Advances in Signal Processing 2(3): 75-78, 2014
DOI: 10.13189/asp.2014.020301 http://www.hrpub.org

System Determining Pronunciation Correctness of Japanese Words

Z. Syroka[1,*] , P. Dubiłowicz[2], T. Zając[2]

[1]University of Warmia and Mazury, Technical Sciences Department, Oczapowskiego 11 Street,10-719, Olsztyn, Poland
[2]University of Warmia and Mazury, Faculty of Mathematics and Computer Science, Słoneczna 54 Street, 10-710, Olsztyn, Poland
*Corresponding Author: syrokaz@onet.eu

Abstract A system determining pronunciation correctness of Japanese words is presented. The system is composed of six separate modules: signal segmentation, transcription interpretation, synthesis, comparison, error interpretation and phonetic database. The system uses two input sources: speech signal and its Latin transcription. The system is designed to deal specifically with Japanese language, due to the language's specificity in terms of speech. Based on the input, a timeline pointing out the locations of vowels is calculated, followed by determining the locations of all the moras (syllable timings). The signal, segmented in such a way, is then compared with a signal synthesized using a phonetic database and the transcription data. As a result the differences are pointed out and interpreted.

Keywords Signal Segmentation, Speech Segmentation, Single Moras, Diphones

1. Introduction

Numerous speech segmentation techniques have been devised, varying in application, but since the presented system is used specifically to check the pronunciation of single words only phonetic segmentation was considered. Apart from thematic segmentation, word segmentation [8, 9, 11], feature segmentation (used in automated creation of linguistic models) [7] and specialized segmentation used in specific systems [3, 4, 5, 6] most commonly they deal with the lowest level of segmentation – breaking up into phone strings [2, 3, 4, 10, 12]. Signal analysis of this sort is indeed accurate, yet it requires a huge phonetic database. In case of the Japanese language, such a database and accuracy of analysis is highly redundant. Japanese language follows certain patterns; every word is composed of moras (mora can be considered as a phonetic timing unit), which are limited in number and may be pronounced in only one way; Due to the aforementioned patterns there is a very limited number of sounds which have to be considered and put inside a phonetic database (between 200 and 300 sounds (whole syllables and diphones) depending on the comparison technique implemented). Research has also been done in the area of syllable segmentation, yet moras are smaller units than syllables, thus the technique would not yield the required results.

2. Phonetic Database

Due to the relatively small size of the phonetic database, one of the approaches may be to have each user record the required samples, which would then be automatically saved in the required format (filtered and resampled). The user would be instructed only to repeat single or double moras after a prerecorded native speaker. This approach leaves the risk of a user making a pronunciation error in the process of creating the database, leaving this error to propagate incorrect pronunciation later on; the problem is solvable by adding an additional comparator module, based on a prerecorded phonetic database determining with some accuracy the correctness of mora pronunciation. The main reason for creating a database by the user is the highest possible accuracy in the process of comparison between the database segments and input signal segments.

Alternative approach to the database construction is to provide more comfort and ease of use to the user, by not requiring them to record their own samples; but it leaves the comparator module open to some inaccuracies coming from the comparison method or the quality and size of the database. Since in both cases a prerecorded database needs to exist (Fig.1. – Phonetic database block), the choice depends purely on the user's comfort level that is to be achieved.

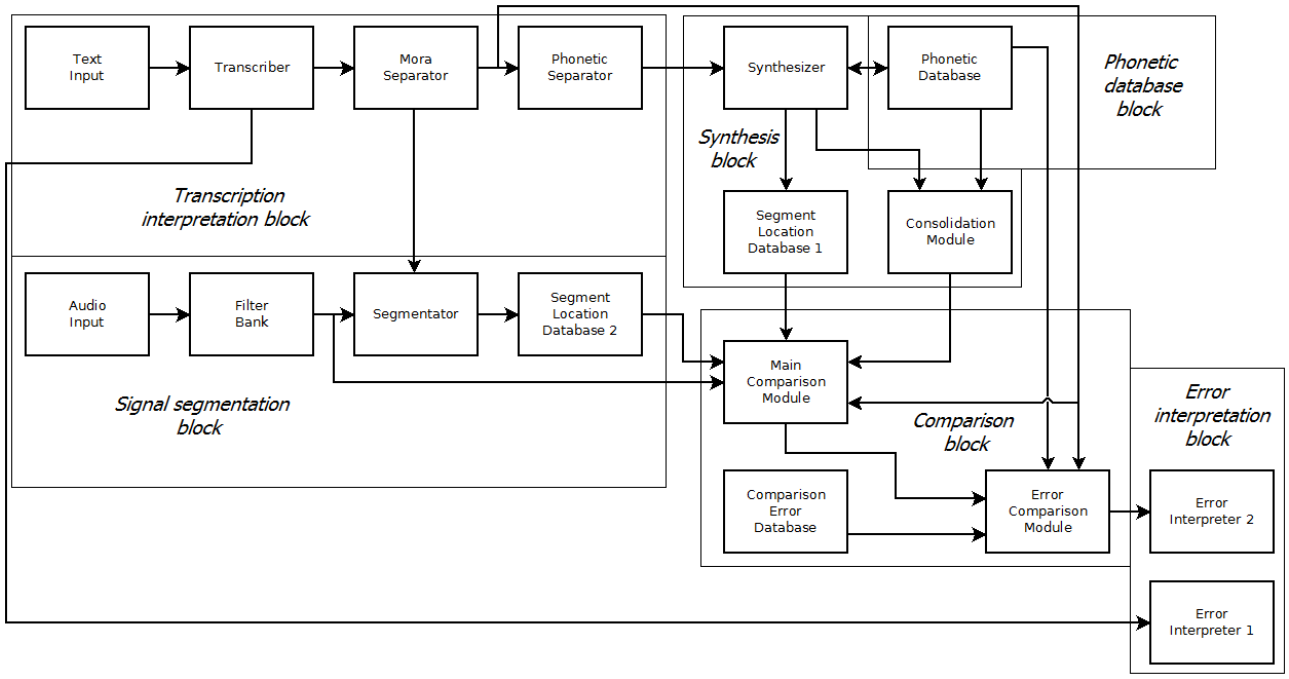

Figure 1. General system schematic, excluding the optional database creation block.

3. Transcription and Synthesis

The transcription module (Fig.1 – Transcription interpretation block) converts a text string into a set of moras, a series of phonetic units and some information about relations between phonetic units. The transcriber (Fig.1 – Transcriber) receives a text string in romaji (Latin transcription of Japanese language) (Fig.1. – Text input) and interprets it into proper moras (Fig.1 – Mora separator). Next the moras are turned into phonetic units and information about their surroundings (Fig.1 – Phonetic separator). The resulting data is then used to synthesize the correct pronunciation of the given word (Fig.1 – Synthesizer). Synthesis problem in case of the Japanese language is less complex than with most other languages [18, 19, 20]. Each mora has only one pronunciation, there exist geminations (doubling of a consonant or its elongation), vowel elongations, disappearing vowels and acute accent on a few specific consonants. Transcription is based on a small set of predefined rules to separate romaji text string into single moras and relations between them. The series of single moras and relations is then used to concatenate (Fig.1. Consolidation module) the corresponding phonetic units from the database (Fig.1 – Phonetic database) in order to create a template for comparison with the input signal.

4. Moraic Segmentation

The signal segmentation module (Fig.1 – Signal segmentation block) prepares the input signal (Fig.1 – Audio input) for comparison. The signal is first filtered and if necessary – resampled (Fig.1 – Filter bank). The most crucial filtration is silence removal [13], to minimize the size of data processed. Next, the segmentator receives the processed input signal as well as a series of moras extrapolated from the romaji transcription and uses both of them to create a series of breakpoints between moras and phonetic units (Fig.1 – Segment location database 2). First step of this process is finding the locations of vowels based on a pseudo-envelope in time analysis of the signal. According to the transcription, the number of vowels and their type is known along with an estimated location, furthermore, the fact that vowels have considerably higher amplitude than consonants makes the task relatively easy to accomplish with maximum accuracy. Most moras will be composed of one vowel, but in some cases it will contain an elongated vowel (2 moras), a geminated consonant (2 moras), the N mora (consisting of only one nasal consonant) or a silent vowel (based on a few linguistic rules [20]). It is possible to consider that each mora that has a vowel will end with it, so the end of a vowel will automatically become a breakpoint between two moras. Gemination is considered to be an elongation of a consonant, which in practice becomes a pause between two same consonants, and halfway during the pause a breakpoint is inserted. Elongated vowels should be analyzed in time to validate the correct length of the vowel. Consonant-consona nt sequence should be separated using the Phonetic Database and the information obtained during the transcription.

5. Comparison

The comparator (Fig.1 – Comparison block) receives the filtered input signal and the synthesized. Using the breakpoints calculated with the segmentation module and during the synthesis (Fig.1 – Segment location database 1), the signal is separated into moras. Next, based on data

received from the transcription block the two signals are compared by the main comparator (Fig.1 – Main comparison module).

There is a multitude of signal comparison algorithms that could be used to point out the differences between the two signals. Depending on the type of errors that are to be found, and on the Phonetic database type (general or user-specific), some will be significantly more accurate than the others. The whole system should be specified for determining the correctness of Japanese word pronunciation by a non-Japanese native. Pronunciation errors made by native English speakers usually revolve around incorrect elongation of vowels and consonants or lack thereof, attaching English accent to vowels and consonants which results in completely different moras, as well as incorrect intonation. Polish speakers for example don't have a problem with attaching accent, but the other problems do persist. In addition there are many cases of using "sh" (voiceless retroflex fricative /ʂ/) (which doesn't appear in Japanese pronunciation) instead of "soft sh" (voiceless alveolo-palatal fricative /ɕ/) and other similar errors. Each type of mora can only be mispronounced in a few specific ways, thus it is possible to associate different moras with various comparison methods used to find very specific differences. Additionally creating the Phonetic database by each separate user will increase the accuracy of most of the comparison methods, but will require some degree of control as well.

The methods [14, 15, 16, 17] considered for use are time-frequency analysis (Gabor transform and wavelet transform), spectral analysis (Wigner-Ville transform) and LPC coefficients. In case of user-specific database it is possible to use LPC as the main comparison method, which requires considerably less computation than the other methods and provides high accuracy for most moras. Other methods would then be considered as backup.

In case when the database is defined from the start it may be necessary to sacrifice computation time for getting the required result, in which case the Gabor transform is used as the main comparison method. To offset the computation time it is possible to lower the signal's quality with a set of filters to deprive it of unique voice features, and leave it is a raw speech signal.

6. User-specific Phonetic Database

In creating a user-specific Phonetic database, the user has to input a signal containing a predetermined sequence of sounds repeated after the prerecorded native guide voice. The signal is then separated it into segments containing only the required information (without silence and in some cases comprising of only diphones) and then comparing it with a prerecorded phonetic database. A comparison phase is required to determine if the user is able to correctly pronounce the base sounds needed in the pronunciation of whole words. An error in pronunciation of a single sound would render the later comparison useless or at least insensitive to some pronunciation errors. This comparator block would have to use a different, smaller set of comparison methods used for short speech signals (from phonemes and diphones to syllables). Such a user recorded phonetic database will guarantee there are no errors due to the voice feature differences.

7. Result Interpretation

Finally the interpreter (Fig.1 – Error interpretation block) uses the information about whether there exists a difference in a given segment (provided by the main comparator) and then compares it further (Fig.1 – Error comparison module) with known errors or with specific features (Fig.1 – Comparison error database) to determine the type of error. Each type of error is associated with a description and instruction how to correct it (Fig.1 – Error interpreter 2).

Another interpreter (Fig.1 – Error interpreter 1) is connected with the transcription module to point out any errors in the text input. It finds errors based on a set of rules about romaji transcription and requires the user to be familiar with this technique.

8. Conclusions

A system of this kind can be really helpful for students in the earliest stages of learning the Japanese language, especially if they do not have any practical experience with the language and if they are trying to learn the language on their own. A system like that may be configured to work with all languages using moraic pronunciation and some of the other languages, but would be completely helpless with most languages, without a present dictionary containing phonetic form of each word (like in case of the English language). Further development of the system can make it sensitive to intonation and accent, but would require an expanded rule set and intonation/accent input for exceptions.

REFERENCES

[1] Z. Syroka, P. Dubiłowicz, T. Zając *System stwierdzający poprawność wymowy słów w języku japońskim* (Patent application, P401449, 11.02.2012)

[2] R. Du, *Method and apparatus for speech segmentation* (Patent application, US20100153109, 06.17.2010)

[3] A. D. Conkie, *Automatic segmentation in speech synthesis* (Patent application, US20090313025, 12.17.2009)

[4] A. D. Conkie, *Automatic segmentation in speech synthesis* (Patent application, US20070271100, 11.22.2007)

[5] Shu, Chang-Qing, *Systems and methods for implementing segmentation in speech recognition systems* (Patent

application, US20050209851, 09.22.2005)

[6] Kuo, Chih-Chung, *Automatic speech segmentation and verification method and system* (Patent application, US20050060151, 03.17.2005)

[7] A. Srivastava, Linguistic segmentation of speech (Patent application, US20040024585, 02.05.2004)

[8] Takizawa Takuya, *Method and apparatus for performing speech segmentation* (Patent application, US20020143538, 10.03.2002)

[9] Wu Yu-Chieh, *A unified machine learning-based Chinese word segmentation and part-of-speech tagging algorithm* (Patent application, TW20090138535, 11.13.2009)

[10] H. Romsdorfer, *Pitch period segmentation of speech signals* (Patent application, EP20090405233, 12.30.2009)

[11] H. Chen, Z. Han, Y. Yang, Automatic segmentation device of single-word speech signal (Patent application, CN20082222733U, 12.02.2008)

[12] Laine Unto, Korhonen Petri, *Method for the automatic segmentation of speech* (Patent application, WO2005FI00519, 11.30.2005)

[13] Theodoros Giannakopoulos *A method for silence removal and segmentation of speech signals, implemented in Matlab*

[14] *Springer handbook of speech processing* (Springer handbooks, 2008)

[15] Shie Quian, Dapang Chen *Joint time-frequency analysis – methods and applications*

[16] H. G. Feichtinger, T. Strohmer *Gabor analysis and algorithms – theory and applications* (1998)

[17] A. V. Oppenheim, R. W. Schafer *Discrete-time signal processing* (1999)

[18] Otake Takashi, Hatano Giyoo, A. Cutler, J. Mehler, *Mora or syllable? Speech segmentation in Japanese* (Journal of memory and language 32, 1993)

[19] *The handbook of East-Asian psycholinguistics* (Cambridge University Press, 2006)

[20] R. Huszcza, Maho Ikushima, J. Majewski *Gramatyka japońska volume 1 & 2* (Wydawnictwo Uniwersytetu Jagiellońskiego, 2003) [in polish]