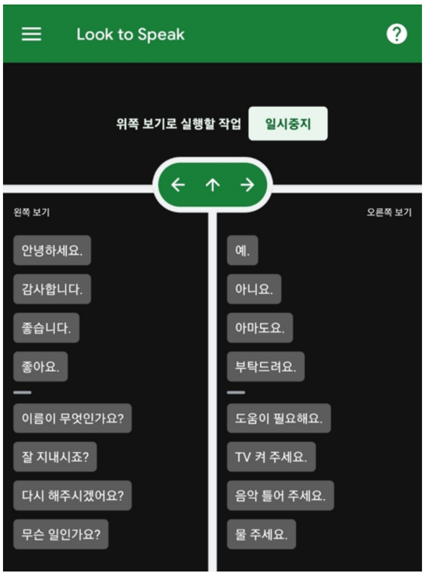
SpeAKN : Speak for ALS with Korean NLP

June Lee, Ohjoon Kwon, Youngjin Jeong
Chaeyeon Kim, Jeongmin Lee

1. Introduction

ALS (Amyotrophic Lateral Sclerosis) is a neurodegenerative disease in which motor nerve cells in the brain and spinal cord progressively deteriorate. Over time, ALS patients lose their ability to communicate using natural language, with 80-95% of patients requiring alternative communication methods known as AAC (Augmentative and Alternative Communication).

AAC is a technology that enhances communication for individuals with speech impairments, primarily utilizing hand movements. However, since ALS patients may experience difficulty with hand-based communication due to declining motor function, research has explored alternative communication methods using head or eye movements as compensation. Among these approaches, an application called "Look to Speak," which utilizes eye-tracking technology, allows users to select desired responses using eye movements. As shown in <Figure 1>, the "Look to Speak" application enables users to choose between left and right options through eye movements until their desired response appears. However, this system has limitations: users must make numerous selections before reaching their intended response, and users cannot pre-input their desired responses.



Look to Speak

위쪽 보기로 실행할 작업 일시중지

왼쪽 보기

안녕하세요.
감사합니다.
좋습니다.
좋아요.
이름이 무엇인가요?
잘 지내시죠?
다시 해주시겠어요?
무슨 일인가요?

오른쪽 보기

예.
아니요.
아마도요.
부탁드려요.
도움이 필요해요.
TV 켜 주세요.
음악 틀어 주세요.
물 주세요.

Fig 1: 'Look-to-Speak' Application

To address these limitations, we propose developing "AI-TRACKING," which combines eye-tracking technology with artificial intelligence using the pipeline illustrated in <Figure 2>. When a conversation partner's question is input as voice, the first AI model converts this input into text. Subsequently, by analyzing the textual context, the system displays response sentences with high probability of being appropriate patient responses on the screen, allowing patients to select suitable answers through eye movements while viewing these options.

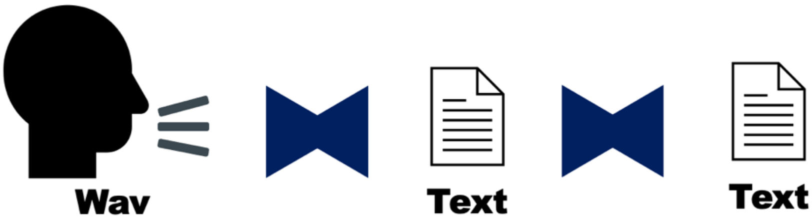


Fig 2: Overall model architecture overview. Two AI models were used: the first model processes voice data, and the second model uses the sentence output from the first model as input to generate final response sentences.

2. Method

Data: This project utilized the National Institute of Korean Language's Everyday Conversation Speech Corpus 2020 and Everyday Conversation Corpus 2020 datasets. The everyday conversation speech corpus consists of 870,162 audio files and 2,231 corresponding sentence data points, containing approximately 500 hours of conversation data from 2,739 participants. The conversations involve two or three participants with an average duration of 15 minutes per conversation. The audio data was sampled at 16kHz in PCM format. The everyday conversation corpus contains 2,232 sentence data points, encompassing the same 500 hours of conversation data from 2,739 participants as the previous dataset. The sentence data is encoded in UTF-8 JSON format.

Model and Training Process: This project experimentally tested various methods to enhance AI performance. First, we compared the performance of AdamW Optimizer and Sophia Optimizer based on learning speed per epoch. As shown in <Figure 2>, the AdamW Optimizer demonstrates faster learning progression and stabilization during training.

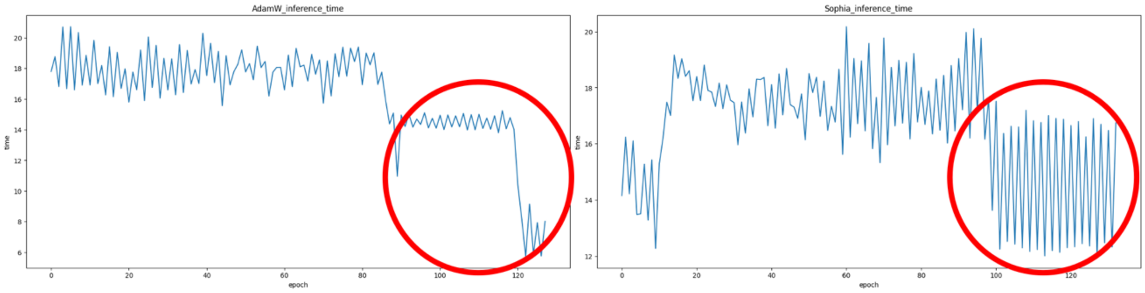


Fig 3. X-axis represents training epochs, Y-axis represents training time. Shows that the AdamW Optimizer's training time stabilizes during the learning process.

Our SpeAKN model employs an RNN layer in the output layer, and <Figure 3> shows that the GRU layer successfully overcomes the vanishing gradient problem. We used GRU layers for effective learning.

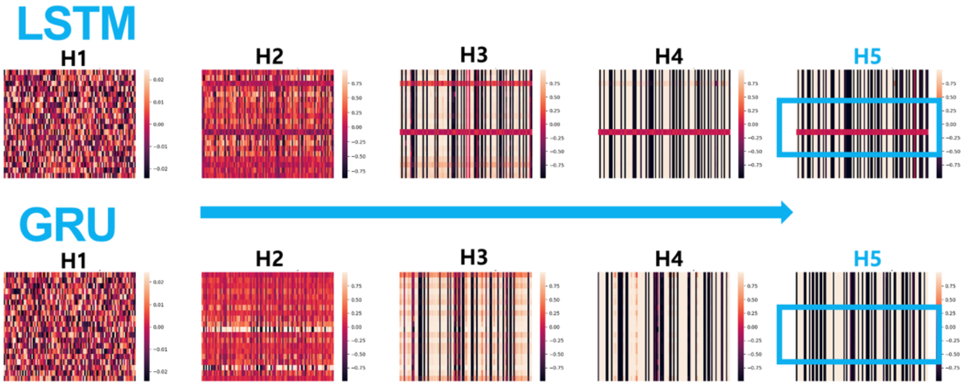


Fig 4. (Top) LSTM layer, (Bottom) GRU layer. Moving rightward shows the parameters of learned hidden states. Compared to the LSTM layer, the GRU layer shows that red parameters (vanishing gradients) disappear at H5.

For audio data, we initially standardized all audio data lengths to match the maximum audio data length to ensure consistent input dimensions. However, according to <Figure 4>, most audio data averages around 25,000, which is approximately 1/10th of the maximum audio data length. To reduce unnecessary complexity and mitigate the curse of dimensionality, we set the audio data length to 100,000.

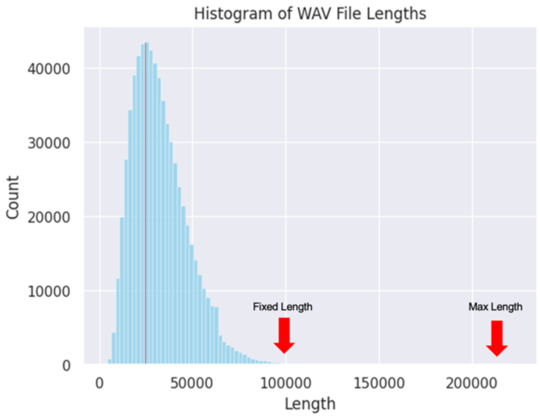


Fig 5. Histogram representation of audio data lengths. The maximum length is 220,000 while the average is 25,000, approximately 10 times smaller. By setting audio data length to 100,000 we can avoid the curse of dimensionality.

Overview of Architecture

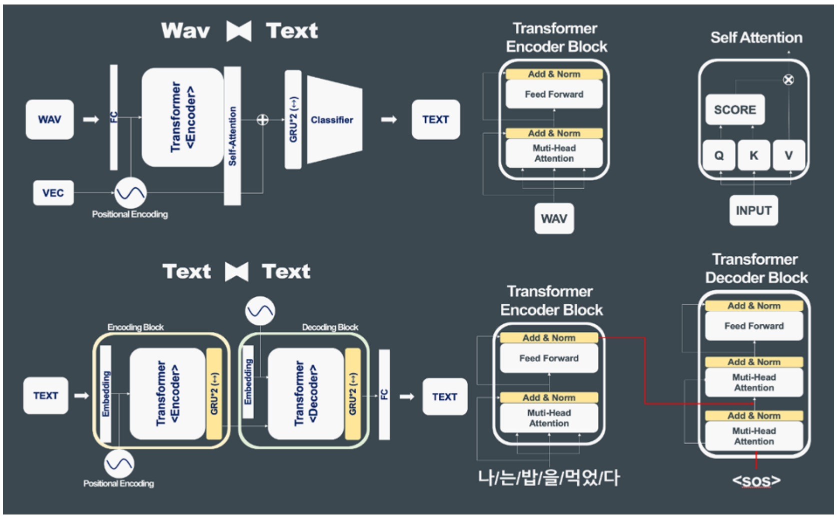
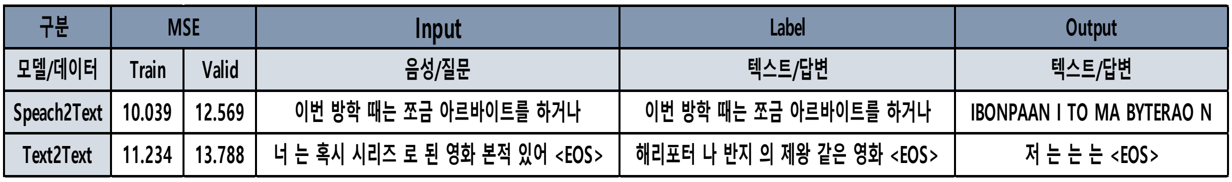


Fig 3: Overall architecture of the SpeAKN model. Composed of Wav2Text and Text2Text models, appropriately utilizing transformer encoder and self-attention layers. Unlike existing models, a GRU layer was added to focus on learning Korean word order.

Evaluation: The evaluation methodology included both numerical and visual assessments. Numerical evaluation employed MSE and BLEU metrics, while visual evaluation consisted of attention layer visualization and real-world usage evaluation. The real-world evaluation was conducted assuming actual ALS patient scenarios. This project simulated a situation where a nurse asks questions to a male ALS patient in his 30s.

3. Results



구분	MSE		Input	Label	Output
모델/데이터	Train	Valid	음성/질문	텍스트/답변	텍스트/답변
Speach2Text	10.039	12.569	이번 방학 때는 쪼금 아르바이트를 하거나	이번 방학 때는 쪼금 아르바이트를 하거나	IBONPAAN I TO MA BYTERAO N
Text2Text	11.234	13.788	너 는 혹시 시리즈 로 된 영화 본적 있어 <EOS>	해리포터 나 반지 의 제왕 같은 영화 <EOS>	저 는 는 는 <EOS>

Table 1: SpeAKN v1 Performance Metrics

These are the performance indicators when SpeAKN was first implemented. We attempted to use natural language processing metrics such as BERTscore and BLEU, but they were not well-suited for our application, so we used MSE values instead. We confirmed significant differences between output and labels and considered methods to improve performance.

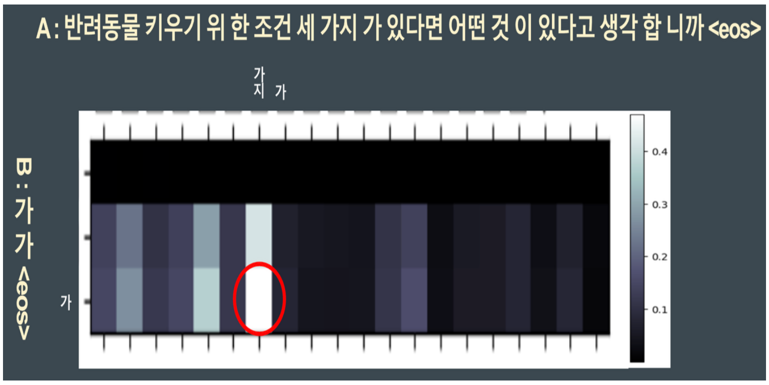


Fig 7: Visualizing Attention

To verify whether the model was learning properly, we visualized the attention output. The results showed that the model failed to recognize the important parts it should focus on during learning, and we confirmed that it was not overfitting even on the training data.

Therefore, we proceeded with the following process. For model training, we used data from the National Institute of Korean Language's everyday conversation corpus, setting the conversation content of two speakers as value and label respectively. However, we discovered that values and labels were not accurately set in the data, so we refined the data by setting values as questions ending with question marks and labels as corresponding answers to those questions. As shown in <Figure 8>, we modified the training to use only meaningful data, which comprised 14.3% of the total dataset.

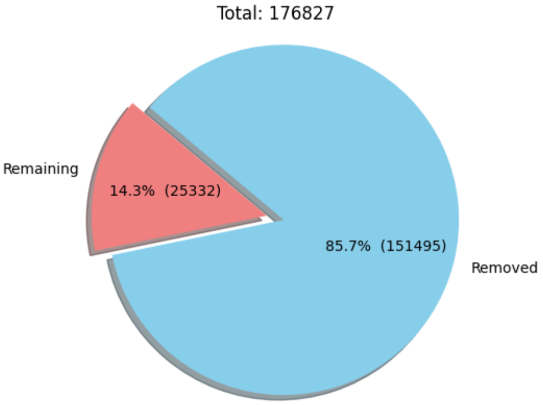


Fig 8. Pie chart showing the data actually used for training.

When we trained with accurate data and increased model complexity, we observed a slight reduction in loss but failed to produce meaningful results. To identify which tokens appeared most frequently among the tokens used, we created a bar chart showing the most frequent tokens by count. The results showed that particles (Korean grammatical particles) comprised most of the tokens appearing more than 5,000 times, as seen in <Figure 8>. We considered the prevalence of particles, characteristic of the Korean language, as a challenging factor for learning.

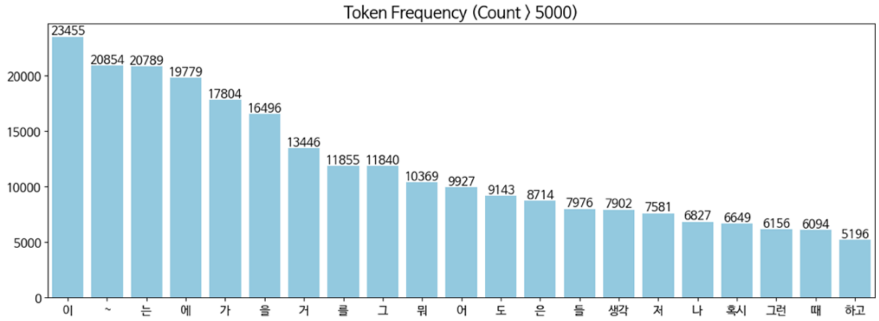


Fig 9. Bar chart showing tokens appearing more than 5,000 times. Korean particles such as '이' and '는' appear frequently.

Additionally, when examining the actual frequency of each token using a pie chart in <Figure 9>, we found that tokens appearing only once accounted for 53.8% of the total. We concluded that setting words appearing only once as <unk> tokens before training would have improved accuracy.

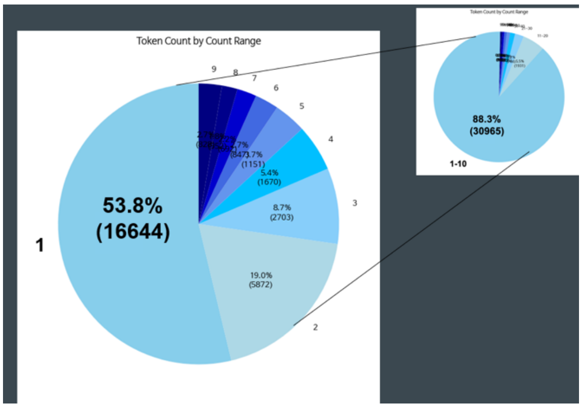
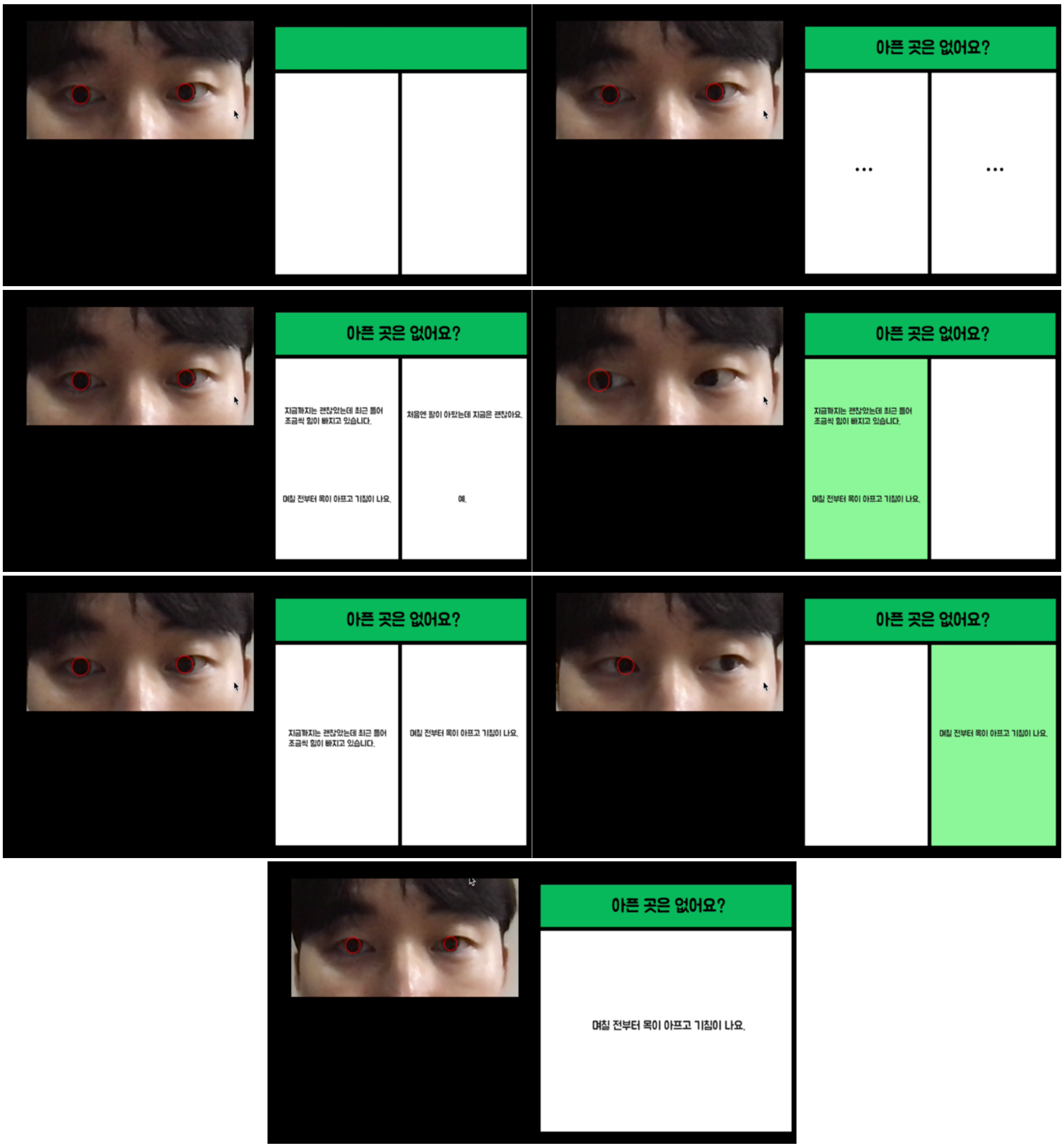


Fig 10: Pie chart showing the proportion of tokens by frequency of appearance.

Despite the process of modifying data to suit the model, performance did not significantly improve. Therefore, we reimplemented the speAKN model using Google's SpeechRecognition and Kakao's KoGPT.

4. Eye-tracking Implementation



아픈 곳은 없어요?

…

…

아픈 곳은 없어요?

지금까지는 괜찮았는데 최근 들어 조금씩 힘이 빠지고 있습니다.

처음엔 팔이 아팠는데 지금은 괜찮아요.

며칠 전부터 목이 아프고 기침이 나요.

예.

아픈 곳은 없어요?

지금까지는 괜찮았는데 최근 들어 조금씩 힘이 빠지고 있습니다.

며칠 전부터 목이 아프고 기침이 나요.

아픈 곳은 없어요?

지금까지는 괜찮았는데 최근 들어 조금씩 힘이 빠지고 있습니다.

며칠 전부터 목이 아프고 기침이 나요.

아픈 곳은 없어요?

며칠 전부터 목이 아프고 기침이 나요.

아픈 곳은 없어요?

며칠 전부터 목이 아프고 기침이 나요.

Fig 11: Eye-Tracking Implementation Results

The model recognizes speech such as "I don't have any pain (아픈 곳은 없어요?)" and generates four related responses. The patient then selects their intended response through eye tracking.

(https://github.com/junhyk-lee/Look_to_Speak)