J Inf Process Syst, Vol.20, No.6, pp.746~757, December 2024
https://doi.org/10.3745/JIPS.04.0325
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Korean Phoneme Boundary Detection Based on Time Domain Metrics

Jae Won Lee*

Abstract

This paper proposes a novel Korean phoneme boundary detection method that can be applied to phoneme-based Korean speech recognition systems. The proposed method employs two time-domain metrics—volatility and bulk metrics—as the foundation for phoneme boundary detection. The input speech signal is divided into blocks of 300 integer samples. For each block, the volatility metric is computed that adds up all the changes between neighboring samples within the block. A bulk is a grouping of consecutive samples with the same sign. For each bulk, two bulk metrics are calculated: bulk size and bulk length. Three dedicated algorithms that utilize both types of metrics are used to detect phoneme boundaries by recognizing vowels, voiced consonants, and voiceless consonants in turn. The experimental results show that the proposed method can significantly reduce the error rate compared to an existing boundary detection method.

Keywords

Bulk Metrics, Phoneme Boundary Detection, Speech Recognition, Volatility Metric

1. Introduction

The most efficient interface between humans and computers is voice, and the technology that enables machines to recognize human speech in order to use voice as an interface is speech recognition technology. Speech recognition technology has been continuously improving since the 1960s through decades of research and development [1]. Especially in recent years, with the rapid proliferation of smartphones among the general population, the demand for speech recognition technology that can operate in a mobile environment has been growing rapidly.

In speech recognition systems, the basic units of recognition are words, syllables, and phonemes. If words or phrases are the basic units of recognition, the number of words and phrases to be recognized is so large that it takes too much time and effort to develop a recognition system, and it is not easy to secure enough training data. Phonemes are often used as fundamental units of recognition because they are smaller than words and syllables, offering a consistent reflection of acoustic characteristics to the recognizer. Since the number of phonemes in Korean is only about 40, most of the large vocabulary-based speech recognition systems developed recently adopt phonemes as the basic unit of recognition [2]. However, accurately segmenting speech into phonemes is a challenging task, even when done by

Manuscript received May 20, 2024; first revision August 8, 2024; accepted August 26, 2024.
***Corresponding Author:** Jae Won Lee (jwlee@sungshin.ac.kr)
School of AI, Sungshin Women's University, Seoul, Korea (jwlee@sungshin.ac.kr)

hand. Detecting boundaries between phonemes is a very difficult and inconsistent task due to many variables, including the articulatory habits of the speaker, the context of the speech, and the physical environment of the speaker [3].

Existing phoneme boundary detection methods can be categorized into statistical boundary detection methods and acoustic boundary detection methods. Statistical methods use hidden Markov models to build linguistic models of phonemes [4,5]. Acoustic methods use feature coefficients such as Mel-frequency cepstral coefficients, linear predicative coding, energy, and formant frequencies instead of linguistic information [6-8]. Statistical methods seek objectivity in boundary detection by using probability information, but they are not very time-efficient because they require large amounts of speech data for training and the process of performing matching between input data and models [9]. In addition, both methods are not computationally efficient because they convert the speech signal to the frequency domain to obtain the coefficients that characterize the phonemes [10].

Statistical methods are more common for phoneme boundary detection that is not language-specific. Most of the studies that have attempted to achieve better phoneme separation performance by taking into account language-specific characteristics have adopted acoustic methods. Lee [11] presented a classification algorithm that considers the characteristic differences between voiced and unvoiced sounds in Korean speech and Seo et al. [2] proposed a formant scaling method to separate vowel segments in Korean. Lachachi [12] used the short-time Fourier transform of the speech signal to identify Arabic phoneme boundaries.

In this paper, unlike the author's previous study in [13], we propose a novel phoneme boundary detection method that can improve computational efficiency by detecting phoneme boundaries in the time domain instead of the frequency domain. The volatility metric measured for each block and the bulk metrics computed from each bulk, defined as the set of adjacent input samples of the same sign, are utilized as the base metrics for phoneme boundary detection. Vowel segments are first identified using the fact that vowel segments have a larger bulk size than voiceless consonants and a higher volatility than voiced consonants. Next, using a recognition algorithm that takes into account the lengths of the bulks, voiced consonants are identified among all consonants. Finally, a block with a gradual and pronounced increase in volatility adjacent to a vowel segment is determined to be the starting boundary of a voiceless consonant.

This paper is organized as follows. Section 2 introduces time-domain metrics, the volatility metric and the bulk metrics. Sections 3 and 4 describe how to recognize vowel segments and consonant segments, respectively. In Section 5, the performance of the proposed method is evaluated through experiments, and finally, conclusions are presented in Section 6.

2. Time-Domain Metrics

This section introduces the time domain metrics used as the basis for phoneme boundary detection. First, the volatility metric is defined, and then the calculation algorithm for the bulk metrics is presented. In the following sections, three algorithms are presented that utilize these basis metrics to recognize each phoneme and detect boundaries between phonemes.

The volatility metric is computed for each block of 300 input speech samples (input integers). The

volatility metric $Vol(i)$ for block i can be expressed by Eq. (1) when the index of the first sample of block i is FS_i, and the value of a sample is assumed to be 0 if there is no sample corresponding to the index. Fig. 1 shows the block-by-block values of $Vol(i)$ for '찌게(jjige)' measured using Eq. (1). Henceforth, $Vol(i)$ will be abbreviated to Vol.

$$Vol(i) = \sum_{k=FS_i}^{FS_i+298} |sample_{k+1} - sample_k|. \quad (1)$$

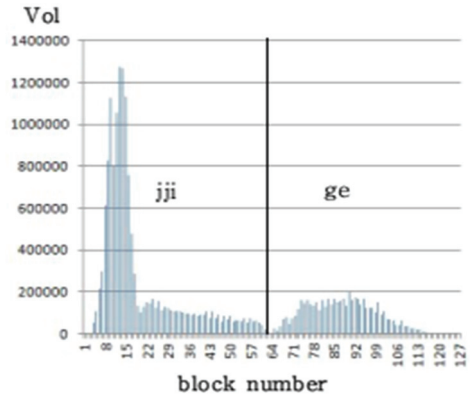


Fig. 1. Volatility metric of '찌게(jjige)'

A bulk is a group of adjacent samples with the same sign, and the boundaries of the bulk are formed whenever the sign of an adjacent sample changes. Vowel and voiced consonant segments often exhibit more regular and periodic waveform patterns than voiceless consonant segments, and the bulk metrics calculated for each bulk are intended to represent these patterns numerically. Fig. 2 shows an example of bulks appearing in the vowel 'ㅗ(o)' position.

Algorithm 1 is the algorithm that computes two bulk metrics: `bulk_size` and `bulk_length`. By thresholding the size, it ignores very small bulks that occur in intervals with frequent sign changes. `bulk_start` records the starting positions of the bulks. Based on the `bulk_start`, it is determined if each bulk belongs to a specific block. In this algorithm, *n* is the total number of input integer samples.

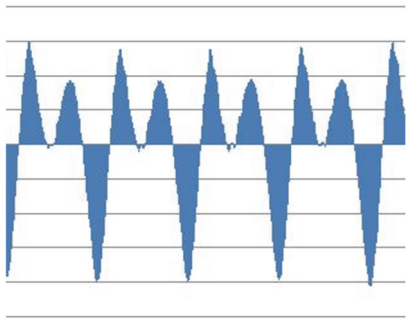


Fig. 2. Bulks of vowel 'ㅗ(o).'

Algorithm 1. Algorithm for calculating bulk metrics

```
Input: sample[0:n-1]: array of input integer samples
Output: arrays of bulk size, length and start
count := 0
tmp_size := sample[0], tmp_length := 1
bulk_start[0]=0;
for i := 1 to n-1 do
  if (sample[i-1]*sample[i]>0)
    tmp_size := tmp_size+sample[i]
    tmp_length := tmp_length+1
  end if
  else
    if (abs(tmp_size)>T1)
      bulk_size[count] := tmp_size
      bulk_length[count] := tmp_length
      count := count+1
    end if
    bulk_start[count] := i
    tmp_size:= 0
    tmp_length := 0
    tmp_size := tmp_size+sample[i]
    tmp_length := tmp_length+1
  end else
end for
```

3. Vowel Recognition

It is not easy to distinguish between consonants and vowels in terms of the volatility metric. This is because vowel segments tend to be more volatile than consonants such as 'ㄷ(d),' 'ㅂ(b),' 'ㅍ(p),' 'ㅁ(m),' 'ㄴ(n),' and 'ㄹ(r),' but consonants such as 'ㅅ(s),' 'ㅈ(j),' and 'ㅊ(ch)' often have very high volatility compared to vowel segments. On the other hand, from the perspective of bulk metrics, most consonants, except voiced consonants, are easily distinguishable from vowels. Fig. 3 is the waveform corresponding to the '그(geu)' in the word '그늘(geu-neul).' In the first part, which corresponds to 'ㄱ(g),' the bulks themselves are not formed, or if they are, they are very small. On the other hand, in the latter part, which corresponds to the 'ㅡ(eu),' bulks are formed almost throughout and are very large.

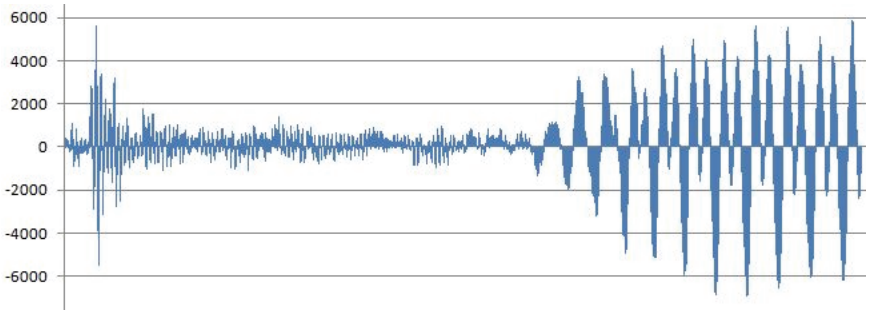


Fig. 3. Waveform of '그(geu).'

However, the voiced consonants ‘ㅁ(m),’ ‘ㄴ(n),’ and ‘ㄹ(r)’ also exhibit a number of very large bulks. Fig. 4 shows the ‘느(neu)’ part of ‘그늘(geu-neul),’ and we can see that large bulks are formed in both the leading ‘ㄴ(n)’ region and the trailing ‘ㅡ(eu)’ region. Nevertheless, voiced consonants are less volatile than vowels, so using a combination of bulk size and volatility metrics can be an effective way to identify vowel segments.

Algorithm 2 computes the Big metric, which expresses the degree to which large bulks appear in each block, and combines the Vol and Big metrics to compute the $Vowel$ metric, which is the base metric for vowel recognition. Operations are performed on all bulks in block i using `start_index[i]`, the index of the first bulk, and `end_index[i]`, the index of the last bulk. If the Big metric divided by the Vol metric is greater than a threshold, the block is considered not to belong to a vowel segment. For voiced consonants, the Big metric is large, but the Vol metric is small, so they are not included in a vowel segment, and for voiceless consonants, the Big metric is very small, so the $Vowel$ metric is also very small. In this algorithm, `num_of_block` is the total number of blocks and `num_bulk` is the total number of bulks.

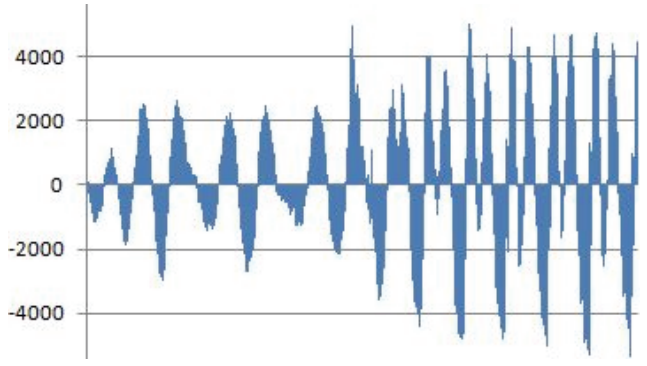


Fig. 4. Waveform of ‘느(neu).’

Algorithm 2. Vowel recognition algorithm

```
Input: bulk_size[0:num_of_bulk-1]
Output: Vowel[0:num_of_block-1]
for i := 0 to num_of_block-1 do
  sum := 0
  for j := start_index[i] to end_index[i] do
    if (abs(bulk_size[j])>T2)
      sum := sum+bulk_size[k]/10
    end if
  end for
  Big[i] := sum
  if (Big[i]/Vol[i]>T3)
    Vowel[i] := 0
  end if
  else
    Vowel[i] := Big[i]
  end else
end for
```

Figs. 5–7 are examples of the Vol, Big, and $Vowel$ metrics for '그늘(geu-neul)' respectively. A group of 10 or more adjacent blocks with a value of the $Vowel$ metric above the threshold T_4 is recognized as a single vowel segment. The index of the beginning block and the index of the end block of each vowel segment are stored in the `vowel_start` and `vowel_end` arrays

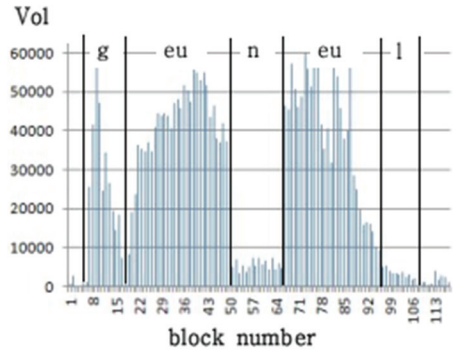


Fig. 5. Vol metric of '그늘(geu-neul).'

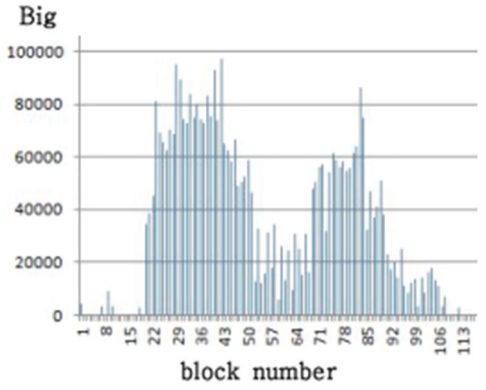


Fig. 6. Big metric of '그늘(geu-neul).'

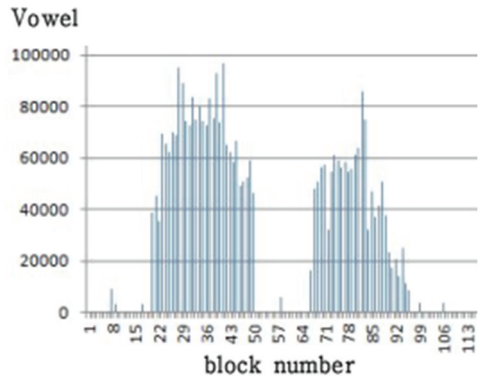
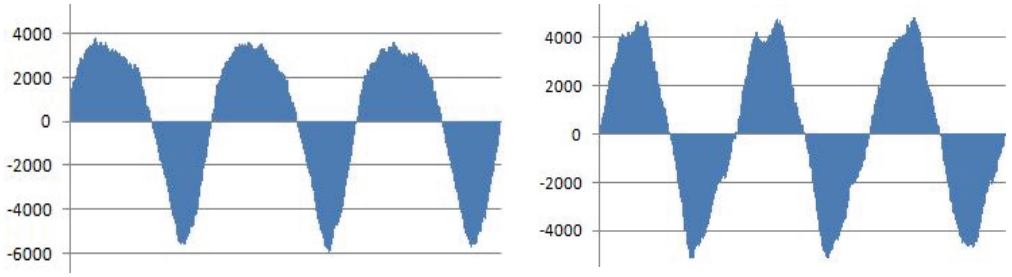


Fig. 7. Vowel metric of '그늘(geu-neul).'

4. Consonant Recognition

Voiced consonants have very different waveform characteristics from the rest of the consonants. Fig. 8(a) and 8(b) are the most typical waveform patterns of voiced consonants. A common characteristic of voiced consonant waveforms is that they appear in chains of very long bulks. Given this characteristic of voiced consonants, along with their low volatility, Algorithm 3 is used to recognize voiced consonant segments.



(a) (b)

Fig. 8. Waveforms of voiced consonant: (a) corresponds to 'ㄴ(n)' and (b) corresponds to 'ㅁ(m).'

Algorithm 3. Voiced consonant recognition algorithm

```
Input: bulk_length[0:num_of_bulk-1]
       and Vol[0:num_of_blok-1]
Output: Voiced_start[0:num_of_voiced_start-1]
       and Voiced_end[0:num_of_voided_end-1]
for i := 0 to num_of_block-1 do
  sum := 0
  for j := start_index[i] to end_index[i]-1 do
    if (bulk_length[j]+bulk_length[j+1]
       >T5 and Vol[i]<T6)
      sum := sum+bulk_length[j]
    end if
  end for
  Voiced[i] := sum
end for

k := 0
for i := 0 to num_of_block-1 do
  if Voiced[i] > T7 and (i=0 or Voiced[i-1] <= T7)
    Voiced_start[k] :=i
  else if Voiced[i] <= T7 and i > 0 and Voiced[i-1] > T7
    Voiced_end[k] := i-1
    k := k+1
  end if
end for

if Voiced[num_of_block-1] > T7 then
    Voiced_end[k] := num_of_block - 1
    k := k + 1
end if
```

The $Voiced$ metric expresses the degree to which each block falls within a voiced consonant segment as the sum of the lengths of the less volatile bulks. T_5 is the minimum value of the sum of the bulk lengths for recognizing voiced consonants. The algorithm identifies consecutive blocks where this metric exceeds the threshold T_7, combining them into single voiced consonant segments. The boundaries of these segments, which represent voiced consonant phonemes, are stored in the `Voiced_start` and `Voiced_end` arrays as output. Fig. 9 shows the Vol metric for '강남(gangnam)' and Fig. 10 shows the $Voiced$ metric obtained with Algorithm 3.

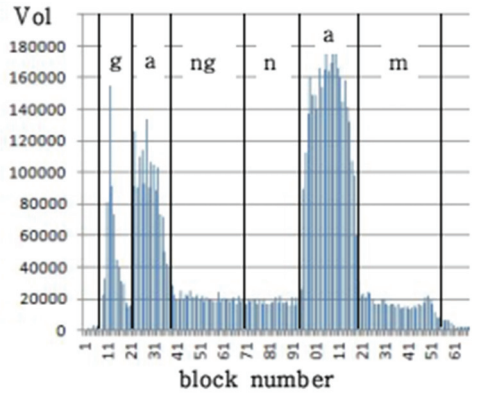


Fig. 9. Volatilities of '강남(gang-nam).'

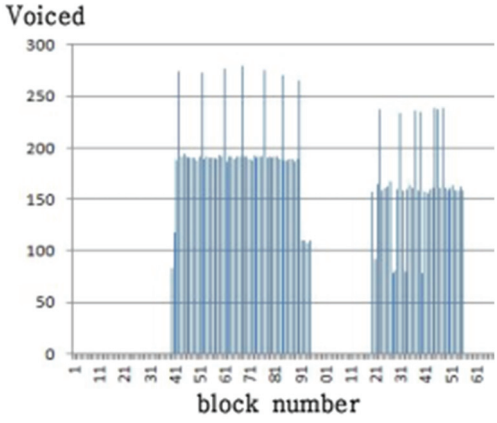


Fig. 10. Voiced metric of '강남(gang-nam).'

Algorithm 4 is the algorithm for recognizing voiceless consonants. For any block i, the volatility of that block is compared to the volatilities of the next six blocks to determine if a gradual increase in volatility occurs. If an increase occurs, the value of the $Voiceless$ metric becomes true in the sense that the block is a candidate for the start position of a consonant. If the values of the $Voiceless$ metric for adjacent blocks are all true, then all but the earliest block is excluded as a candidate for a voiceless consonant start, because only the first point of increased volatility is the consonant start. Finally, blocks that are within 35 blocks from a vowel's start position are considered as boundaries and stored in the

`voiceless_start` array, an array of voiceless consonant start positions. In this algorithm, `num_of_vowel` is the total number of vowels and `num_of_voiceless_consonant` is the total number of voiceless consonants.

Algorithm 4. Voiceless consonant recognition algorithm

```
Input: Vol[0:num_of_block-1]
Output:
 voiceless_start[0:num_of_voiceless_consonant-1]
for i := 0 to num_of_block-7 do
  count := 0, biggest := 0
  for j := i+1 to i+6 do
    if (Vol[j]>1.2*Vol[i])
      count := count+1
    end if
    if (Vol[j]>biggest)
      biggest := Vol[j]
    end if
  end for
  if ((count>4 and biggest>2*Vol[i])
    or (count>2 and biggest>5*Vol[i]))
    Voiceless[i] := True
  end if
  else
    Voiceless[i] := False
  end else
end for

for i := num_of_block-7 to 1 do
  if (Voiceless[i-1]=True)
    Voiceless[i] := False
  end if
end for

count := 0
for i := 0 to num_of_block-7 do
  if (Voiceless[i]=True)
    for j := 0 to num_of_vowel-1 do
      if (vowel_start[j]-i<35 and
         vowel_start[j]-i>0)
        voiceless_start[count] := i
        count := count+1
      end if
    end for
  end if
end for
```

5. Experiment

Experiments were conducted to measure the performance of phoneme boundary detection using the time domain metrics proposed in this paper as a new method for phoneme boundary detection. For the

experimental data, we used 3,000 recordings of speech consisting of 2–5 syllables. This marks a 15-fold increase in dataset size compared to the author's earlier study [13]. All data was recorded in the wave format of 16 kHz, 16 bit, and mono, using a smartphone in a normal noise environment. The thresholds for the four algorithms were determined through a heuristic process. Threshold T_1, the minimum bulk size in algorithm 1, was set to 7,000. Thresholds T_2, T_3, and T_4 for vowel recognition were set to 30,000, 2.2, and 8,000, and thresholds T_5, T_6, and T_7 for voiced consonant recognition were set to 150, 25,000, and 50.

If the length of the vowel segment recognized by the vowel recognition algorithm was 75 blocks or more, it was treated as side-by-side vowels with no final in the preceding syllable and no initial in the following syllable, such as '가위(ga-wi),' and the center of the vowel segment was treated as the boundary between the two vowel phonemes.

In addition, if the length of the voiced consonant segment recognized by the voiced consonant recognition algorithm was 50 blocks or more, it was treated as a case where a voiced consonant final and a voiced consonant initial appear side by side, such as '강남(gang-nam)' and the center point was treated as the boundary between the final consonant and initial consonant.

If the final consonant is a voiced consonant, the recognition algorithm can detect the correct boundary. However, if the final consonant is a voiceless consonant, it can be difficult to detect. If there is no initial consonant in the syllable after the voiceless final consonant, such as '학원(hak-won),' the final consonant is recognized as the initial consonant of the next syllable. In other words, '학원(hak-won)' is recognized as '하권(ha-gwon),' and in this case, the boundary detection was considered successful. However, if there is an initial consonant in the syllable after the voiceless final consonant, such as '학번(hak-beon),' the boundary detection was considered to fail because the final consonant is not recognized.

Table 1 shows the organization of the experimental data. The experiment was performed by separating the entire dataset of 3,000 words into two parts: dataset 'DS1,' consisting of 1,536 words with no syllable with a voiceless final consonant; and dataset 'DS2,' consisting of 1,464 words with one or more syllables with a voiceless final consonant.

Table 2 shows the comparison results of the phoneme boundary detection performance of the Fourier transform method proposed in [12] and the method proposed in this paper. Unlike the author's previous study [13], we conducted an evaluation to assess the accuracy of phoneme boundary detection. Boundary detection was considered correct if the distance between the real boundary and a detected boundary was within four blocks. Accuracy is the percentage of correctly detected boundaries out of all boundaries. A false detection (FD) is when there is no real boundary within four blocks before or after a detected boundary. The proposed method improves the accuracy by 10.2% compared to the Fourier transform method, which means that the error rate is reduced by about 40% compared to the Fourier transform method. In terms of false detection rate (FDR), the proposed method also performed better on both datasets.

Table 1. Experimental data

Dataset	Word	Syllable	Phoneme	Boundary
DS1	1,536	5,280	13,464	17,868
DS2	1,464	5,409	14,334	18,947

Table 2. Performance comparison of phoneme boundary detection

	Total boundary	Fourier transform				Proposed			
		CD	Acc (%)	FD	FDR (%)	CD	Acc (%)	FD	FDR (%)
DS1	17,868	14,419	80.7	2,502	14.0	15,974	89.4	2,055	11.5
DS2	18,947	13,263	70.0	3,088	16.3	15,461	81.6	2,539	13.4
Total	36,815	27,682	75.2	5,590	15.2	31,435	85.4	4,594	12.5

CD=correct detection, Acc=accuracy.

6. Conclusion

In this paper, a new Korean phoneme boundary detection method is proposed and implemented that can improve the computational efficiency of phoneme boundary detection by existing statistical or acoustic methods and reduce the error rate of boundary detection. By integrating time-domain metrics—volatility and bulk metrics—Korean phonemes, including vowels, voiced consonants, and voiceless consonants, can be separated efficiently and effectively. However, the handling of cases where the final consonant is a voiceless consonant is not fully accomplished. Additional consonant recognition algorithms should be developed to complement this in the future. There is also a need to develop phoneme separation algorithms that allow for more sophisticated separation of consecutive vowels and consecutive voiced consonants.

Conflict of Interest

The author declares that they have no competing interests.

References

[1] S. K. Gaikwad, B. W. Gawali, and P. Yannawar, "A review on speech recognition technique," *International Journal of Computer Applications*, vol. 10, no. 3, pp. 16-24. 2010. https://doi.org/10.5120/1462-1976

[2] Y. Y. Seo, J. D. Song, and J. H. Lee, "Phoneme segmentation in consideration of speech feature in Korean speech recognition," *Journal of Internet Computing and Services*, vol. 2, no. 1, pp. 31-38, 2001.

[3] M. Y. Nam, J. J. Lee, J. H. Park, and S. Y. No, "Recognition of Korean fricatives and affricates using modified Teager energy measurement method," *Proceedings of the IEEK Conference*, vol. 15, no. 1, pp. 23-26, 1993.

[4] S. Brognaux and T. Drugman, "HMM-based speech segmentation: improvements of fully automatic approaches," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 5-15, 2016. https://doi.org/10.1109/TASLP.2015.2456421

[5] P. B. Ramteke and S. G. Koolagudi, "Phoneme boundary detection from speech: a rule based approach," *Speech Communication*, vol. 107, pp. 1-17, 2019. https://doi.org/10.1016/j.specom.2019.01.003

[6] F. Kreuk, Y. Sheena, J. Keshet, and Y. Adi, "Phoneme boundary detection using learnable segmental features," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 8089-8093. https://doi.org/10.1109/ICASSP40776.2020.9053053

[7] K. K. Ravi and S. R. Krothapalli, "Phoneme segmentation-based unsupervised pattern discovery and

clustering of speech signals," *Circuits, Systems, and Signal Processing*, vol. 41, no. 4, pp. 2088-2117, 2022. https://doi.org/10.1007/s00034-021-01876-6

[8] B. Lin and L. Wang, "Learning acoustic frame labeling for phoneme segmentation with regularized attention mechanism," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, 2022, pp. 7882-7886. https://doi.org/10.1109/ICASSP43922.2022.9746880

[9] H. Frihia and H. Bahi, "HMM/SVM segmentation and labelling of Arabic speech for speech recognition applications," *International Journal of Speech Technology*, vol. 20, no. 3, pp. 563-573, 2017. https://doi.org/10.1007/s10772-017-9427-z

[10] A. E. Sakran, S. M. Abdou, S. E. Hamid, and M. Rashwan, "A review: automatic speech segmentation," *International Journal of Computer Science and Mobile Computing*, vol. 6, no. 4, pp. 308-315, 2017.

[11] Y. Lee, "Phoneme segmentation using phoneme combination and formant scaling in Korean," M.S. thesis, Department of Computer Engineering, Inha University, Incheon, Korea, 2003.

[12] N. Lachachi, "Unsupervised phoneme segmentation based on main energy change for Arabic speech," *Journal of Telecommunications and Information Technology*, vol. 2017, no. 1, pp. 12-20, 2017. https://doi.org/10.26636/jtit.2017.1.645

[13] J. W. Lee, "Phoneme segmentation based on volatility and bulk indicators in Korean speech recognition," *KIISE Transactions on Computing Practices*, vol. 21, no. 10, pp. 631-638, 2015. https://doi.org/10.5626/KTCP.2015.21.10.631

Jae Won Lee **https://orcid.org/0009-0009-3162-5699**

He received the B.S. degree in computer engineering from Seoul National University, Seoul, Korea, in 1990, the M.S. degree in computer engineering from Seoul National University, Seoul, Korea, in 1992 and the Ph.D. degree in computer engineering from Seoul National University, Seoul, Korea, in 1998. He has been a professor of the School of AI Convergence at Sungshin Women's Unversity, Seoul, Korea since 1999. His research interests include speech recognition, computational finance, and computational music.