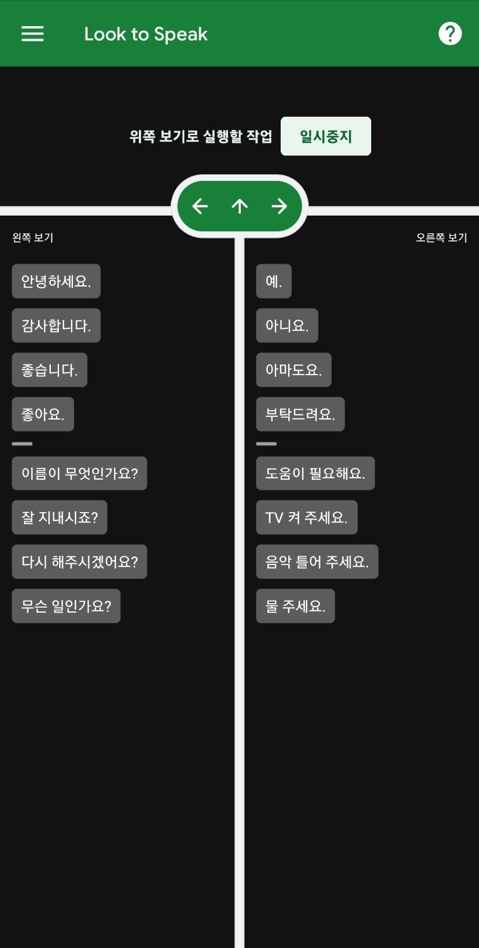
SpeAKN : Speak for ALS with Korean NLP

201802848 이준혁, 201900473 권오준, 201903228 정영진,

202001054 김채연, 202102667 이정민

1. Introduction

ALS는 근위축성 측삭 경화증으로 뇌와 척수의 운동신경 세포가 사멸하는 신경질환이다. ALS 환자들은 시간이 지남에 따라 자연어를 사용하여 의사소통하는 능력을 상실하게 되며, 80~95%의 환자들이 대체 소통방식인 ACC를 필요로 한다.

ACC는 언어 표현에 장애를 보이는 사람들의 의사소통을 향상시키는 기술로, 주로 손의 움직임을 활용한다. 그러나, ALS 환자들은 운동 능력의 저하로 인해 손을 이용한 의사소통이 어려워질 수 있기 때문에, 이를 보완하기 위해 머리나 눈을 이용한 의사소통 방식을 채택한 연구들이 있다. 그 중, 눈동자 추적 기술인 eye-tracking을 활용한 'Look to Speek'라는 어플리케이션은 눈동자를 이용하여 원하는 대답을 선택할 수 있는 어플이다. 'Look to Speak' 어플은 <그림 1>과 같이 원하는 대답이 나올 때까지 눈동자의 움직임을 통해서 왼쪽과 오른쪽 선택지를 고르는 어플이다. 그러나, 사용자가 원하는 응답을 할때까지 선택을 많이 해야하며, 사용자가 원하는 응답을 미리 입력하지 못한다는 한계점이 있다. 이런 한계점을 극복하고자 <그림 2>의 파이프라인으로 eye-tracking 기술과 인공지능을 결합한 'AI-TRACKING'을 만들고자 한다. 상대방의 질문이 음성 형태로 입력이 들어오면, 이 입력을 첫번째 인공지능 모델을 활용하여 텍스트로 변환한다. 그 후, 텍스트의 문맥을 인식하여 환자가 대답할 수 있는 확률이 높은 답변 문장들을 화면에 띄우고, 문장들을 보고 환자는 적절한 대답을 눈동자 움직임으로 선택할 수 있도록 한다.

그림 1: ‘Look-to-Speak’ 어플리케이션

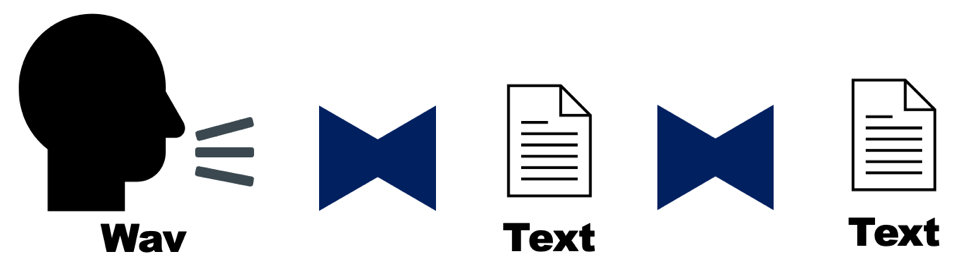


그림 2: 전체적인 모델의 개요도, 두 개의 인공지능 모델을 사용하였으며, 첫번째 모델은 음성데이터를 두번째 모델은 첫번째 모델의 출력데이터인 문장을 입력으로 사용하여 답변 문장을 최종적으로 생성함.

2. Method

데이터: 현 프로젝트는 국립국어원의 일상대화 음성말뭉치 2020년, 일상대화 말뭉치 2020년 데이터를 사용하였다. 먼저 일상대화 음성말뭉치 데이터는 음성파일 870,162개와 해당 음성말뭉치의 문장데이터 2231개로 구성되어있으며 총 2739명의 500시간여 대화 데이터를 담고 있다. 대화는 두명 혹은 세명으로 구성되어 있으며 평균 15분정도의 대화를 진행한다. 음성 데이터는 PCM 형식의 16kHz로 sampling 되었다. 일상대화 말뭉치 데이터는 문장데이터 총 2232개로 앞선 데이터와 동일하게 총 2739명의 500시간여 대화 데이터를 담고 있다. 문장데이터는 Json 형식의 UTF-8로 인코딩 되어 있다.

모델 및 학습과정: 본 프로젝트는 인공지능 성능을 높이기 위해서 다양한 방법을 실험적으로 거쳤다. 먼저 AdamW Optimizer와 Sophia Optimizer의 성능을 에포크별 학습속도를 놓고 비교하였다. <그림 2>에서 볼 수 있듯이, AdamW Optimizer가 학습을 진행함에 있어서 학습 속도가 빨라지며 안정화 되는 것을 확인할 수 있다.

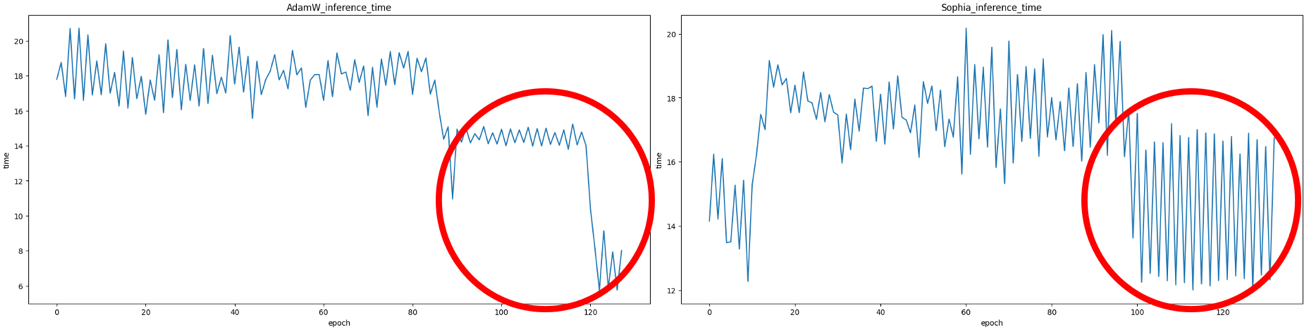


그림 3: x축-학습량, y축-학습에 걸리는 시간, AdamW Optimizer가 학습에 걸리는 시간이 안정화 되는 것을 볼 수 있음.

우리의 SpeAKN 모델은 RNN layer를 출력층에 사용하며, <그림 3>을 통해 GRU layer가 vanishing gradient 문제를 극복한 것을 볼 수 있다. 효과적인 학습을 위해 GRU layer를 사용하였다.

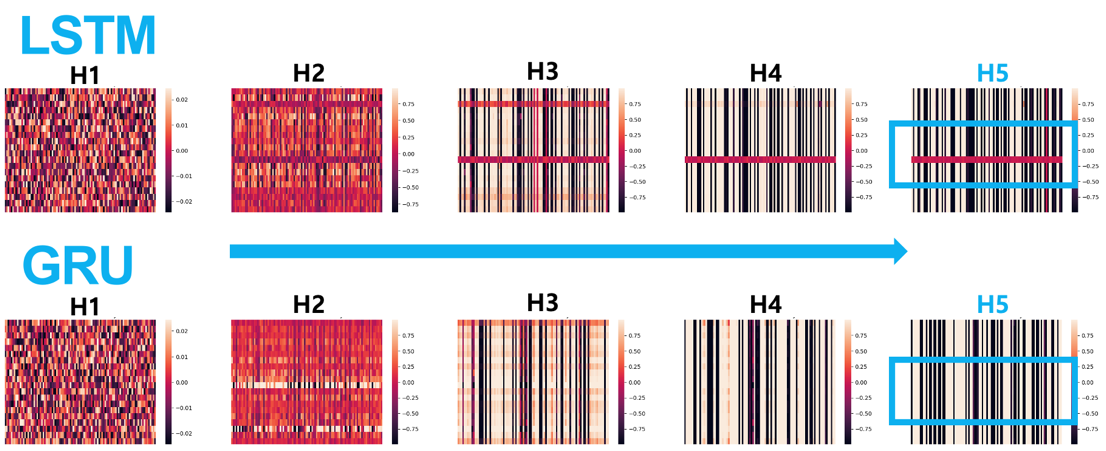


그림 4: (위)LSTM layer, (아래)GRU layer로 오른쪽으로 이동할수록 학습된 hidden states의 parameter를 표시하였다. LSTM layer에 비해 GRU layer가 H5에서 빨간색의 parameter(vanishing gradients)가 사라지는 것을 알 수 있음.

음성데이터 역시 일정한 길이를 맞추기 위해서 처음에는 모든 음성데이터의 길이를 최대 음성 데이터에 맞춰서 진행하였다. 하지만 <그림 4>에 따르면 대부분의 음성데이터는 평균 25,000으로 최대 음성데이터의 1/10배이다. 불필요한 복잡성을 낮추고, 차원의 저주를 낮추기 위해서 우리는 음성 데이터 길이를 100,000으로 설정하였다.

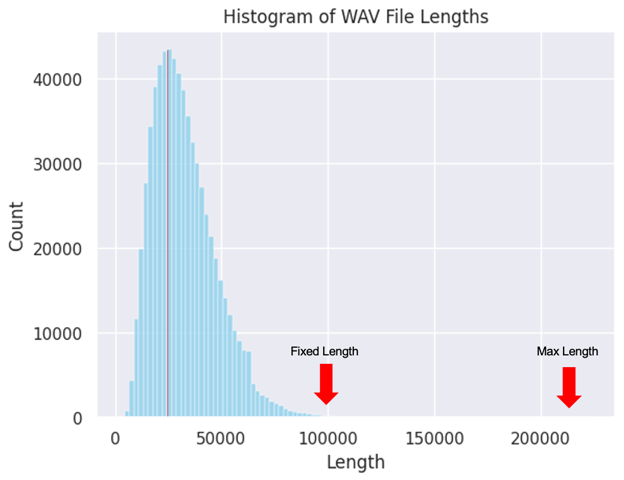


그림 5: 음성데이터의 길이를 히스토그램으로 표현하였다. 최대 길이 220,000으로 평균이 25,000에 비해 약 10배이다. 음성데이터의 길이를 100,000으로 맞춤으로써 차원의 저주에서 벗어날 수 있음.

모델 및 학습과정:

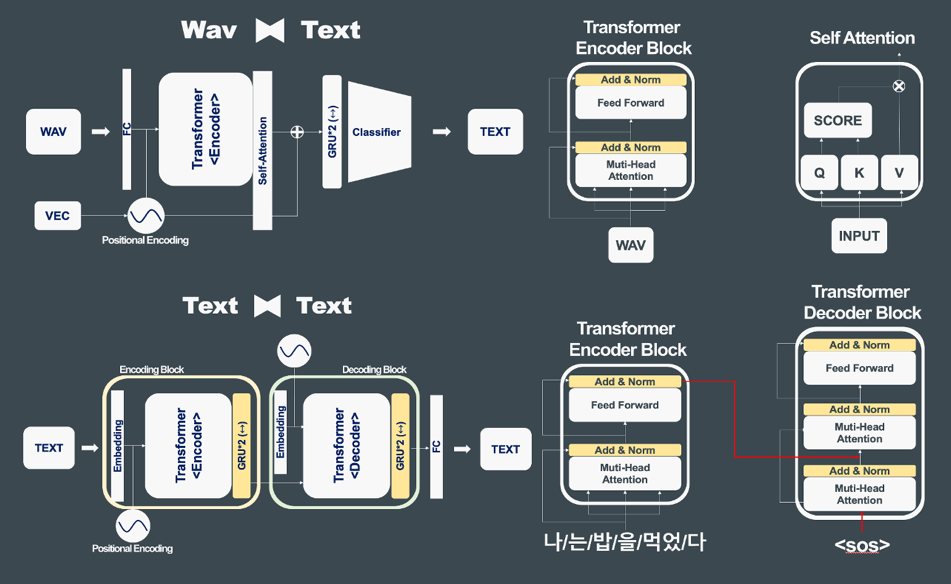


그림 6: SpeAKN 모델의 전체적 개요, Wav2Text 모델과 Text2Text 모델로 구성되어있으며 각각 transformer encoder와 self-attention layer를 적절하게 활용했다. 기존 모델과 다르게 GRU layer를 추가하여 한국어의 어순을 학습하는데 초점을 둠.

평가: 평가 방법에는 수치적 평가와 시각적 평가를 사용하였다. 수치적 평가에는 MSE, BLEU가 사용되었으며, 시각적 평가로는 Attention layer visualization과 실사용 후 평가로 이루어졌다. 실사용 평가에는 실제 ALS 환자의 상황을 가정하고 진행한다. 본 프로젝트에서는 ALS 30대 남성 환자로 환자에게 간호사가 질문하는 상황으로 진행하였다.

3. Results

**표 1: speAKN v1 성능지표**

speAKN를 처음 구현 하였을 때의 성능 지표이다. BERTscore, BLEU과 같은 자연어처리에 많이 사용되는 성능지표를 사용하려 했으나 잘 적용되지 않아 MSE값을 사용하였다. Output과 Label 사이에 큰 차이가 있음을 확인했으며, 이를 개선하고 성능을 향상시키기 위한 방법을 고민 해 보았다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

**그림 7: Visualizing Attention**

모델이 잘 학습 된 것인지 확인하기 위해 Attention output을 시각화 해 보았다. 그 결과 모델이 중요하게 생각하여 학습해야 하는 부분을 인식하지 못했고, train data에서도 overfitting되지 않은 것을 확인 했다.

따라서 다음과 같은 과정을 진행하였다. 모델의 학습에 국립국어원의 일상 대화 말뭉치에서 두명의 대화 내용을 각각 value와 label로 설정한 데이터를 사용하였다. 하지만, 데이터에서 value와 label이 정확하게 설정되지 않은것을 확인하여서 value를 물음표로 끝나는 질문으로, label을 그 질문에 해당하는 대답으로 설정하여 데이터를 더 정확하게 하였다. <그림 8>처럼 실제로 사용한 데이터는 전체 데이터 중 14.3%로 유의미한 데이터로만 학습을 진행하도록 수정하였다.

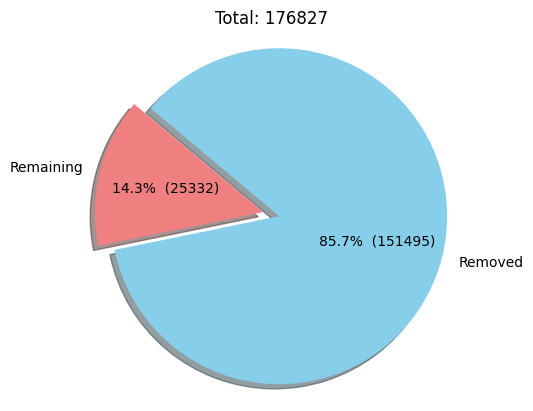


그림 8: 실제로 학습에 사용한 데이터를 pie chart로 나타내었음.

정확한 데이터로 학습을 하고, 모델을 더 복잡하게 했을 때, loss가 조금 줄어드는 것을 확인했지만, 유의미한 결과를 출력하진 못했다. 그래서 사용한 토큰 중 빈도수가 많이 등장한 토큰이 무엇인지 확인하기 위해 토큰의 개수별로 가장 많이 등장한 토큰을 bar chart로 확인하였다. 확인해본 결과 5000번 이상 등장하는 토큰 중 조사가 대부분이라는 것을 <그림 8>에서 확인할 수 있었고, 한국어 특성상 조사가 많이 등장하는 것이 학습을 하는데 어려운 점이라고 생각했다.

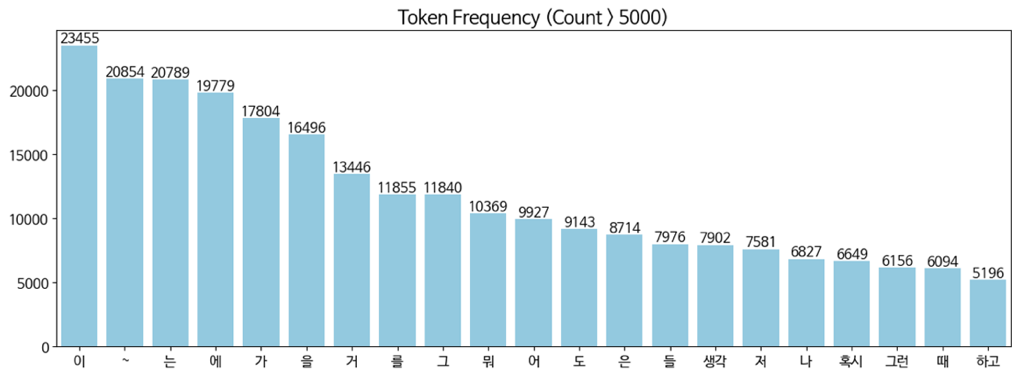


그림 9: 5000번 이상하는 token을 나타낸 bar chart로, '이', '는' 등의 조사가 많이 등장하는 것을 알 수 있음.

또, 실제로 각 토큰이 몇 번 등장하는지 <그림 9>에서 pie chart로 확인해 보았을 때, 한번만 등장하는 토큰이 53.8%로 많이 차지한다는 것을 알 수 있었다. 한번만 등장하는 단어를 <unk>로 설정한 후 학습을 진행했다면 더 정확도가 높아졌을 것이라는 결론을 내렸다.

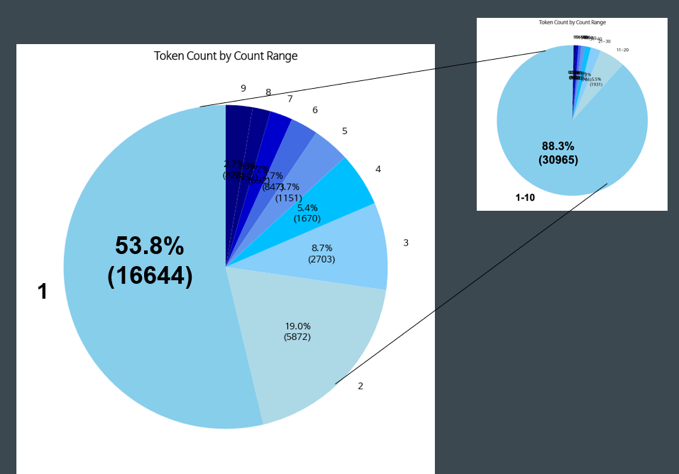


그림 10: 토큰 등장 횟수별 차지하는 비율을 나타낸 pie chat임.

데이터를 모델에 알맞게 수정하는 과정을 거쳤음에도 성능이 크게 향상되지 못하였고 따라서 구글의 SpeechRecognition과 카카오의 KoGPT를 사용하여 speAKN 모델을 다시 구현하여 보았다.

4. Eye-tracking 구현 모습

스크린샷이(가) 표시된 사진

자동 생성된 설명스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷, 텍스트, 인간의 얼굴이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 인간의 얼굴이(가) 표시된 사진

자동 생성된 설명

스크린샷, 인간의 얼굴이(가) 표시된 사진

자동 생성된 설명스크린샷, 인간의 얼굴, 텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 인간의 얼굴이(가) 표시된 사진

자동 생성된 설명

**그림 11: Eye-Tracking 구현 결과**

“아픈 곳은 없어요”와 같은 음성을 모델이 인식하고 그와 관련된 4개의 답변이 생성된다. 그리고 환자가 답변하고자 하는 내용을 eye tracking을 통해서 결정한다.

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