Real-time Informatized caption enhancement based on speaker pronunciation time database

Yong-Sik Choi¹ · Jin-Gu Kang² · Jong Wha J. Joo² · Jin-Woo Jung²

Received: 18 February 2019 / Revised: 1 December 2019 / Accepted: 11 August 2020 /

Published online: 5 September 2020

© The Author(s) 2020

Abstract

IBM Watson is one of the representative tools for speech recognition system which can automatically generate not only speech-to-text information but also speaker ID and timing information, which is called as Informatized Caption. However, if there is some noise in the voice signal to the IBM Watson API, the recognition performance is significantly decreased. It can be easily found in movies with background music and special sound effects. This paper aims to improve the inaccuracy problem of current Informatized Captions in noisy environments. In this paper, a method of modifying incorrectly recognized words and a method of enhancing timing accuracy while updating database in real time are suggested based on the original caption and Informatized Caption information. Experimental results shows that the proposed method can give 81.09% timing accuracy for the case of 10 representative animation, horror and action movies.

Keywords Informatized caption · Speaker pronunciation time · IBM Watson API · Speech to text translation

Abbreviations

X	word
$T_s(X)$	Original caption of a word X
$T_s^+(X)$	Informatized Caption of a word X
$t_s(X)$	Start time of a word X
$t_E(X)$	End time of a word X
S-DB	Speaker pronunciation time database
S_p	p-th speaker
W_{pk}	k-th word of p-th speaker in S-DB

✉ Jin-Woo Jung
jwjung@dongguk.edu

¹ Department of Artificial Intelligence, Dongguk University, 30, Pildong-ro 1-gil, Jung-gu, Seoul 04620, Republic of Korea

² Department of Computer Science and Engineering, Dongguk University, 30, Pildong-ro 1-gil, Jung-gu, Seoul 04620, Republic of Korea

$U_p(W_{pk})$	Appearance frequency of k-th word of p-th speaker
$D_p(W_{pk})$	Average pronunciation time of k-th word of p-th speaker
$V(X)$	The number of characters in a word X
$D(S, X)$	Average pronunciation time of a word X of speaker S.

1 Introduction

In recent years artificial intelligence has come into wide use in various fields [1, 2, 7, 12, 13, 17, 19]. Artificial intelligence currently encompasses a huge variety of subfields, ranging from the general learning and perception to the specific, such as playing chess, proving mathematical theorems, writing poetry, driving a car on a crowded street, and diagnosing diseases. Artificial intelligence is relevant to any intellectual task, it is truly a universal field. [15] One of the areas that is actively researched is natural language processing by speech recognition. However, machines are difficult to speak, hear and read human language. Therefore, natural language processing and speech recognition are some of the most difficult and important field in artificial intelligence [6]. One of the most popular speech recognition technologies is the IBM Watson API [9]. Among captions in which speech is converted into characters, captions including timing and speaker ID information are called as Informatized Caption [3–5]. And, this Informatized Caption can be generated using the IBM Watson API [10]. However, the IBM Watson API is more susceptible to incorrect recognition when there is some noise in the audio signal, especially for movies where background music or special sounds are used. To solve this problem, many researches have been done. However, in the previous researches [3–5], there is still some problem that the speaker cannot be distinguished well when there is a plurality of speakers who can speak the same word differently [3], and an assumption that the database with speaker pronunciation time information should be ready in advance [4, 5]. In this paper, a method of modifying incorrectly recognized word using original caption is proposed to enhance the timing performance while updating the database in real time using the Informatized Caption information.

2 Background

2.1 Informatized caption

As shown in Fig. 1, IBM Watson API [10] makes a caption that includes not only the speech recognition result word but also the timing information (e.g. start time, end time, and speaker ID information) of each word. This kind of caption is called as an Informatized Caption. However, when speech recognition is performed on a noisy sound, there are many incorrectly recognized words. Therefore, the timing information in the caption is not correct.

2.2 Linear estimation method

The problem to modify the incorrectly recognized words and their timing information with the accurate original caption is the problem of estimating the timing information for each incorrectly recognized word. One of possible methods is the linear estimation

Fig. 1 The concept of Informatized Caption T_s^+

method based on the number of characters of words. For example, when there are two incorrectly recognized words B' and C' between the correctly recognized words A and D , the words B' and C' can be replaced by the correct words B and C by comparing with the original caption, and the timing information for B and C can be estimated by Eq. (1) where $V(X)$ means the number of characters of the word X . Here, it can be assumed that $t_S(C)$ is equal to $t_E(B)$ to reduce the number of uncertain variables.

$$t_E(B) = t_S(B) + (t_E(C) - t_S(B)) \times \frac{V(B)}{V(B) + V(C)} \quad (1)$$

where $V(X)$ means the number of characters in the word of Informatized Caption.

3 Speaker pronunciation time database (S-DB)

3.1 Database design

Figure 2 shows the model of speaker pronunciation time database, S-DB, which consists of nodes for each speaker S_p . The node is composed by word ID W_{px} , the average pronunciation time D_{px} , and its appearance frequency U_{px} . The nodes for each speaker are managed in ascending order based on the alphanumeric order of words, and are connected to each other by the linked list data structure with *NULL* node at the end.

3.2 Real-time update method of S-DB

The construction and update of S-DB are basically divided into three cases, the case that there are speaker S_p and word W_{pk} in S-DB, the case that there is speaker S_p in S-DB but not word W_{pk} in S-DB, and the case that there is not both speaker S_p and word W_{pk} in S-DB. The algorithm to update S-DB in real-time in each case is as follows.

Algorithm 1 Speaker pronunciation time database update

Inputs: Informatized Caption T_s^+ (word) (word $W^{\text{new}}[I, \dots, m]$, speaker_ID $S^{\text{new}}[I, \dots, m]$, $t_s^{\text{new}}[I, \dots, m]$, $t_E^{\text{new}}[I, \dots, m]$),
 S-DB (speaker_ID $S[I, \dots, n]$, word $W[I, \dots, n][\]$, average pronunciation time $D[I, \dots, n][\]$,
 appearance frequency $U[I, \dots, n][\]$),
// m means the number of words in Informatized Caption,
// n means the number of speakers in S-DB

S-DB_update() *// Algorithm to update S-DB*

```

1  for( $j \leftarrow 1, 2, \dots, m$ ) do
2     $D^{\text{new}}[j] \leftarrow t_E^{\text{new}}[j] - t_s^{\text{new}}[j]$ 
3    if(  $\exists$  a speaker_ID  $p \in S[\ ]$ , s.t.  $p = S^{\text{new}}[j]$ ) then
      // There exists some speaker_ID p in S-DB, which is equal to  $S^{\text{new}}[j]$ 
4      if(  $\exists$  a word index  $k$ , s.t.  $p = S^{\text{new}}[j]$  and  $W[p][k] = W^{\text{new}}[j]$ ) then
        // k means the number of words for p-th speaker in S-DB
        // There exists a word for the speaker_ID p in S-DB, which is equal to  $W^{\text{new}}[j]$ 
5         $D[p][k] \leftarrow D[p][k] \times (U[p][k]-1) / U[p][k] + D^{\text{new}}[j] / U[p][k]$ 
          // Update the average pronunciation time of k-th word for p-th speaker in S-DB
6         $U[p][k] \leftarrow U[p][k] + 1$     // Increase the appearance frequency
7      else
8         $k \leftarrow |W[p][\ ]|$     //  $|W[p][\ ]|$  means the number of words for speaker_ID p,
          // which is already registered in S-DB
9         $W[p][k] \leftarrow W^{\text{new}}[j]$     // Update the new word
10        $D[p][k] \leftarrow D^{\text{new}}[j]$     // Update the pronunciation time of new word
          // for p-th speaker in S-DB
11        $U[p][k] \leftarrow 1$     // Set the appearance frequency of new word
12       Append ( $S[p]$ ,  $W[p][k]$ ,  $D[p][k]$ ,  $U[p][k]$ ) to S-DB
          // Update the word list of p-th speaker of S-DB for the new word
13       Sort  $W[p][\ ]$     // Sort of the word list of p-th speaker of S-DB in alphanumeric order
14     else    // If there does not exist the speaker_ID  $S^{\text{new}}[j]$  in S-DB
15        $p \leftarrow n+1$     // The number of speakers is increased
16        $k \leftarrow 1$     // Set the word index for new speaker to 1
17        $S[p] \leftarrow S^{\text{new}}[j]$     // Update the new speaker_ID
18        $W[p][k] \leftarrow W^{\text{new}}[j]$     // Update the new word of new speaker
19        $D[p][k] \leftarrow D^{\text{new}}[j]$     // Update the pronunciation time of new word
          // for new speaker in S-DB
20        $U[p][k] \leftarrow 1$     // Set the appearance frequency to 1
21       Append ( $S[p]$ ,  $W[p][k]$ ,  $D[p][k]$ ,  $U[p][k]$ ) to S-DB
          // Update the word list of new speaker with the new word
22       Sort  $S[\ ]$     // Sort of the speaker list in alphanumeric order
23     end if
24  end for
```

Outputs: S-DB (speaker_ID $S[\]$, word $W[\][\]$, average pronunciation time $D[\][\]$, appearance frequency $U[\][\]$)

Fig. 2 S-DB Model

3.2.1 Case that there is speaker S_p and word W_{pk} in S-DB

In this cas, the first step is to find the corresponding word W_{pk} in the linked list for the speaker S_p in S-DB. And then, the appearance frequency U is incremented by 1 and the average pronunciation time D is updated by the Eq. (2).

$$D = D \times \frac{U-1}{U} + \frac{t_E(X)-t_S(X)}{U} \quad (2)$$

where $t_S(X)$ and $t_E(X)$ are the start and end time of pronunciation of word X , respectively.

3.2.2 Case that there is speaker S_p in S-DB, but not word W_{pk} in S-DB

In this case, the first step is to create a new node with the word W_{pk} in the linked list for the speaker S_p in S-DB. And then, 1 is assigned to the appearance frequency U of the new node and the pronunciation time of the word W_{pk} is updated for the initial average pronunciation time. Finally, insert sorting is performed for this linked list according to the alphanumeric order.

3.2.3 Case that there is not both speaker S_p and word W_{pk} in S-DB

To update new speaker in S-DB, the first step is to increment the number of speakers p by 1 and insert it to the existing speaker list of S-DB. And a new node is created in the linked list for the speaker S_p in S-DB. Then, 1 is assigned to the appearance frequency U of the new node and the pronunciation time of the word W_{pk} is updated for the initial average pronunciation

Fig. 3 The overall structure of the proposed system

time. Finally, insert sorting is performed for the speaker list according to the alphanumeric order.

4 Proposed algorithm

4.1 Algorithm to modify incorrectly recognized word and timing information using S-DB

Figure 3 shows the overall structure of the proposed system. When a user speaks, the speech is translated into a type of informatized caption by special Speech-to-Text engine such as IBM Watson which can generate not only the text information but also user and pronunciation timing information. Then, the timing information of this informatized caption is modified through the proposed algorithm with the original caption. Specifically, the inputs of the proposed algorithm are original caption, Informatized Caption, discrete voice signal, S-DB (e.g. Speaker ID, Word, Pronunciation time, and appearance frequency), the threshold value to recognize the start time and end time of a word in the discrete voice signal, and the search range in Informatized Caption to find the matched word with original caption and to modify the incorrectly recognized word and its timing information. The output of the proposed algorithm is modified Informatized Caption and its timing information. Here, S-DB updates are occurred only when there is no incorrectly recognized word. If an incorrectly recognized word is found between correct recognized words, the number of incorrectly recognized words is counted. And, the wrong words are replaced by the original caption, and the timing information is calculated as Fig. 4 or Algorithm 2 for each case.

Algorithm 2 Algorithm to modify incorrectly recognized word and timing information using S-DB

Inputs: Original caption $T(\text{word})$ (word $W^{\text{orig}}[1, \dots, l]$),
 Informatized Caption $T_s(\text{word})$ (word $W^{\text{new}}[1, \dots, m]$, speaker_ID $S^{\text{new}}[1, \dots, m]$, $t_s^{\text{new}}[1, \dots, m]$, $t_e^{\text{new}}[1, \dots, m]$),
 Discrete_Voice[$t_{\text{start}}, \dots, t_{\text{last}}$],
 S-DB (speaker_ID $S[1, \dots, n]$, word $W[1, \dots, n]$), average pronunciation time $D[1, \dots, n]$,
 appearance frequency $U[1, \dots, n]$,
 Threshold T ,
 Offset offset,
 // m means the number of words in Informatized Caption
 // l means the number of words in original caption
 // n means the number of speakers in S-DB
 // T means the threshold value to recognize the start time and end time of a word in the discrete voice signal
 // offset means the search range in Informatized Caption to find the matched word with original caption

Modify_Inform_Caption() // Algorithm to modify incorrectly recognized word and its timing

```

1   $q \leftarrow 0$  //  $q$  means the number of incorrectly recognized words
2   $i^* \leftarrow 0$  //  $i^*$  means the index of the most recently matched word in inform_IDX
3   $k^* \leftarrow 0$  //  $k^*$  means the index of the most recently matched word in org_IDX
4  for(org_IDX  $\leftarrow 1, \dots, l$ ) do
    // org_IDX means the index of currently processing word in the original caption
5    for(inform_IDX  $\leftarrow \max(1, i^* - \text{offset}), \dots, \min(m, \text{org\_IDX} + \text{offset})$ ) do
      //inform_IDX means the index of currently processing word in the Informatized Caption
6      if  $W^{\text{orig}}[\text{org\_IDX}] = W^{\text{new}}[\text{inform\_IDX}]$  then
        // If a pair of words in original caption and Informatized Caption are the same
7        if  $((i^* + 1 < \text{inform\_IDX} - 1) \ \& \ (k^* + 1 < \text{org\_IDX} - 1))$  then
          // If both  $i^* + 1$  and  $k^* + 1$  are not the last word index
8           $W^{\text{new}}[i^* + 1, \dots, \text{inform\_IDX} - 1] \leftarrow W^{\text{orig}}[k^* + 1, \dots, \text{org\_IDX} - 1]$ 
          // modify all the incorrectly recognized words which are located
          // between previously matched word and currently matched word
          // For the real implementation, linked list is more feasible by the different length
9           $i^* \leftarrow \text{inform\_IDX}$  // Update the index  $i^*$  with newly matched word
10          $k^* \leftarrow \text{org\_IDX}$  // Update the index  $k^*$  with newly matched word
11         S-DB_update( $T_s(W^{\text{new}}[i^*])$ , S-DB) // Update S-DB with newly matched word
12         if  $(q = 1)$  then // Case that the number of incorrectly recognized words is only one
13           Case1( $T_s()$ , Discrete_Voice[],  $T$ , inform_IDX)
14            $q \leftarrow 0$  // Reset the number of incorrectly recognized words
15         end if
16         if  $(q = 2)$  then // Case that the number of incorrectly recognized words are two
17           Case2( $T_s()$ , Discrete_Voice[],  $T$ , inform_IDX, S-DB)
18            $q \leftarrow 0$  // Reset the number of incorrectly recognized words
19         end if
20         if  $(q \geq 3)$  then // Case that the number of
          // incorrectly recognized words are more than three
21           Case3( $T_s()$ , Discrete_Voice[],  $T$ ,  $q$ , inform_IDX, S-DB)
22            $q \leftarrow 0$  // Reset the number of incorrectly recognized words
23         end if
24       end if
25       break // Search for the next word in the original caption
26     end if
27     if  $(\text{inform\_IDX} = \min(m, \text{org\_IDX} + \text{offset}))$  then
      // when there is no matched word in the searching range with a word in original caption
28      $q \leftarrow q + 1$  // increase the number of incorrectly recognized words
29     end if
30   end for
31 end for
32  $W^{\text{new}}[i^* + 1, \dots, m] \leftarrow W^{\text{orig}}[k^* + 1, \dots, l]$ 
  // Modify the last remaining incorrectly recognized words
33 if  $(q = 1)$  then // Case that the number of incorrectly recognized words is only one
34   Case1( $T_s()$ , Discrete_Voice[],  $T$ , inform_IDX)
35 end if
36 if  $(q = 2)$  then // Case that the number of incorrectly recognized words are two
37   Case2( $T_s()$ , Discrete_Voice[],  $T$ , inform_IDX, S-DB)
38 end if
39 if  $(q \geq 3)$  then // Case that the number of incorrectly recognized words are more than three
40   Case3( $T_s()$ , Discrete_Voice[],  $T$ ,  $q$ , inform_IDX, S-DB)
41 end if

```

Outputs: Informatized Caption $T_s(\text{word})$ (word $W^{\text{new}}[1, \dots, m]$, speaker_ID $S^{\text{new}}[1, \dots, m]$, $t_s^{\text{new}}[1, \dots, m]$, $t_e^{\text{new}}[1, \dots, m]$)

4.2 Case 0: No incorrectly recognized word

In this case, the final output is the same with the output from IBM Watson API. Therefore, Informatized Caption is used just to update S-DB.

4.3 Case 1: The number of incorrectly recognized words is only one

Algorithm 3 Case that the number of incorrectly recognized words is only one

Inputs: Informatized Caption $T_s^+(\text{word})$ (word $W^{\text{new}}[1, \dots, m]$, speaker_ID $S^{\text{new}}[1, \dots, m]$, $t_s^{\text{new}}[1, \dots, m]$, $t_e^{\text{new}}[1, \dots, m]$),
 Discrete_Voice[$t_{\text{START}}, \dots, t_{\text{LAST}}$],
 Threshold T ,
 c_IDX // c_IDX means the index of currently processing word in the Informatized Caption

Case1()

```

1   $IDX \leftarrow c\_IDX - 1$  //  $IDX$  means the index of incorrectly recognized word
2   $t_s^{\text{new}}[IDX] \leftarrow t_e^{\text{new}}[IDX - 1]$  // Assign the end time of previous word to
   // the start time of incorrectly recognized word
3   $t_e^{\text{new}}[c\_IDX - 1] \leftarrow t_s^{\text{new}}[c\_IDX]$  // Assign the start time of currently processing word to
   // the end time of incorrectly recognized word
4  for( $k \leftarrow t_e^{\text{new}}[IDX - 1] + 1, \dots, t_s^{\text{new}}[c\_IDX - 1]$ )
5    if(Discrete_Voice[ $k$ ]  $\geq T$ ) // find a minimum time index  $k$  above
   // the threshold value  $T$  in the forward direction
6       $t_s^{\text{new}}[IDX] \leftarrow k$ 
7      break
8    end if
9  end for
10 for( $k \leftarrow t_s^{\text{new}}[c\_IDX - 1], \dots, t_e^{\text{new}}[IDX - 1] + 1$ )
11   if(Discrete_Voice[ $k$ ]  $\geq T$ ) // find a maximum time index  $k$  above
   // the threshold value  $T$  in the backward direction
12      $t_e^{\text{new}}[c\_IDX - 1] \leftarrow k$ 
13     break
14   end if
15 end for
```

Outputs: Informatized Caption $T_s^+(\text{word})$ (word $W^{\text{new}}[1, \dots, m]$, speaker_ID $S^{\text{new}}[1, \dots, m]$, $t_s^{\text{new}}[1, \dots, m]$, $t_e^{\text{new}}[1, \dots, m]$)

In the case of Fig. 5, the number of incorrectly recognized words is only one. And, the required function is just to replace the wrong word B with the correct one and to find its timing information. If there is the minimum time t_{MIN} between $t_e(A)$ to $t_s(C)$ whose value of discrete voice signal is more than the threshold T , it can be assumed as the start time of the correct word B . Otherwise, the start time of the correct word B is assumed as the same with the end time of the previous word $t_e(A)$. Likewise, if there is the maximum time t_{MAX} between $t_e(A)$ to $t_s(C)$ whose value of discrete voice signal is more than the threshold T , it can be assumed as the end time of the correct word B . Otherwise, the end time of the correct word B is assumed as the same with the start time of the next word $t_s(C)$.

4.4 Case 2: The number of incorrectly recognized words are two

Algorithm 4 Case that the number of incorrectly recognized words are two

Inputs: Informatized Caption $T_s^*(\text{word})$ (word $W^{\text{new}}[1, \dots, m]$, speaker_ID $S^{\text{new}}[1, \dots, m]$, $t_s^{\text{new}}[1, \dots, m]$, $t_e^{\text{new}}[1, \dots, m]$),
 Discrete_Voice[$t_{\text{START}}, \dots, t_{\text{LAST}}$],
 Threshold T ,
 $c_IDX(c_IDX$ means the index of currently processing word in the Informatized Caption), S-DB (speaker_ID $S[1, \dots, n]$, word $W[1, \dots, n]$ [], average pronunciation time $D[1, \dots, n]$ [], appearance frequency $U[1, \dots, n]$ [])

Case2()

```

1   $IDX \leftarrow c\_IDX - 2$  //  $IDX$  means the start index of incorrectly recognized words
2   $t_s^{\text{new}}[IDX] \leftarrow t_e^{\text{new}}[IDX - 1]$  // Assign the end time of previous matched word to
   // the start time of incorrectly recognized words
3   $t_e^{\text{new}}[c\_IDX - 1] \leftarrow t_s^{\text{new}}[c\_IDX]$  // Assign the start time of currently processing word to
4  // the end time of incorrectly recognized words
5  for( $k \leftarrow t_e^{\text{new}}[IDX - 1] + 1, \dots, t_s^{\text{new}}[c\_IDX - 1]$ )
6    if(Discrete_Voice[ $k$ ]  $\geq T$ ) // find a minimum time index  $k$  above
   // the threshold value  $T$  in the forward direction
7       $t_s^{\text{new}}[IDX] \leftarrow k$ 
8      break
9    end if
10 end for
11 for( $k \leftarrow t_s^{\text{new}}[c\_IDX - 1], \dots, t_e^{\text{new}}[IDX - 1] + 1$ )
12   if(Discrete_Voice[ $k$ ]  $\geq T$ ) // find a maximum time index  $k$  above
   // the threshold value in the backward direction
13      $t_e^{\text{new}}[c\_IDX - 1] \leftarrow k$ 
14     break
15   end if
16 end for
17  $\text{indP1} \leftarrow$  Speaker index of S-DB, which is equal to  $S^{\text{new}}[IDX]$ 
18  $\text{indK1} \leftarrow$  Word index for the speaker ID  $\text{indP1}$  in S-DB, which is equal to  $W^{\text{new}}[IDX]$ 
19  $\text{indP2} \leftarrow$  Speaker index of S-DB, which is equal to  $S^{\text{new}}[IDX + 1]$ 
20  $\text{indK2} \leftarrow$  Word index for the speaker ID  $\text{indP2}$  in S-DB, which is equal to  $W^{\text{new}}[IDX + 1]$ 
21 if(( $S^{\text{new}}[IDX], W^{\text{new}}[IDX]$ )  $\in$  S-DB)  $\wedge$  (( $S^{\text{new}}[IDX + 1], W^{\text{new}}[IDX + 1]$ )  $\in$  S-DB))
   // Both the first and the second incorrectly recognized words exist in S-DB
22    $t_e^{\text{new}}[IDX] \leftarrow t_s^{\text{new}}[IDX] + (t_e^{\text{new}}[IDX + 1] - t_s^{\text{new}}[IDX]) \times D[\text{indP1}][\text{indK1}] /$ 
   ( $D[\text{indP1}][\text{indK1}] + D[\text{indP2}][\text{indK2}]$ )
23    $t_s^{\text{new}}[IDX + 1] \leftarrow t_e^{\text{new}}[IDX]$ 
24 end if
25 if(( $S^{\text{new}}[IDX], W^{\text{new}}[IDX]$ )  $\in$  S-DB)  $\wedge$  (( $S^{\text{new}}[IDX + 1], W^{\text{new}}[IDX + 1]$ )  $\notin$  S-DB))
   // Only the first incorrectly recognized word exists in S-DB
26    $t_e^{\text{new}}[IDX] \leftarrow t_s^{\text{new}}[IDX] + D[\text{indP1}][\text{indK1}]$ 
27    $t_s^{\text{new}}[IDX + 1] \leftarrow t_e^{\text{new}}[IDX]$ 
28 end if
29 if(( $S^{\text{new}}[IDX], W^{\text{new}}[IDX]$ )  $\notin$  S-DB)  $\wedge$  (( $S^{\text{new}}[IDX + 1], W^{\text{new}}[IDX + 1]$ )  $\in$  S-DB))
   // Only the second incorrectly recognized word exists in S-DB
30    $t_e^{\text{new}}[IDX] \leftarrow t_e^{\text{new}}[IDX + 1] - D[\text{indP2}][\text{indK2}]$ 
31    $t_s^{\text{new}}[IDX + 1] \leftarrow t_e^{\text{new}}[IDX]$ 
32 end if
33 if(( $S^{\text{new}}[IDX], W^{\text{new}}[IDX]$ )  $\notin$  S-DB)  $\wedge$  (( $S^{\text{new}}[IDX + 1], W^{\text{new}}[IDX + 1]$ )  $\notin$  S-DB))
   // Both the first and the second incorrectly recognized words do not exist in S-DB
34    $t_e^{\text{new}}[IDX] \leftarrow t_s^{\text{new}}[IDX] + (t_e^{\text{new}}[IDX + 1] - t_s^{\text{new}}[IDX]) \times V^{\text{new}}[IDX] / (V^{\text{new}}[IDX] +$ 
    $V^{\text{new}}[IDX + 1])$ 
   //  $V^{\text{new}}[q]$  means the number of characters in the word of Informatized Caption
35    $t_s^{\text{new}}[IDX + 1] \leftarrow t_e^{\text{new}}[IDX]$ 
36 end if

```

Outputs: Informatized Caption $T_s^*(\text{word})$ (word $W^{\text{new}}[1, \dots, m]$, speaker_ID $S^{\text{new}}[1, \dots, m]$, $t_s^{\text{new}}[1, \dots, m]$, $t_e^{\text{new}}[1, \dots, m]$)

In the case of Fig. 6, there are two consecutive and incorrectly recognized words B and C between the correct words A and D . Therefore, $t_S(B)$ and $t_E(C)$ can be estimated by using the same algorithm in case 1 (Algorithm 3). Here, $t_E(B)$ is assumed as the same with $t_S(C)$ for the simplicity of problem. And, the timing information of $t_E(B)$ or $t_S(C)$ can be estimated by using the linear estimation method or by using the previous pronunciation time information of the same speaker, $D(S, B)$ and $D(S, C)$ in the S-DB. If there is not both $D(S, B)$ and $D(S, C)$ in S-DB, i.e. pronunciation time of words B and C are unknown, linear estimation method is used such as Eq. (3).

$$t_E(B) = t_S(B) + (t_E(C) - t_S(B)) \times \frac{V(B)}{V(B) + V(C)} \quad (3)$$

where $V(X)$ means the number of characters in the word of Informatized Caption.

If there is only $D(S, B)$ in S-DB, $t_E(B)$ is estimated by adding the average pronunciation time $D(S, B)$ to $t_S(B)$ such as Eq. (4).

$$t_E(B) = t_S(B) + D(S, B) \quad (4)$$

If there is only $D(S, C)$ in S-DB, $t_E(B)$ is estimated by subtracting the average pronunciation time $D(S, C)$ from $t_E(C)$ such as Eq. (5).

$$t_E(B) = t_S(B) + (t_E(C) - t_S(B) - D(S, C)) = t_E(C) - D(S, C) \quad (5)$$

If there is both $D(S, B)$ and $D(S, C)$ in S-DB, Eq. (6) using the ratio of $D(S, B)$ and $D(S, C)$ is used.

$$t_E(B) = t_S(B) + (t_E(C) - t_S(B)) \times \frac{D(S, B)}{D(S, B) + D(S, C)} \quad (6)$$

Finally, $t_S(C)$ is estimated by using $t_E(B)$.

4.5 Case 3: The number of incorrectly recognized words are more than three

Algorithm 5 Case that the number of incorrectly words are more than three	
Inputs: Informalized Caption $T_s^*(\text{word})$ (word $W^{\text{new}}[1, \dots, m]$, speaker_ID $S^{\text{new}}[1, \dots, m]$, $t_s^{\text{new}}[1, \dots, m]$, $t_e^{\text{new}}[1, \dots, m]$), Discrete_Voice $\{t_{\text{START}}, \dots, t_{\text{LAST}}\}$, Threshold T , in_IDX //in_IDX means the number of incorrectly recognized words), c_IDX // c_IDX means the index of currently processing word in the Informalized Caption), S-DB (speaker_ID $S[1, \dots, n]$, word $W[1, \dots, n][i]$, average pronunciation time $D[1, \dots, n][i]$, appearance frequency $U[1, \dots, n][i]$)	
1	Case3()
2	idx $\leftarrow c_IDX - in_IDX$
3	sumV $\leftarrow 0$ // sumV means the summation of number of characters in all incorrectly // recognized words of Informalized Caption which is not in S-DB
4	sumD $\leftarrow 0$ // sumD means the summation of pronunciation time in all incorrectly // recognized words of Informalized Caption which is in S-DB
5	$t_s^{\text{new}}[idx] \leftarrow t_e^{\text{new}}[idx - 1]$
6	$t_e^{\text{new}}[c_IDX - 1] \leftarrow t_s^{\text{new}}[c_IDX]$
7	for ($k \leftarrow t_s^{\text{new}}[idx - 1] + 1, \dots, t_s^{\text{new}}[c_IDX - 1]$)
8	if (Discrete_Voice $[k] \geq T$) // find a minimum time index k above // the threshold value T in the forward direction
9	$t_s^{\text{new}}[idx] \leftarrow k$
10	break
11	end if
12	end for
13	for ($k \leftarrow t_s^{\text{new}}[c_IDX - 1], \dots, t_e^{\text{new}}[idx - 1] + 1$)
14	if (Discrete_Voice $[k] \geq T$) // find a maximum time index k above // the threshold value T in the backward direction
15	$t_e^{\text{new}}[c_IDX - 1] \leftarrow k$
16	break
17	end if
18	end for
19	for ($q \leftarrow idx, \dots, c_IDX - 1$) // for all incorrectly recognized word indexes (1 st scanning)
20	if ($\exists$ a speaker_ID $p \in S$], s.t. $p = S^{\text{new}}[j]$) then // If there exists some speaker_ID p in S-DB, which is equal to $S^{\text{new}}[j]$
21	if ($\exists$ a word index q , s.t. $q = S^{\text{new}}[j]$ and $W[p][q] = W^{\text{new}}[j]$) then // If there exists a word for the speaker_ID p in S-DB, which is equal to $W^{\text{new}}[j]$ // Here, q means the number of words for p-th speaker in S-DB
22	sumD $\leftarrow$ sumD + D[p][q]
23	else // For the case that the speaker is included in S-DB but not word in S-DB
24	sumV $\leftarrow$ sumV + V[q] // V[q] means the number of characters in the word of Informalized Caption
25	end if
26	else // For the case that the speaker is not included in S-DB
27	sumV $\leftarrow$ sumV + V[q]
28	end if
29	end for
30	for ($q \leftarrow idx + 1, \dots, c_IDX - 1$) // for the incorrectly recognized word indexes (2 nd scanning)
31	if ($\exists$ a speaker_ID $p \in S$], s.t. $p = S^{\text{new}}[j]$) then // If there exists some speaker_ID p in S-DB, which is equal to $S^{\text{new}}[j]$
32	if ($\exists$ a word index q , s.t. $q = S^{\text{new}}[j]$ and $W[p][q] = W^{\text{new}}[j]$) then // If there exists a word for the speaker_ID p in S-DB, which is equal to $W^{\text{new}}[j]$ // Here, q means the number of words for p-th speaker in S-DB
33	$t_s^{\text{new}}[q] \leftarrow t_s^{\text{new}}[q - 1] + D[p][q]$
34	$t_e^{\text{new}}[q - 1] \leftarrow t_s^{\text{new}}[q]$
35	else
36	$t_s^{\text{new}}[q] \leftarrow (t_e^{\text{new}}[c_IDX - 1] - t_s^{\text{new}}[idx] - \text{sumD}) / \text{sumV} \times V[q]$
37	$t_e^{\text{new}}[q - 1] \leftarrow t_s^{\text{new}}[q]$
38	end if
39	else
40	$t_s^{\text{new}}[q] \leftarrow (t_e^{\text{new}}[c_IDX - 1] - t_s^{\text{new}}[idx] - \text{sumD}) / \text{sumV} \times V[q]$
41	$t_e^{\text{new}}[q - 1] \leftarrow t_s^{\text{new}}[q]$
42	end if
43	end for
Outputs: Informalized Caption $T_s^*(\text{word})$ (word $W^{\text{new}}[1, \dots, m]$, speaker_ID $S^{\text{new}}[1, \dots, m]$, $t_s^{\text{new}}[1, \dots, m]$, $t_e^{\text{new}}[1, \dots, m]$)

Fig. 6 A segment of Informatized Caption which has two consecutive and incorrectly recognized words

$$t_S(W_{q+1}) = t_S(W_q) + D(S, W_q) \quad (8)$$

The above equations can also be applied to calculate $t_E(W_1)$, $t_S(W_2)$, $t_E(W_2)$, ..., $t_S(W_l)$.

4.6 Mathematical validation of the convergence of average pronunciation time as S-DB update

The pronunciation time of a word is generally varying by each person, his/her average speaking speed, and connecting word near it. For example, different persons can make different voice and different pronunciation time for the same word. Even for the same person, he/she can be hurry or relaxed when speaking. In addition, the same word spoken by the same person may have different pronunciation time by the connecting word which is located before or after the word such as soft consonant phenomenon. In this paper, we have assumed two assumptions to simplify the problem.

Assumption 1. The pronunciation time of a word is independent with the connecting words.

Assumption 2. The pronunciation time of a word is an independent and identically distributed (i. i. d.) random variable drawn from a distribution of expected value m and finite variance σ^2 .

Theorem 1. As the numbers of update in S-DB is increased with more pronunciations of the same word by the same speaker, the average pronunciation time of the word by the speaker (D value in S-DB) will be converged.

(Proof) Let us assume that there was already $U(W_{ij}) - 1$ numbers of update in S-DB with the average pronunciation time $D_i(W_{ij})$ of a word W_{ij} by a speaker S_i .

$$D_i(W_{ij}) = \frac{X(W_{ij}, 1) + X(W_{ij}, 2) + X(W_{ij}, 3) + \cdots + X(W_{ij}, n)}{n} \quad (9)$$

Fig. 7 A segment of Informatized Caption which has more than three consecutive and incorrectly recognized words

where $n = U(W_{ij}) - 1$ and $X(W_{ij}, n)$ is the n -th pronunciation time of the word W_{ij} by a speaker S_i .

Then, the next update with the current pronunciation time $D(W_{ij})$ of the word W_{ij} by a speaker S_i is done by the following rule:

$$D_i^{new}(W_{ij}) = D_i(W_{ij}) \times \frac{U(W_{ij})-1}{U(W_{ij})} + \frac{\text{End time}-\text{Start time}}{U(W_{ij})} \quad (10)$$

$$= D_i(W_{ij}) \times \frac{U(W_{ij})-1}{U(W_{ij})} + \frac{X(W_{ij}, U(W_{ij}))}{U(W_{ij})} \quad (11)$$

$$= \frac{X(W_{ij}, 1) + X(W_{ij}, 2) + X(W_{ij}, 3) + \cdots + X(W_{ij}, U(W_{ij}))}{U(W_{ij})} \quad (12)$$

By the Central Limit Theorem [11, 14], the sampling distribution of the sample means, $D_i^{new}(W_{ij})$, approaches to a normal distribution with the mean m and the variance $\frac{\sigma^2}{U(W_{ij})}$ as the sample size $U(W_{ij})$ gets larger regardless of the original shape of population distribution. As the numbers of update $U(W_{ij})$ in S-DB approaches to the infinity, the variance $\frac{\sigma^2}{U(W_{ij})}$ approaches to zero. Therefore, the pronunciation time $D(W_{ij})$ of the word W_{ij} by the speaker S_i is converged to m which is the expected value of $D(W_{ij})$. (Q.E.D.)

5 Experimental results

5.1 Experiment setup

There are lots of factors that reduce accuracy in speech recognition. Among many factors, the most critical one is noise. 7 types of noise have been used for experiments as Fig. 8. 6 colors of noise [16], gray noise, purple noise, blue noise, brown noise, pink noise and white noise, have been used for the first experiment, which is to show how different types of noise affect the proposed method. A kind of typical examples of white noise, rainy noise which was captured from rainy sound, has been used for the second experiment, which is to show how different levels(decibels) of noise affect the proposed method. The last experiment is to show whether the proposed method can work with various movie samples or not. In this last experiment, each movie sample includes natural noise basically. Therefore, any additional noise did not add.

The problem to modify the incorrectly recognized words and their timing information with the accurate original caption is the problem of estimating the timing information for each incorrectly recognized word. The correct timing information of each word in each test audio file was measured by checking the timing of each word during playing the audio file based on the wave display panel of Adobe CC 2019 software. The measurement values were represented with 2 places of decimals in seconds. To verify the accuracy of timing information from

Fig. 8 Various types of noise (a) white noise, (b) pink noise, (c) brown noise, (d) blue noise, (e) purple noise, (f) gray noise, (g) rainy noise, and (h) an example of speech signal

Informatized Caption, three kinds of experiments are performed. The measure of accuracy in the three experiments is like Eq. (13).

$$Accuracy = \frac{|\{x | x \in CRW \text{ and } |t_S(x) - t_S^*| < 0.01 \text{ and } |t_E(x) - t_E^*| < 0.01\}|}{Total \text{ number of test words}} \times 100 \quad (13)$$

In Equation (13), where CRW means the set of correctly recognized words, $|set A|$ means the number of elements in the set A , t_S^* and t_E^* means the correct start and end time of the word X

Table 1 Information on 4 test audio files used in the experiment 1 and 2

Sample ID	Test audio	Extracted time period	Number of total words
Sample A	English listening test 1	00:00:00–00:00:15	41
Sample B	English listening test 2	00:00:00–00:00:14	35
Sample C	English listening test 3	00:00:00–00:00:23	47
Sample D	English listening test 4	00:00:00–00:00:14	37

Fig. 9 Test data samples for experiment 1, (a) original speech signal, (b) speech signal with white noise ($\text{SNR}_{\text{dB}} = 20$ dB), (c) speech signal with white noise ($\text{SNR}_{\text{dB}} = 15$ dB), (d) speech signal with white noise ($\text{SNR}_{\text{dB}} = 10$ dB), and (e) speech signal with white noise ($\text{SNR}_{\text{dB}} = 5$ dB)

(timing unit: seconds). The first and second experiment is done with a clean sound such as an English listening test audio [8] file which is recorded in a well-structured noise-free environment. Detailed information for each test audio file is shown in Table 1. Each of the test files is mixed with noise whose signal-to-noise ratio [18] is 20, 15, 10 and 5(dB) as shown in Fig. 9. The first experiment is based on the colors of noise. The six noises were synthesized according to different levels of signal-to-noise ratio to make a total of 96 samples. The second experiment is done with a total of 16 samples based on the sound of rain, a typical example of white noise that can be heard in real life.

The last experiment was for the performance in various real noisy environments and done with three genres of films for the accuracy test of timing information from Informatized Caption after speech recognition using IBM Watson API. The genre of films consists of 5 animation movies dubbing in a well-structured noise-free environment, 5 horror and action movies with various sound effects, and 5 musical movies with many background music. The sample selection conditions for each film genre are as follows. Animation chose a noise-free sample. Horror and action chose samples with a noise like staccato. Musical chose a constant sample of noise like background music. The time periods extracted for this experiment were 6 to 20 s which reflect the

Table 2 Information on movie sample data by genre for the experiment 3

Sample ID	Genre	Title	Extracted time period	Number of total words
Sample 1	Animation	The Complete (season 3 episode 11)	00:02:30–00:02:37	21
Sample 2	Animation	Disenchantment (season 1 episode 3)	00:20:07–00:20:17	24
Sample 3	Animation	The Boss Baby: Back in Business season 2	00:11:21–00:11:27	8
Sample 4	Animation	Big Hero 6	00:27:47–00:27:53	17
Sample 5	Animation	Rick and morty (season 2 episode 8)	00:05:32–00:05:41	22
Sample 6	Horror and Action	The equalizer	00:50:17–00:50:29	18
Sample 7	Horror and Action	Inception	01:05:03–01:05:14	35
Sample 8	Horror and Action	Lucy	01:09:30–01:09:54	39
Sample 9	Horror and Action	Matrix	01:52:41–01:52:55	12
Sample 10	Horror and Action	Mission Impossible: Rogue Nation	01:45:47–01:46:02	16
Sample 11	Musical	Les Misérables	00:43:19–00:43:43	40
Sample 12	Musical	Been So Long	00:59:32–00:59:53	23
Sample 13	Musical	Christmas Wedding Planner	00:57:30–00:57:41	35
Sample 14	Musical	Come Sunday	00:49:24–00:49:52	20
Sample 15	Musical	Yes-Man	01:06:18–01:06:45	45

Fig. 10 Accuracy of timing information according to colors of noise, (a) Experimental result for Sample A, (b) Experimental result for Sample B, (c) Experimental result for Sample C, and (d) Experimental result for Sample D

characteristics of each genre. Detailed information for each movie is shown in Table 2.

Fig. 11 Accuracy of timing information according to varying intensity of noises, (a) Experimental result for Sample A, (b) Experimental result for Sample B, (c) Experimental result for Sample C, and (d) Experimental result for Sample D

Table 3 Timing information accuracy of Informatized Caption in animation movies

	Number of total words	IBM Watson API		Proposed Algorithm	
		Number of words with incorrect timing	Accuracy (%)	Number of words with incorrect timing	Accuracy (%)
Sample 1	21	6	71.42	2	90.47
Sample 2	24	11	54.16	0	100.00
Sample 3	8	2	75.00	0	100.00
Sample 4	17	2	88.23	0	100.00
Sample 5	22	21	4.54	19	13.63
Average	18.4	8.4	58.67	4.2	80.82

5.2 Experimental results

5.2.1 Experiment 1: Samples with different colors of noise

Figure 10 shows the results of accuracy of the proposed algorithm according to different levels of each colors of noise. The vertical axis shows signal-to-noise ratio and the horizontal axis shows timing accuracy. The average accuracy of each experiment is 73.10% for white noise, 80.80% for pink noise, 81.20% for brown noise, 79.71% for blue noise, 79.96% for purple noise, and 81.60% for gray noise. Regardless of the levels of noise, the accuracy of white noise was the lowest. Since the proposed algorithm uses the results of speech recognition, the result of the proposed algorithm may be less accurate when the noise level is too much high and similar to the voice level.

5.2.2 Experiment 2: A sample with different noises

Figure 11 shows the result of experiments with 4 test audio files after applying 4 different noises to each sample. The vertical axis represents the accuracy and the horizontal axis represents the signal-to-noise ratio of mixed audio files with varying intensity of noises. Experimental results show that the accuracy of timing information of Informatized Caption by IBM Watson API is more than 84.80% for English listening test audio files when noise is small. And, the accuracy decreases significantly as noise increases. The timing accuracy of proposed algorithm shows more than 95.70% for the same English listening test audio files. However, the accuracy of the proposed algorithm is also lowered when the noise level is very high and similar to the voice level since the proposed algorithm compensates the result of IBM Watson API.

Table 4 Timing information accuracy of Informatized Caption in horror and action movies

	Number of total words	IBM Watson API		Proposed Algorithm	
		Number of words with incorrect timing	Accuracy (%)	Number of words with incorrect timing	Accuracy (%)
Sample 6	18	14	22.22	7	61.11
Sample 7	35	27	22.85	19	45.71
Sample 8	39	4	89.74	0	100.00
Sample 9	12	5	58.33	0	100.00
Sample 10	16	8	50.00	0	100.00
Average	24	11.6	48.63	5.2	81.36

Table 5 Timing information accuracy of Informatized Caption in musical movies

	Number of total words	IBM Watson API		Proposed Algorithm	
		Number of words with incorrect timing	Accuracy (%)	Number of words with incorrect timing	Accuracy (%)
Sample 11	40	29	27.50	25	37.50
Sample 12	23	17	26.08	10	43.47
Sample 13	35	29	17.14	21	40.00
Sample 14	20	17	15.00	14	30.00
Sample 15	45	35	22.22	30	33.33
Average	32.6	25.4	21.56	20	36.86

5.2.3 Experiment 3: Various movie samples

The accuracy of timing information of each sample movie for each genre is shown in Table 3 (Animation movies), Table 4 (Horror and action movies), and Table 5 (Musical movies). The average accuracy of IBM Watson method was 42.95% for all movies and 53.65% for animation, horror and action movies. Since most of time period in musical movies include background music and singing voices, the performance of speech recognition by IBM Watson is basically very low and consequently, the accuracy for musical movies are the lowest. The average accuracy of the proposed method is 66.35% for all movies and 81.09% for animation, horror and action movies. The proposed method shows more than 1.5 times of performance compared with the IBM Watson method.

6 Concluding remarks

IBM Watson API is one of the most popular speech recognition technologies in these days. It can perform not only the translation process from speech to text, but also give some useful data about word timing and speaker ID information which are called as Informatized Caption. However, the IBM Watson API is very weak for the noisy audios and consequently, it is not good for the use in movie audios where background music or special sound effects are mixed together.

In this paper, a novel method of modifying incorrectly recognized word using original caption is proposed to enhance the timing performance of Informatized Caption while updating S-DB database in real time. Mathematical validation is done for the convergence of average pronunciation time in S-DB as S-DB update continually. Experimental results show that improved performance on speech recognition can be achieved in the presence of various types of noise. And three kinds of experiments have been performed to verify the accuracy of timing information by the proposed method. The average accuracy of the proposed method is 81.09% for animation, horror and action movies and 66.35% for all movies. And this result means that the proposed method shows more than 1.5 times of performance compared with the original IBM Watson method.

Acknowledgments This research was partially supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2020-2020-0-01789) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation), the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2020R1F1A1074974), the KIAT(Korea Institute for Advancement of Technology) grant funded by the Korea Government(MOTIE :

Ministry of Trade Industry and Energy). (No. N0001884, HRD program for Embedded Software R&D), the AURI(Korea Association of University, Research institute and Industry) grant funded by the Korea Government(MSS : Ministry of SMEs and Startups). (No.S2938281, HRD program for Enterprise linkages R&D), the MSIT(Ministry of Science and ICT), Korea, under the National Program for Excellence in SW supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation)(2016-0-00017).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Alsamhi SH, Ma O, Ansari MS (2018) Artificial intelligence-based techniques for emerging robotics communication: a survey and future perspectives. arXiv preprint arXiv:1804.09671
2. Ban F, Wu D, Hei Y (2018) Combined forecasting model of urban water consumption based on adaptive filtering and BP neural network. *International Journal of Social and Humanistic Computing* 3(1):34–45. <https://doi.org/10.1504/IJSHC.2018.095011>
3. Choi YS, Park HM, Son YS, Jung JW (2017) Informatized caption enhancement based on IBM Watson API. *Proceedings of KIIS Autumn Conference* 27(2):105–106
4. Choi YS, Son YS, Jung JW (2018) Informatized caption enhancement based on IBM Watson API and speaker pronunciation time-DB. *Computer Science & Information Technology – computer science conference proceedings* :105–110
5. Choi YS, Son YS, Jung JW (2018) A method to enhance Informatized caption from IBM Watson API using speaker pronunciation time-DB. *International Journal on Natural Language Computing* 7(1):1–11
6. Chowdhury GG (2003) Natural language processing. *Annu Rev Inf Sci Technol* 37(1):51–89. <https://doi.org/10.1002/aris.1440370103>
7. Drigas AS, Argyri K, Vrettaras J (2009) Decade review (1999–2009): progress of application of artificial intelligence tools in student diagnosis. *International Journal of Social and Humanistic Computing* 1(2):175–191. <https://doi.org/10.1504/IJSHC.2009.031006>
8. English listening test audios by Korea Institute for Curriculum and Evaluation. <http://www.kice.re.kr/main.do?s=suneung>
9. Ferrucci DA (2012) Introduction to "this is Watson". *IBM J Res Dev* 56.3(4):1–15. <https://doi.org/10.1147/JRD.2012.2184356>
10. IBM Cloud Documentation. <https://console.bluemix.net/docs/>
11. Kipnis C, Varadhan SRS (1986) Central limit theorem for additive functionals of reversible Markov processes and applications to simple exclusions. *Commun Math Phys* 104(1):1–19. <https://doi.org/10.1007/BF01210789>
12. Kiumarsi B, Vamvoudakis KG, Modares H, Lewis FL (2018) Optimal and autonomous control using reinforcement learning: a survey. *IEEE transactions on neural networks and learning systems* 29(6):2042–2062. <https://doi.org/10.1109/TNNLS.2017.2773458>
13. Mata J, de Miguel I, Durán RJ et al (2017) Artificial intelligence (AI) methods in optical networks: a comprehensive survey. *Optical switching and networking* 28:43–57. <https://doi.org/10.1016/j.osn.2017.12.006>
14. Rosenblatt M (1956) A central limit theorem and a strong mixing condition. *Proc Natl Acad Sci* 42(1):43–47. <https://doi.org/10.1073/pnas.42.1.43>
15. Russell SJ, Norvig P (2016) *Artificial intelligence: a modern approach*. Pearson Education Limited, Malaysia
16. Shahamiri SR, Salim SSB (2014) Real-time frequency-based noise-robust automatic speech recognition using multi-nets artificial neural networks: a multi-views multi-learners approach. *Neurocomputing* 129: 199–207. <https://doi.org/10.1016/j.neucom.2013.09.040>

17. Shickel B, Tighe PJ, Bihorac A, Rashidi P (2017) Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *IEEE Journal of Biomedical and Health Informatics* 22(5):1589–1604. <https://doi.org/10.1109/JBHI.2017.2767063>
18. Stallings W (2006) *Data and computer communications eighth edition*. Prentice Hall, New Jersey, pp.92–96
19. Tan WK, Hassanpour S, Rundell SD et al (2018) Comparison of natural language processing rules-based and machine-learning systems to identify lumbar spine imaging findings related to low Back pain. *Acad Radiol* 25:1422–1432. <https://doi.org/10.1016/j.acra.2018.03.008>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Yong-Sik Choi received the M.S. degree from the Department of Computer Science and Engineering, in 2020, and he has been Ph.D. degree candidate at the Department of Artificial Intelligence, Dongguk University, Seoul, Korea, since 2020. His research area include intelligent systems, machine learning, and human-robot interaction.

Jin-Gu Kang received the B.S. and LL.B. degrees from Dongguk University, Seoul, Korea, in 2019. Currently, he has been under the M.S. degree candidate at the Dept. of Computer Science and Engineering, Dongguk University, since 2019. His research areas include Robot AI, Vision and Computer Networks.

Jong Wha J. Joo received the B.S. degree from the Dept. of Computer Science and Engineering, Seoul University, Seoul, Korea, in 2005. M.S. degrees from the Dept. of Computer Science, Brown University, in 2007, Ph.D. degrees from the Dept. of Bioinformatics, UCLA, in 2015. Currently, she is an assistant professor of the Dept. of Computer Science and Engineering, Dongguk University, Seoul, Korea. Also, her research areas include Bio Big data analysis, Computational and statistical genetics, Bioinformatics, Proteomics, Computational biology.

Jin-Woo Jung received the B.S. and M.S. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1997 and 1999, respectively and received the Ph.D. degree in electrical engineering and computer science from KAIST, Korea in 2004. Since 2006, he has been with the Department of Computer Science and Engineering at Dongguk University, Korea, where he is currently a Professor. During 2001~2002, he worked as visiting researcher at the Department of Mechano-Informatics, University of Tokyo, Japan. During 2004~2006, he worked as researcher in Human-friendly Welfare Robot System Research Center at KAIST, Korea. During 2014, he worked as visiting scholar at the Department of Computer and Information Technology, Purdue University, USA. His current research interests include human behaviour recognition, multiple robot cooperation and intelligent human-robot interaction.