the cost of how many changes or operations have to be performed, to turn a string into another string. Terms for these operations are defined as Substitution, Insertion and Deletion. Normally, the cost is chosen, so that the goal of alignment is reached with a low cost. Meaning that e.g., if a Substitution does not apply, the cost should be low. Substitution, insertion and deletion should have a high(er) cost. With help of Sequence Alignment, wrongly recognized words by ASR can be found, as it looks for common subsequences. Implementation wise, Sequence Alignment uses tables to compare two containers (in this case the containers are DNA strings), each row representing one element of the first container and each column representing one element of the second container. The costs of the operations are recorded in the table accumulatively and the optimal path is computed, representing the lowest costs for turning the first container into the second [13].

2.2 Subtitling guidelines

When writing subtitles, there are a few general rules that should be followed. In this work BBC's guidelines to writing subtitles were used and the most important points are listed here:

- Recommended words per minute (WPM), the duration of a displayed subtitle should be around and not exceed 160-180 words per minute. (~330-375ms per word)
- Subtitles should be displayed for at least 300 ms per word
- A displayed subtitle should not be shorter than 1 second.
- A displayed subtitle should aim to contain a single sentence.
- One line of a displayed subtitle should not exceed 37 characters.
- If they exceed 37 characters, they should be displayed in 2 lines.
- If a subtitle has the size of 2 lines, the line break should be at „logical points", which means that two words that „belong together" syntactically should not be split up.
- The displayed subtitles should match the speech onsets and offsets.

- Consistent timing, consecutive subtitles of similar length should be displayed for the same amount of time

There are many different standards to which subtitles can be written e.g., Netflix's limit to the length of a line amounts to 42 and the speed of subtitles to 17 characters per second. But each standard's main argument results in the "readability" of subtitles, which is a vague term. It is also important to consider that these standards relate mostly to subtitles for videos [14,15].

3 System

In this section the system used for the process of automatic subtitling is described. Also, some technical information regarding the tools used are shown.

3.1 Procedure

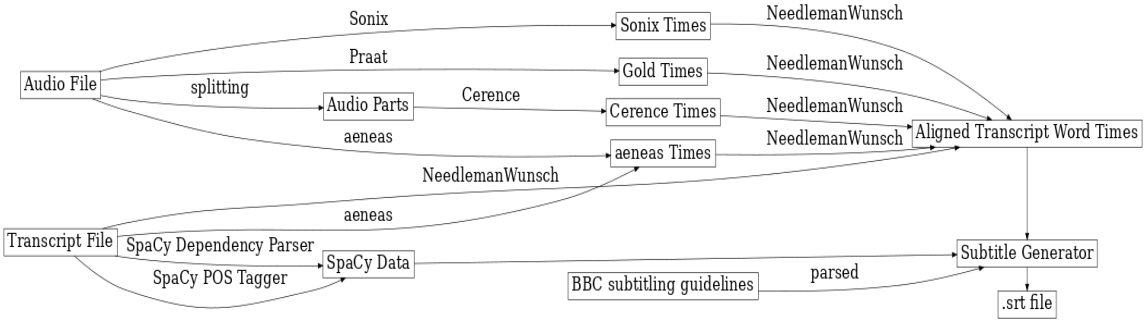

Figure 1 - Pipeline of Tools used

Figure 1 depicts the tool pipeline that was implemented to generate subtitles. First gold standard annotations were created by hand using *Praat* [16], providing the gold standard word-timestamps. For the word-timestamps from the FA, *aeneas* was run using the audio (all in German) with the given transcripts, providing the word-timestamps of the FA-tool. For C*erence*, if the duration of the audio file exceeded one minute, it was split into shorter parts. The parts had a duration of 50 seconds, as to provide an overlap of 10 seconds between two parts as well as to match the audio and times more reliably. Also, it was crucial not to split words while uttered. For *Sonix* there was only the need to upload the audio manually. For the transcript, *SpaCy* [17] was used to parse the text extracting syntactic information.

Needleman-Wunsch (further explained in Section 3.2.4) was applied to each individual word-timestamp with the transcript to include punctuation marks and the correct capitalizations in the final subtitles. But mainly to provide the timing of the uttered sentences.

The optimal paths (Section 2.1) through the tables generated by the sequence alignment were split at corresponding whitespaces, providing the indices used for splitting the texts into subtitles. Because some words were tagged together (e.g., "die sehr" with one start and end time), not every index could be used.

If the length of a sentence did not exceed 37 characters, only the start-timestamp of the first word and the end-timestamp the last word was used.

For determining which splits would provide the best result, the sentences were split at every possible combination of indices. To avoid single words in a row of a subtitle before and after a single word it was not possible to split. Spans of texts did not exceed 37 chars. Additionally, every index had a specific value representing the "splitability" of that particular index. E.g., the index after a punctuation mark like a comma had the highest "splitability"-value. Also, information from SpaCy was used providing a dependency-tree and POS-tags, so the index after a word classified as an article had a very low value. Also, the leaves of the dependency-tree (if the leaves were neighbours in the actual text) had a very low value. The higher a neighbouring word and a phrase were positioned in the hierarchy (with the root of the tree as the highest

position) the higher was its corresponding value. Each batch of spans for each sentence were scored. The score depended on the following:

- the afore mentioned “index-value” (higher the better)
- the lengths of the spans (longer the better)
- count of splits (lower the better)
- duration of spans (spans with similar lengths were better)

The scores were normalized and the spans with the highest score was chosen.

Finally, in the generation of the actual subtitle files, following the srt-format, the number of spans was checked. If this resulted in an even number all subtitles consisted of two rows, each row being one span of text. If the number was uneven the longest pause between spans was determined as to which span was to stand in a one row subtitle. If no pauses were available, the last span stood alone. Through this scoring, subtitles for the same audio file sometimes differed for each tool. The WPM was not considered in the generation of the subtitles due to a lack of time. In the next subsection we describe some tool specific characteristics.

3.2 Tools

3.2.1 *aeneas*

aeneas is a tool used for FA. It is available as a *python* library, which simplifies its usage. The inner workings can be explained in one sentence, stated in the documentation [18]:

“Using the Sakoe-Chiba Band Dynamic Time Warping (DTW) algorithm to align the Mel-frequency cepstral coefficients (MFCCs) representation of the given (real) audio wave and the audio wave obtained by synthesizing the text fragments with a TTS engine, eventually mapping the computed alignment back onto the (real) time domain.” *aeneas* does not depend on techniques used in ASR, but uses a “classic, signal-processing-based approach” [18]. The word-timestamps provided by *aeneas* did not include pauses between words.

3.2.2 *Cerence*

Cerence (a spin-off of *Nuance*) is a tool used for ASR, specifically for real-time (online) speech recognition, directly spoken into a microphone. It streams directly onto a server and the recognition is run simultaneously. The usage was problematic for uploading a whole wave file, it was necessary to split the file into shorter parts because of a limitation set up by *Cerence*, which only allowed audio to be uploaded with a length of < 1 minute. Splitting the wave files and uploading the parts separately turned out to be more problematic than expected because of a seemingly inherent problems with ASR, where each part had a certain time-offset (of about 120ms), which carried over to the next part of the audio. This was handled by subtracting the mean of the timing differences. Additionally, in the results from *Cerence*, the timestamps of the words conflated with the pauses between said words.

3.2.3 *Sonix*

Sonix is a tool for transcribing audio. It uses speech-recognition to recognize words in a wave-file and timestamps every word. The usage of the API was restricted, so wave-files had to be uploaded manually and the timestamps had to be downloaded manually. Fortunately, the results were good, also providing the length of the pauses between words.

3.2.4 *Other Tools*

Three additional tools were used, *SpaCy*, *Praat* and the *Needleman-Wunsch algorithm*. *SpaCy* is a powerful open-source Natural Language Processing library for *python*. *Praat* is “...a computer program with which you can analyse, synthesize, and manipulate speech...” [19].

The *Needleman-Wunsch algorithm* is used for global sequence alignment of two strings [20]. It was used for two tasks: First in the alignment of the individual wave file parts, which were

created for the *Cerence* ASR. It was used to align the overlapping 10 seconds of word-timestamps on a word-level, meaning every row represented one word from the first word-timestamps (last 10 seconds) and every column represented one word from the second word-timestamps (first 10 seconds). In the second case it was used to align the word-timestamps provided by the different tools and the gold standard with the transcript of the audio file. Here the alignment was done on a character-level, meaning every row represented one character from all the words (joined by whitespace) of the word-timestamps and every column represented one character from the transcript. Global sequence alignment on word-level is also used for calculating the word error rate (WER) where the number of matches is counted.

Table 1 - WER for each ASR tool and an older version of *Cerence* (old Cer.)

Cerence	97.4%	94.4%	97.6%	97.3%	98.6%	96.7%
Sonix	91.0%	90.2%	90.0%	90.4%	88.6%	91.8%
old Cer.	97.0%	82.0%	/	/	/	/

4 Technical Evaluation Data

As it can be seen in Table 1 there was a huge improvement to *Cerence*'s ASR regarding the WER in comparison to when this tool was started to be used for this work (third column). On average *Sonix*'s WERs have a lower value than *Cerence*'s, which indicates a superiority in the word recognition domain of *Cerence*. The last 4 columns represent the audio files which have been used for the case study in Section 5.

Table 2 – WPMs/rounded mean of WPMs for each subtitle file for each tool

Tool/Speaker	Speaker 1	Speaker 2	Speaker 3	Speaker 4
aeneas	69-200 / ~134	68-170 / ~116	82-151 / ~116	80-176 / ~109
Cerence	84-187 / ~133	68-177 / ~117	80-150 / ~114	74-163 / ~108
Gold	93-206 / ~154	88-196 / ~141	95-151 / ~127	89-318 / ~147
Sonix	89-202 / ~151	88-196 / ~138	98-151 / ~127	88-165 / ~122

Table 2 shows the WPMs of all generated subtitles used in the case study. On average, *aeneas* and *Cerence* have a higher WPM than *Gold* and *Sonix*, due to the pauses in between the word-timestamps that were only included in the latter two.

5 Evaluation on the basis of a Case Study

To evaluate the generated subtitles a case study was conducted. The aim was to investigate if timestamps provided by the ASR tools or the subtitles generated by the FA will result in better subtitles compared to the usage of the gold timestamps. Our hypothesis is that the usage of the *Gold* timestamps or the timestamps provided by the ASR will result in better subtitles.

5.1 Preparation

For this study 5 audio files were selected, with a duration between 34 – 37 seconds. Every audio file was combined with subtitles generated on the base of every tested tool *aeneas, Cerence*, *Sonix* and the gold standard, amounting to 16 different stimuli. The stimuli were generated as video files with a black background and hardcoded white subtitles using *ffmpeg*. To cover all permutations of tools, every person would be subjected to 4 videos with different speakers, each generated with a different tool and one with the gold standard, amounting to 24 permutations

of stimuli. The order of the shown speakers was randomized. Also, one additional separate subtitle video was generated with the usage of the gold standard. The platform on which the stimuli were shown was written in *html*, *javascript* and *css*.

5.2 Procedure

The participants were instructed to rate each subtitle video. It was pointed out to each participant that the emphasis should be placed on the timing of the shown subtitles. A short demographic questionnaire was conducted. Then the additional separate subtitle video was shown to every participant to establish a baseline and showing what kind of stimulus was considered as the "best" stimulus (this same subtitle video was shown to every participant). It was pointed out that the shown subtitle video is the "best" stimulus. In the first part of the study every video was shown once separately. Each was subjected to a scoring from 1 – 4 (good – bad). In the second part the participant was able to look at the same 4 stimuli again but not separately and with the possibility to pause and rewatch the videos as many times as needed. It was possible to leave a comment for each of these videos which was not possible in the first part. Here a ranking of the 4 videos was demanded from ranks 1 – 4 (good – bad).

5.3 Participants

18 of the 24 study participants replied with their results, 8 being men and 10 being women. The age-span ranged from 21 - 33 years. The highest levels of education ranged from Abitur to Promotion. The majority of the participants specified German as their native language. Two German/Polish bilingual speakers and a single native Spanish speaker participated. 9 participants used subtitles frequently, 8 did not and one only used subtitles sometimes.

5.4 Results & Discussion

The scores of the videos from the first part of the study are not discussed any further due to the fact that the values of the scores were all too close together (range of scores: 26-32).

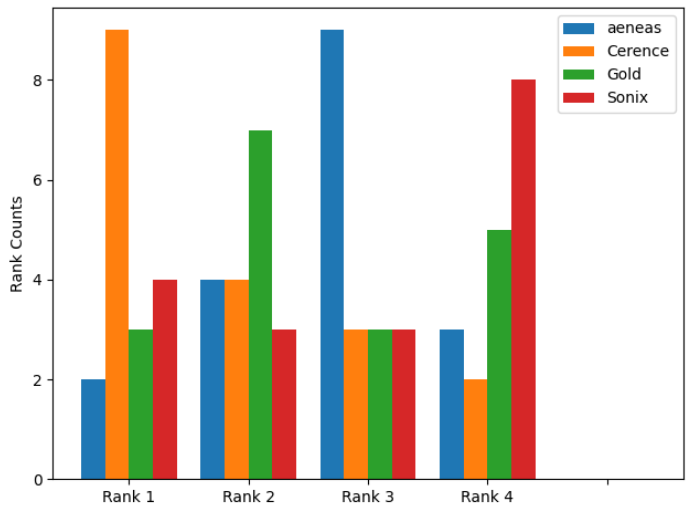

Figure 2 - counted rankings for each tool

Figure 2 is a visualization of the counts of the rankings from the case study. The different tools are represented by differently coloured columns and visually it seems clear that the tools were ranked in a distinct manner.

Table 3 - Mean ranks for each tool, sorted after the mean value

Cerence	*Gold*	*aeneas*	*Sonix*
1.889	2.556	2.722	2.833

With the usage of *Kendall's coefficient of concordance* [21] which can be used to assess whether there is an agreement amongst participants of a study, the ranking data was tested for significance using a χ^2-test.

Hypotheses:

- H0: There was no agreement or concordance between the participants regarding the ranking of the subtitle videos
- H1: There was an agreement or concordance between the participants regarding the ranking of the subtitle videos

At a level of significance of $\alpha = 0.05$ and a degree of freedom of $df = 3$, the null Hypothesis H0 was accepted, $\chi^2\ (3, 18) = 5.8$, $p > 0.05$, there was no agreement between the participants. (p-value of ~.13)

Nevertheless, on further inspection of the data two datapoints seemed irregular. All scores from the first part of the study had the same value, also it was possible to deduct from the comments left by those participants that their focus on the study was incorrect, rather focusing on e.g., the pitch of the voice and not on the timing.

If either one of those two datapoints was excluded, the null Hypothesis was rejected both times, $\chi^2\ (3, 17) = 8.0$, $p > 0.05$. Furthermore, the ranking itself in Table 3 was maintained. It could be safe to say that the ranking in Table 3 is accurate.

6 Discussion

Considering the technical evaluation of WPMs (Table 2) the *Gold* and *Sonix* subtitles had higher means than *aeneas* and *Cerence*. The *Gold* subtitles may have not reached the first place due to this. In the case study some participants seemed to be perturbed by the pauses between subtitles and them being shown to abruptly which was also pointed out in the comments. Pauses between subtitles cause this effect on the WPM. There is a chance of a better result for *Sonix* and *Gold* if the pauses in the audio files were longer thus showing subtitles for an extended time.

Additionally, *Sonix*'s WER was also subpar in comparison to *Cerence*. There could be a correlation between the WER and inaccuracies in timestamps which could have hence led to worse subtitles.

The WPMs stated in the BBC guidelines were not reached by any tool (regarding the mean value) which could mean that the participants do not confirm to this standard.

As to the hypothesis that the usage of the *Gold* timestamps or the timestamps provided by the ASR will result in better subtitles has proven to be only partially true. *Cerence* as an ASR tool did have the best ranking. It is unexpected that the performance was better than the one of the *Gold* standards. *Sonix* as an ASR tool however did land on the last place after the FA tool *aeneas*. It might not be possible to conclude that the ASR tools used are better suited to generate subtitles. Nevertheless, it seems evident to say that *Cerence* is better suited for generating subtitles than *aeneas* when considering the timing of said subtitles.

Although there were comments about the difficulty of observing differences in the presented stimuli, the results were reasonably distinct which assures the validity of the study and the thesis.

7 Conclusion

Concerning the topic of automatic subtitle generation, *aeneas* drops out of consideration with its more “classic” methods of signal-processing while the ASR method illustrates the better performance, but only if the said methods are implemented sufficiently well.

The performance of the *Gold* subtitles hints at the fact that creating timestamps by hand might not be the best method for recognizing and determining word timestamps.

Although there were few deliberately discernible differences for the normal user there appears to be a residual factor between the generated subtitles.

In our future work, we plan to implement the consideration of the WPMs and to make better usage of the syntactic information.

Acknowledgements

This research is part of the ChiM project of the research initiative " KMU-innovativ: Mensch-Technik-Interaktion", which is funded by the Federal Ministry of Education and Research (BMBF) of the Federal Republic of Germany under funding number 16SV8331.

References

[1] https://www.readbeyond.it/aeneas/
[2] https://www.cerence.com/
[3] https://sonix.ai/
[4] https://www.isip.piconepress.com/projects/speech/software/tutorials/production/fundamentals/v1.0/section_04/s04_04_p01.html
[5] Ljolje, A. and Riley, M.: *Automatic segmentation and labeling of speech* in Proc. IEEE Int'l Conf. Acous., Speech, and Signal Processing, 1991.
[6] https://github.com/pettarin/forced-alignment-tools
[7] Gales, M.J.F. & Young, S: *The Application of Hidden Markov Models in Speech Recognition*. Foundations and Trends in Signal Processing, 1, pp. 195-304, 2007.
[8] Jurafsky, D. & Martin, J.: *Speech and Language Processing: An Introduction to Natural Language Processing*, Computational Linguistics, and Speech Recognition, 2008.
[9] Huang, X., Acero, A. & Hon, H.-W.: *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, 2001.
[10] Krishna Gouda, S., Kanetkar, S., Harrison, D. and Warmuth, M.K.: *Speech Recognition: Keyword Spotting Through Image Recognition*, 2018.
[11] Albawi, S., Mohammed, T. A. and Al-Zawi, S.: *Understanding of a Convolutional Neural Network*, 2017
[12] Bracewell, R. N.: *The Fourier Transform and Its Applications*, 1965
[13] https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-096-algorithms-for-computational-biology-spring-2005/lecture-notes/lecture5_newest.pdf
[14] BBC Subtitles Style Guide: https://bbc.github.io/subtitle-guidelines
[15] Netflix Subtitles Style Guide: https://partnerhelp.netflixstudios.com/hc/en-us/articles/215758617-Timed-Text-Style-Guide-General-Requirements
[16] https://www.fon.hum.uva.nl/praat/
[17] https://spacy.io/
[18] https://github.com/readbeyond/aeneas/blob/master/wiki/HOWITWORKS.md
[19] https://www.fon.hum.uva.nl/praat/manual/Intro.html
[20] Needleman, S. B., Wunsch, C. D.: *A general method applicable to the search for similarities in the amino acid sequence of two proteins*, Journal of Molecular Biology, Volume 48, Issue 3, pp. 443-453, 1970
[21] Kendall, M. G., & Babington Smith, B.: *The Problem of m Rankings*, The Annals of Mathematical Statistics, vol. 10, no. 3, pp. 275–287, 1939
[22] Garcia Laínez, J. E. & Ortega Gimenez, A. & Lleida Solano, E. & Lozano, T. & Bernues, E. & Sanchez, D.: *Audio and text synchronization for TV news subtitling based on Automatic Speech Recognition*, IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, Bilbao, pp. 1-6, 2009.
[23]Zekveld, Adriana A. & Kramer, S. E. & Kessens, J. M. & Vlaming, M. S. M. G. & Houtgast, T.: *The Influence of Age, Hearing, and Working Memory on the Speech Comprehension Benefit Derived from an Automatic Speech Recognition System*, Ear and Hearing, Volume 30, Issue 2, pp. 262-272, 2009