

## Deep learning을 활용한 역점역 프로그램



과목명	창업연계융합종합설계1	담당교수	정경훈
팀이름	전공	학번	이름
국민브레일	융합기계공학전공	20181206	이한서
	융합기계공학전공	20181219	황원재
	융합기계공학전공	20171208	서형준
	융합기계공학전공	20171229	지임준

## < 목 차 >

### 제1장 서론

1-1 프로그램 필요성 .....	1
1-2 작품을 위한 배경지식 .....	2
1-3 기존 프로그램의 한계 .....	4

### 제2장 본론

2-1 개발 언어 선정 .....	5
2-2 학습 모델 선정 .....	5
2-3 학습을 위한 데이터 .....	8
2-4 BART 모델 학습 .....	11
2-5 기존 프로그램과의 성능 비교 .....	13
2-6 성능 비교 결과 평가 .....	16

### 제3장 결론

3-1 프로젝트의 요약 .....	17
3-2 프로그램 활용방안 .....	17
3-3 사회적 측면에서의 기대효과 .....	18

# 1. 서론

## 1-1. 프로그램의 필요성

시각장애인들은 비시각장애인과 다르게 많은 제약이 있다. 그 중 글씨를 읽을 수 없다는 것이 큰 문제로 비춰진다. 그를 해결하기 위해서 개발된 문자 체계가 바로 점자이다. 점자는 시각장애인들이 지식과 기술을 습득할 수 있는 가장 중요한 도구이다. 시각장애인들에게 점자는 문자 이상의 의미가 있다. 자신감과 독립성은 물론 사회생활의 동등권을 획득하는 시각장애인들의 의사소통 수단이 바로 점자이기 때문이다.

하지만 우리나라 시각장애인 약 25만명 중, 점자를 실제로 해독할 수 있는 비율을 15%가 채 되지 않는다. 다음은 2017년 보건복지부 장애인 실태조사 내용이다.<sup>1)</sup>

<표1 - 2017 보건복지부 장애인실태조사>

〈표 6-3-4〉 시각장애의 점자해독 여부

(단위: %, 명)

구분	남자	여자	전체
가능하다	19.5	4.7	12.4
불가능하다	80.1	92.4	86.0
배우는 중이다	0.4	2.9	1.6
계	100.0	100.0	100.0
전국추정수	38,477	35,823	74,300

주: 시력장애를 가진 시각장애인 중 1~4급 장애인에 대한 점자해독 가능비율임.

비시각장애인들은 모든 시각장애인이 점자를 해독할 수 있다고 생각하는 경우가 대부분이지만 결과는 그렇지 않다. 시각장애인들이 점자를 해독하지 못하는 가장 큰 이유는 후천적으로 시각장애인이 된 경우가 90.3%로 대부분이기 때문이다.<sup>2)</sup> 점자는 하나의 언어이다. 새로운 언어를 익힌다는 것은 시간이 오래 걸리고 어려운 일이다. 게다가 점자는 다른 문자와 달리 눈이 아닌 손가락 끝의 감각으로만 읽어야 한다. 이에 시각장애인이 점자를 완전히 익히기 위해서는 손끝의 감각을 예민하게 곤두세워야 할 뿐만 아니라, 새로운 언어 체계를 익히기 위해 많은 시간과 노력을 들여야 한다.

특히 후천적 시각장애인은 이미 시각 문자 체계에 익숙하므로, 점자 습득에 더 큰 어려움을 겪는다. 눈으로 글을 읽을 때는 한눈에 문장의 앞뒤 구조를 파악하고, 문단의 모양과 밀줄 등의 보조적인 부분을 통해 내용 파악이 가능했지만, 점자는 글자 하나하나를 쓰인 순서에 의존하여 읽어야 하다 보니 전체적인 내용 파악이 어렵기 때문이다. 이에 후천적 시각장애인들은 점자의 학습 자체를 포기하거나, 발전된 기술로 점자를 익히지 않아도 불편함 없이 읽을 수 있는 미래를 바란다. 아래의 기사를 보면, 점자학습의 필요성을 알게 될 것이다.

“한국장애인단체총연맹에 따르면 2021년 기준 전체 출판되는 책 중 점자로 출간하는 비율은 0.2%에 불과한 데다, 책값이 일반 도서에 비해 5배 이상 비싸다. 자연히 점자를 학습할 수 있는 교구 보급률로 낮아 1% 미만에 그친다. 그런 탓에 학습 기회가 부족한 시각장애인은 86%가 글을 읽지 못한다. 또 전세계 시각장애인 중 점자를 사용할 수 있는 경우는 5%에 불과하다.”<sup>3)</sup>

위에서 시각장애인에 관련된 두 가지 기사를 참고하였다. 첫 번째는 2012년 11월 기사이다. 그 때 당시 조사된 시각장애인 점자 문맹률은 93.9%였다. 그 후 두 번째로 2022년 11월 기사를 보면 점자 문맹률은 86%로 나타나 있다. 10년이라는 기간 동안, 시각장애인의 점자 문맹률 실태는

1) 보건복지부, 2017년 장애인실태조사 발간등록번호 11-1352000-000568-12

2) 최우리(2012.12.02.), 「당신도 어느날 갑자기 안 보일 수 있다...후천성이 90%」, 한겨레, URL: [https://www.hani.co.kr/arti/society/society\\_general/558762.html](https://www.hani.co.kr/arti/society/society_general/558762.html).




3) 김민석(2022.11.06.), “점자책 출간 비율 0.2%... 시각 장애인 86% 글 못 읽어”, 서울신문 URL: <https://www.seoul.co.kr/news/newsView.php?id=20221106500034>

크게 나아지지 않았음을 확인할 수 있다. 시각장애인은 문맹으로 인하여 먼 미래에까지 사회 진출의 제약까지도 스스로 안게 될 수 있다. 따라서 시각장애인과 비시각장애인뿐만 아니라 비장애인과 장애인이 서로 더불어 살아가는 사회를 만들기 위해서 점자 학습은 선택이 아닌 필수이다. 후천적 시각장애인의 점자 해독능력을 향상시키기 위해서는, 점자를 한글로 바꾸어주는 “역점역 프로그램”이 필요하다고 생각했다. 하지만 기존의 역점역 프로그램은 점자의 특성상 제대로 번역할 수 없는 한계가 있다. 우리는 그것을 해결하기 위해 “Deep learning”을 활용하여 그 문제를 해결하고자 한다.

## 1-2. 작품을 위한 배경지식

점자는 19세기 초 프랑스 육군 포병 장교 니콜라스 바루비에가 야간 작전 시 암호용으로 처음 개발했다. 세로로 6개의 점 2줄로 만들어졌던 12점 암호 점자는 그 후 1821년 프랑스의 파리맹 학교에 전달되었고, 당시 학생이던 시각장애인 루이 브라이유에 의해 10여 년간 연구, 실험 과정을 거쳐 1834년 지금의 시각장애인 문자인 6점 점자가 완성 되었다. 이 6점 점자가 영국과 미국, 일본을 거쳐 우리나라에도 전해졌다. 이를 바탕으로 1923년 당시 특수교육기관인 제생원 맹아부 교사였던 박두성은 브라이유의 6점식 점자를 토대로 한글점자 개발에 착수했다. 박두성은 제생원 학생, 일반 시각장애인들과 함께 브라이유식 한글점자 연구를 시작하여 1921년 6점식 한글점자를 내놓게 되었다. 그 후 수차례의 수정, 보완을 거쳐 1926년 11월 4일 훈민정음과 음이 비슷한 ‘훈맹정음’이란 이름으로 한글점자를 발표하였다. 한글점자의 창안이 세상에 알려지면서 박두성은 시각 장애인들의 세종대왕이라 일컬어지게 되었다. 그에 의해 만들어진 한글점자는 시각장애인 교육의 기틀이자 재활의 통로가 되어왔다.<sup>4)</sup>

점자는 시각장애인을 위해 개발된 문자로, 볼록한 점(혹은 색이 칠해져 있는 점)들의 위치를 사용해서 문자를 표기하도록 만들어져있다. 좌 상단 1번 점을 시작으로 그 아래 2번과 3번, 우 상단 4번과 그 아래 5번, 6번 점으로 각각 구성되어있다. 일반적으로, 아래의 그림 1과 같이 점 6개를 이용해서 문자를 표기한다.

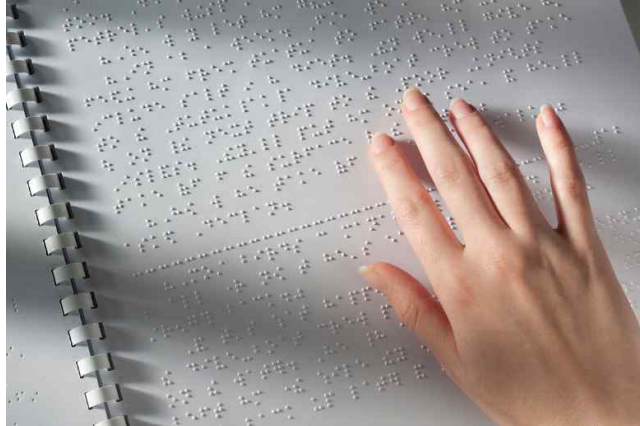
1		4	← 상단
2		5	← 중단
3		6	← 하단

<그림 1 - 총 6개의 점으로 구성되는 점자><sup>5)</sup>

시각장애가 없는 사람이 점자를 배워 눈으로 읽는 것을 ‘시독(視讀)’이라 하여 일반인도 점자로 된 글을 읽을 수는 있지만, 점자는 시각장애인을 위해 만든 문자이므로 볼록 튀어나온 부분을 아래의 그림 2와 같이 손가락(촉각)을 통해 만지면서 글을 읽는다.

4) 서울시각장애인복지관, URL : [https://bokji.or.kr/sub/sub06\\_03.php](https://bokji.or.kr/sub/sub06_03.php)

5) 국립국어원, URL : [https://www.korean.go.kr/front/page/pageView.do?page\\_id=P000302](https://www.korean.go.kr/front/page/pageView.do?page_id=P000302)



<그림 2 - 촉각을 통해 글을 읽는 점자>

점역이란 일반적으로 사용되는 한글, 영어, 숫자, 문장 부호등의 문자들로 이루어진 단어나 문장을 점자로 번역하는 것을 뜻한다. 역점역이란 점자를 다시 한글, 영어, 숫자, 문장 부호로 번역하는 것을 뜻한다. 점역의 과정은 (문자 → 점자)이며 이 구조는 <one to one> 구조를 이룬다. 하지만 역점역의 경우, 6개의 점으로 표현되기 때문에 표현할 수 있는 개수가  $64(2^6)$ 가지로 한정적이다. 그렇기 때문에 역점역 과정(점자 → 문자)은 하나의 점자가 여러 문자로 표현될 수 있는 <one to many> 구조를 이룬다.

점자로 표현해야 하는 문자의 경우 자음 19개, 모음 21개로 국어로 한정하여도 40개 이상이고, [영어, 일본어, 중국어 등]과 같은 외국어와 0부터 9까지의 아라비아 숫자, [!, ?, ~, .]와 같은 문장 부호 등을 모두 포함하려면, 64가지보다 훨씬 더 많은 경우의 수가 필요하다. 그러므로 같은 모양이지만 서로 다른 의미로 번역될 수 있는 점자들이 존재할 수밖에 없다. 따라서 기존의 1:1 대응 방식으로 100% 정확도의 역점역 수행은 근본적으로 불가능하다. 예를 들어, 아래의 그림 4과 같이 한국어 초성 'ㄴ', 알파벳 'C', 숫자 '3'의 3가지 경우 모두 1, 4번 점자( )로 표현되어 점자의 모양이 똑같다.

자음	초성														된소리
	중성														
숫자 / 연산	수표	1	2	3	4	5	6	7	8	9	0	.	.		
	*	-	x	=	=										
영어	로마자표														
	대문자표														
	이중대문자표														

여기서 역점역의 문제가 끝이 아니다. 한글을 점자로 바꾸는 과정에서 점자의 길이가 매우 길어져 가독성이 떨어지는 현상을 방지하기 위해 “축약법칙”이라는 과정이 추가된다. 이 과정으로 인하여 점자를 다시 문자로 바꾸게 되는 역점역 과정은 더욱 해결하기 어려운 상황에 놓이게 되었다. 다음 그림5는 국립국어원에서 개정한 2017 한글점자규정개정서 내용의 일부이다.<sup>6)</sup>

## 제2장 약자와 약어

### 제6절 약자

**제12항**

다음 글자가 포함된 글자들은 아래 표에 제시한 약자 표기를 이용하여 적는 것을 표준으로 삼는다.

가	나	다	마	바	사	자	카	타	파	하
가	나	다	마	바	사	자	카	타	파	하
가	나	다	마	바	사	자	카	타	파	하
가	나	다	마	바	사	자	카	타	파	하

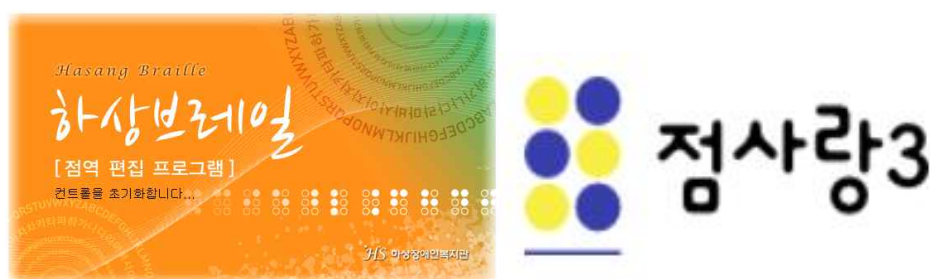
가자      바다      자동차

<그림4 - 축약법칙 제 12항>

제 12항뿐만 아니라 18항까지 다양한 축약 법칙이 존재한다, 이는 점자를 다시 문자로 바꾸는 역점역 과정의 걸림돌이 된다.

### 1-3. 기존 프로그램의 한계

“점사랑”, “하상브레일” 이라는 기존 점역과 역점역 프로그램이 존재한다. 점사랑은 한국시각장애인연합회에서 제공하는 점역 및 역점역을 수행하는 프로그램이고, 하상브레일은 하상장애인복지관에서 제공하는 점역 프로그램이다.

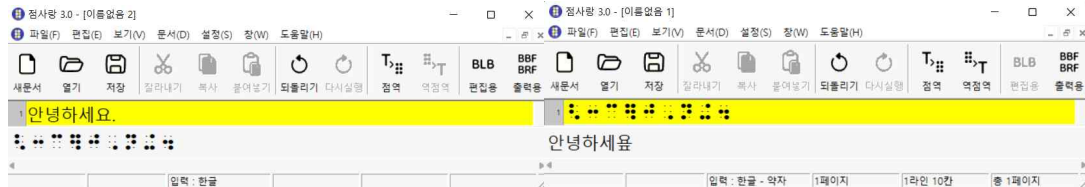


<그림5 - 기존 프로그램 하상브레일&점사랑>

이 두 프로그램의 한계는 예시 문장을 점자로 바꾸는 점역을 한 후, 그것을 그대로 역점역하면 처음 입력했던 예시 문장과 달라지는 경우가 발생한다. 예시로 “안녕하세요.”를 입력하고 점역을 통해 점자로 변경 후 그것을 그대로 역점역하면 ‘안녕하세요’와 같이 출력된다. 온점의 점자와 한글 종성 ‘ㅃ’이 같은 점자로 지정되어있기 때문이다. 역점역 될 수 있는 모든 경우의 수를 출력하는 것이 아니라 역점역 될 수 있는 문구 중 임의의 하나의 값만을 출력한 것으로, 이는 모양이 중복되는 점자로 인해 생기는 1:1 대응 방식 역점역의 근본적인 문제이다.

6) 국립국어원, 2017 개정 한국 점자 규정, 발간등록번호 11-1371028-000702-01

다시 말하자면, ‘안녕하세요.’로도, ‘안녕하세요.’로도 역점역 될 수 있기에 생기는 오류인 것이다. 사람은 일반적으로 ‘안녕하세요.’라고 역점역된 것은 옳은 문장이고, ‘안녕하세요.’이라고 역점역된 것은 잘못된 역점역이라는 것을 판단할 수 있지만, 컴퓨터의 경우 역점역 시 ‘안녕하세요.’도 ‘안녕하세요.’와 마찬가지로 역점역의 규칙을 어기지 않은 번역이기에, 이를 그대로 출력하는 것이다. 아래의 그림 7는 동일한 모양의 점자가 다르게 역점역 되는 것에 대한 이해를 돕기 위해, 위 예시에 대한 점사랑의 역점역 결과를 나타낸 것이다.



<그림6 - 점사랑 프로그램, 좌(점역 결과), 우(역점역 결과)>

그림6에서 볼 수 있듯이 동일한 점역을 시행한 점자를 다시 역점역시 정확한 역점역이 되지 않는 것을 알 수 있다. 이처럼 역점역시 한가지의 점자가 여러 가지의 문자를 나타내는 <one to many>구조의 역점역 결과는 근본적으로 정확할 수 없다. 따라서 우리는 이를 해결하기 위해 기계 학습(machine learning)을 활용하여 이 문제를 해결하고자 했다. 기존의 낮은 정확도의 역점역 프로그램이 아닌, 틀릴 수밖에 없는 근본적인 문제를 해결한 역점역 프로그램을 발하고자 한다.

## 2. 본론

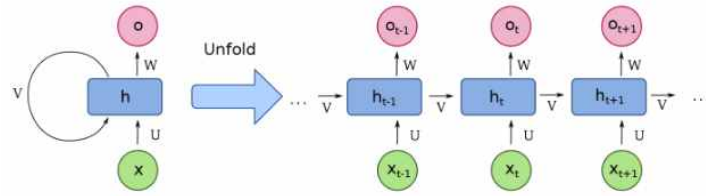
### 2-1. 개발 언어 선정

본 프로젝트의 목표는 Deep learning을 활용하여 점자를 기존의 언어로 번역해주는 프로그램을 개발하는 것이다. 따라서 가장 유명한 딥러닝 방식인 CNN, RNN, GAN을 사용할 수 있는 Python 개발 환경이 프로젝트를 위해 적절할 것으로 생각했고, Python은 각종 웹 프로그램뿐만 아니라 스마트폰 어플리케이션, 키오스크 등 프로그램의 후속 활용 범위 또한 넓다고 생각했기에, 개발 언어로 최종 선정하였다.

### 2-2. 학습 모델 선정

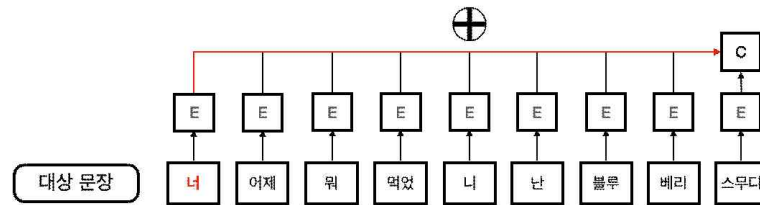
순환 신경망은 출력이 입력으로 돌아가는 구조이다. 문장을 표현할 때, 이전 단어에서의 hidden state가 현재 단어의 hidden state를 결정하는 데 사용되면서, 단어의 뜻을 순차적으로 반영하는 구조가 순환 신경망이다. 이 구조에서 각 단어의 벡터는 바로 앞 단어의 벡터에 의해서만 영향을 받는다. 즉, 순차적인 입력 때문에 거리가 먼 단어의 영향력이 줄어드는 Long Term Dependency Problem이라는 단점이 존재한다. 긴 문장의 경우, 앞쪽에 나온 단어의 영향력을 모델이 잊어버리게 된다. 역점역에서는 이 문제를 반드시 해결해야 한다. 음절 단위가 아닌 자음, 모음과 같은 음절 구성 요소를 모두 점자로 표현해야 하므로, 문장의 길이가 다른 언어에 비해 압도적으로 길기 때문이다. 아래는 기존 순환 신경망 구조의 문제인 Long Term Dependency Problem을 그림으로 나타낸 것이다.





<그림7 - Long Term Dependency Problem>

Long Term Dependency Problem를 해결하기 위해서는 우리는 RNN방식이 아닌 Attention이라는 구조를 선택하게 되었다. Attention 구조는 각 입력 단어와 출력 상태가 직접 연결되는 신경망을 추가하여, 직전의 단어뿐 아니라 다른 모든 단어가 현재 결과에 기여할 수 있도록 만들어진 구조이다. 이는 기존 구조와 달리 각 단어의 상관관계를 직접적으로 나타낼 수 있다. 각 단어가 가중치의 형태로 직접 연결되어 있기 때문이다. 예를 들어, train data에서 두 단어가 같이 자주 나오면 가중치가 크게 학습되고, 모델은 해당 단어 쌍이 서로 강한 상관관계를 가짐을 알게 된다. 즉, 아래의 Attention 구조를 표현한 블록선도와 같이, 서로 다른 단어의 상관관계를 가중치의 형태로 반영한다.



<그림8 - Attention Method에 대한 블록선도>

Attention 구조는 RNN에서 보조적으로만 활용되다가, 2017년에 구글이 Attention만으로도 충분히 문장을 모델링 할 수 있다는 의견을 냈고, 결국 Attention만을 이용해 언어를 기술할 수 있는 모델이 탄생했다. 이것이 Transformer 구조이다. 기존 구조에서는 문장을 “단어의 연속적인 배열”로 간주하였다. 그러나 Transformer에서 문장은 “단어 간의 Attention들의 합”으로 나타낸다. 즉, 문장 전체 구조를 Attention을 그물처럼 엮은 형태로 나타낼 수 있는 것이다. 또한, Transformer는 문장을 잠재 표현으로 변환하는 Encoder와 이를 복원하는 Decoder, 이렇게 크게 두 부분으로 나눌 수 있다. Transformer의 전체 구조를 나타낸 블록선도를 아래와 같이 첨부하였다.

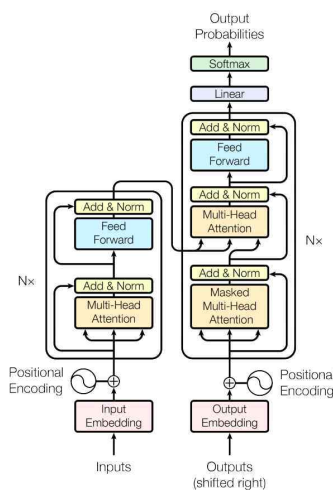


Figure 1: The Transformer - model architecture.

<그림9 - Transformer 구조>



앞서 언급했듯이, 점자, 특히 한글 점자는 문장의 길이가 매우 길다. 따라서 본 프로젝트를 성공적으로 수행하기 위해서는 Long Term Dependency 문제를 반드시 극복해야만 한다. 따라서 역점역을 위한 최종적인 딥러닝 학습 모델로 Transformer 계열을 선정하였다. 그리하여 Transformer 계열 모델들의 성능 평가 논문을 참고하였고, 다음은 우리가 참고한 논문의 성능 평가표이다.

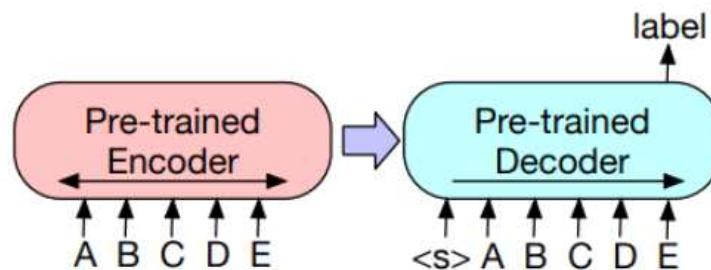
<표2 – BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, 논문><sup>7)</sup>

Model	SQuAD 1.1 F1	MNLI Acc	ELI5 PPL	XSum PPL	ConvAI2 PPL	CNN/DM PPL
BERT Base (Devlin et al., 2019)	88.5	<b>84.3</b>	-	-	-	-
Masked Language Model	90.0	83.5	24.77	7.87	12.59	7.06
Masked Seq2seq Language Model	87.0	82.1	23.40	6.80	11.43	6.19
Permutated Language Model	76.7	80.1	<b>21.40</b>	7.00	11.51	6.56
Multitask Masked Language Model	89.1	83.7	24.03	7.69	12.23	6.96
	89.2	82.4	23.73	7.50	12.39	6.74
BART Base						
w/ Token Masking	90.4	84.1	25.05	7.08	11.73	6.10
w/ Token Deletion	90.4	84.1	24.61	6.90	11.46	5.87
w/ Text Infilling	<b>90.8</b>	84.0	24.26	<b>6.61</b>	<b>11.05</b>	5.83
w/ Document Rotation	77.2	75.3	53.69	17.14	19.87	10.59
w/ Sentence Shuffling	85.4	81.5	41.87	10.93	16.67	7.89
w/ Text Infilling + Sentence Shuffling	<b>90.8</b>	83.8	24.17	6.62	11.12	<b>5.41</b>

Table 1: Comparison of pre-training objectives. All models are of comparable size and are trained for 1M steps on a combination of books and Wikipedia data. Entries in the bottom two blocks are trained on identical data using the same code-base, and fine-tuned with the same procedures. Entries in the second block are inspired by pre-training objectives proposed in previous work, but have been simplified to focus on evaluation objectives (see §4.1). Performance varies considerably across tasks, but the BART models with text infilling demonstrate the most consistently strong performance.

참고한 논문에서는 다양한 데이터로 여러 모델들을 테스트하였고 그에 대해 성능지표를 여러 부분으로 나누어 모델들을 평가하였다.

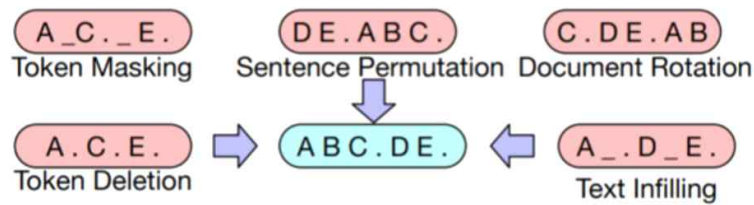
그 중 BART(Bidirectional and Auto-Regressive Transformers)가 ELI5를 제외한 모든 task에서 가장 좋은 성능을 가짐을 확인하였고 특히 BART는 generation task에서 성능 향상을 이루면서 text-infilling task에도 좋은 성능을 보였기 때문에 우리의 프로젝트의 학습 모델로 적합하다고 판단하여 BART를 학습 모델로 선정하였다.



<그림10 – BART 모델의 개념도>

BART 모델은 transformer 계열인 BERT와 GPT를 합친 형태를 가지며 양방향으로 정보를 교환할 수 있으며 자기 자신을 입력으로 하여 자기 자신을 예측하는 구조를 동시에 가진다. denoising autoencoder로 많은 종류의 downstream 테스트에서 잘 작동한다는 장점이 있다. BART의 학습 방법은 텍스트에 임의적인 noising 함수를 통해 오염시키고 Sequence to Sequence 모델이 원래의 텍스트를 복원하기 위해 학습된다. noising 기법은 다음과 같다.

7) Cornell Univ, Computation and Language (cs.CL); Machine Learning (cs.LG); Machine Learning (stat.ML), (2019) BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. p7810

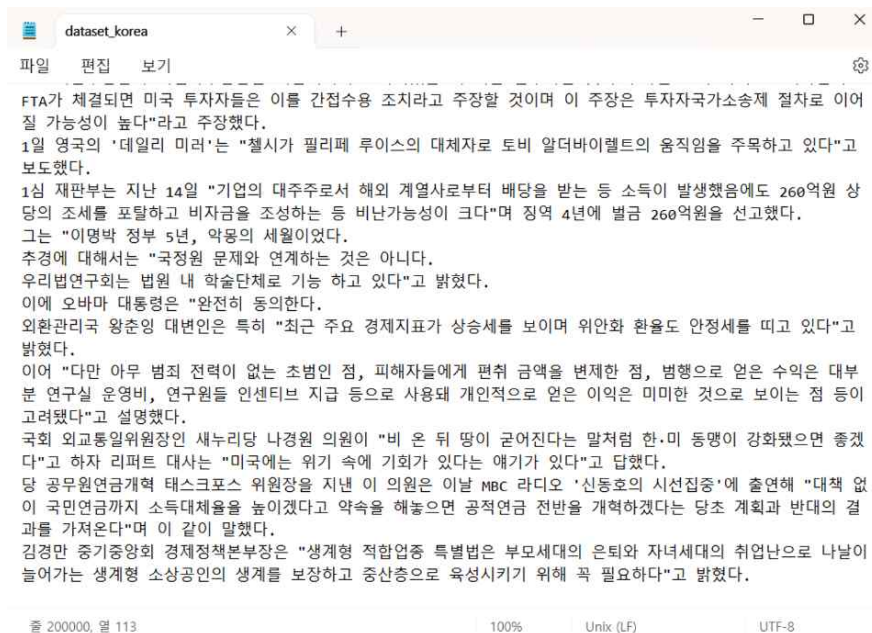


<그림11 - BART의 Noising 기법>

## 2-3. 학습을 위한 데이터

### 2-3-1. 데이터 선정

딥러닝 모델을 통한 학습을 위해서는 [기존 언어 - 점자]의 데이터 쌍이 필요하다. 하지만 시중에 공개되어있는 [점자 - 기존 언어] 데이터 쌍은 현재 없는 상황이다. 따라서 기존의 한글 말뭉치 데이터를 정역하여 데이터 쌍을 직접 제작하기로 했다. 본 프로젝트에서 사용된 말뭉치 데이터는 AI 허브에서 내려 받았으며,<sup>8)</sup> 길이는 총 20만 문장이며 우리나라 기사들을 모아둔 자료이다. 이는 한글, 문장 부호, 영어가 모두 포함된 TXT 파일로 구성되기에 본 프로젝트의 목표에 적합한 Data set이라고 생각했다.

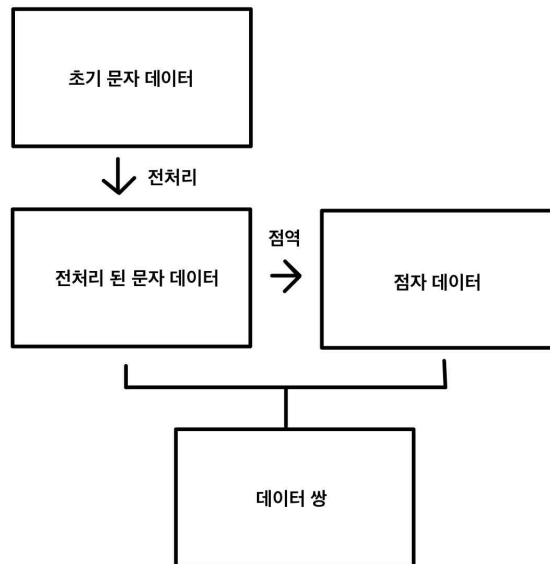


<그림12 - 기존 문자 데이터>

### 2-3-2. 데이터 전처리

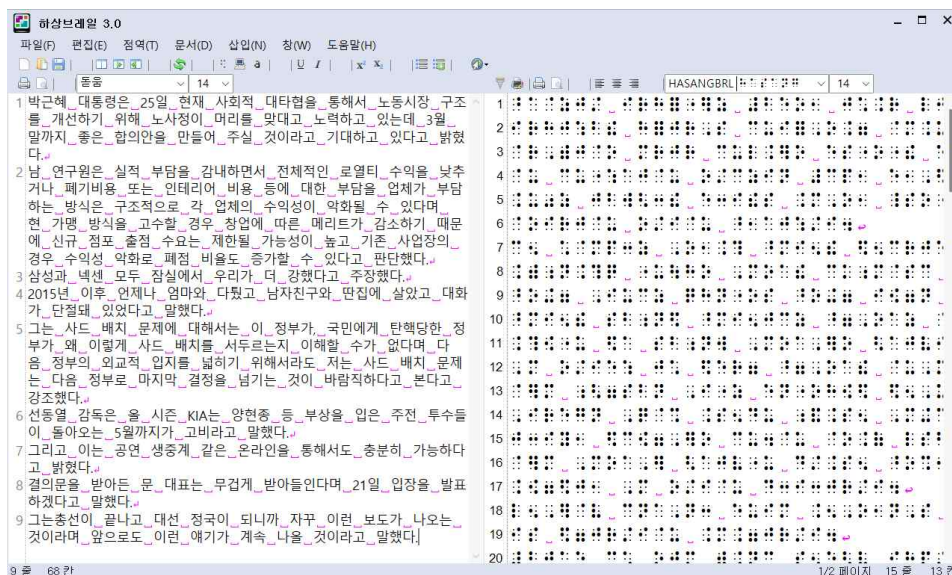
딥러닝에 있어서 학습 데이터는 매우 중요한 요인이다. GIGO(garbage-in garbage-out)라는 말도 있듯이, 우리가 선정한 데이터는 아직 정제되지 않았기에 정제하는 과정, 즉 데이터 전처리 과정을 거쳐야 한다. 우리는 20만 문장의 데이터에서 학습 모델에 악영향을 미칠 것 같다고 판단되는 요소를 다음과 같이 뽑았다. 문장을 해석하는데 아무 지장 없으며 점자로 표현하기 힘든 특수 문자들을 제거하기로 했다. 전처리 코드는 python에서 re 모듈을 사용하여 제거하였다. 기존 언어 데이터는 txt 확장자로, 점자 데이터는 pickle 확장자로 각각 변환하였다. 전체 데이터 전처리에 대한 요약 블록선도는 아래 그림과 같다.

8) AI Hub, URL : <https://www.aihub.or.kr/>



<그림13 - 데이터 처리 과정 블록 선도>

전처리 과정은 python을 이용하여 진행하였고 점역 과정에서는 기존 프로그램을 사용하기로 했다. 점사랑&하상브레일은 역점역 시 문제가 발생하였고, 점역 과정은 <one to one> 과정이기 때문에 축약법칙까지 반영된 올바른 점자를 얻을 수 있었다.



<그림14 - 하상브레일의 점역 수행>

하지만, 점자를 바로 우리가 원하는 상태로 얻을 수 없었다. 하상브레일과 점사랑은 점자를 BRF(Braille Ready Format) 변환하여 사용하고 있음을 알게 되었다.

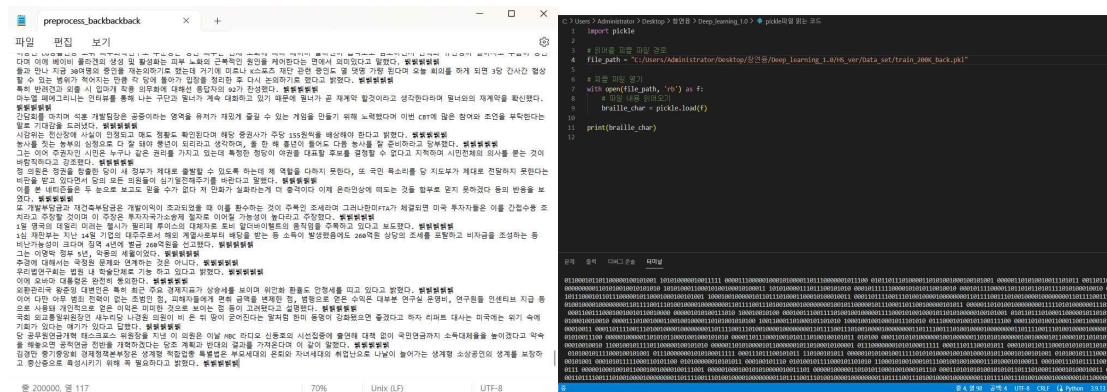
BRF 조판법은 점자 관련 소프트웨어에서 사용하는 방식으로 시각장애인이 접하는 점자를 읽기 쉽고 효율적으로 만들기 위해서 다른 글자(영문, 숫자)를 점자처럼 표기하는 방식이다.<sup>9)</sup> 기존의 점자 폰트는 점자 판독 속도를 높이기 위해 글자를 단순화하고 압축하는 등의 방법을 사용했지만, 이는 점자로 표현되는 글자의 의미 전달을 어렵게 만들 수 있다. BRF 조판법은 이러한 문제를 해결하기 위해, 시각 장애인이 손쉽게 읽을 수 있는 자연스러운 문장을 만들기 위한 폰트 디자인 및 조판 기술을 개발한 것이다.

9) 점자번역기, URL: <https://t.hi098123.com/braille>



<그림15 - BRF 변환을 풀기 위한 디렉터리>

우리는 BRF 변환을 우리가 원하는 형태인 6자리의 2진수로 표현할 수 있게 되었고 [기존 문자 - 점자] 이 두 가지의 데이터 쌍을 얻을 수 있었다.



<그림 16 - [전처리 된 문자(좌측 하단) - 그에 대응되는 점자(우측 하단)] 데이터 쌍>

BRF를 점자로 바꾸는 과정에서 줄 바꿈을 해결하지 못하여 임의의 단어를 전처리 된 문자 끝에 추가하여 줄 바꿈의 역할을 할 수 있도록 하여 문자열 txt파일과 점자 pickle 파일 형태의 최종 데이터 쌍을 제작하였다. pickle파일을 사용하게 된 이유는 데이터의 용량을 줄이게 하여 Python 내부에서 편리하게 다루기 위함이다.



## 2-4. BART 모델 학습

### 2-4-1. 학습 준비

모델 학습을 위해 우리는 BART의 모델 중 한국어를 중심으로 다루는 koBART를 사용하고자 한다. koBART<sup>10)</sup>란 SKT에서 공개한 한국어 BART 모델이며, Text Infilling 노이즈 함수를 사용하여 40GB 이상의 한국어 텍스트에 대해서 학습한 한국어 encoder-decoder 언어 모델이다. 우리는 우선 모델의 base 버전을 분석하였다. 6개의 encoder와 decoder로 구성되어 있는 형태이며, 우리가 제작한 두 가지의 데이터 쌍을 학습 시켜 보았다.

학습을 진행하려면 몇 가지 설정을 해야 한다. train 데이터를 계산되는 데이터의 개수를 정하는 batch\_size, 학습에서 train 데이터를 모두 소진했을 때의 횟수를 의미하는 epochs, 학습 시에 최대로 인식할 수 있는 길이를 의미하는 max\_length, gradient의 보폭을 말하는 learning rate 줄여서 lr, 학습 과정에서 lr을 조정해주는 스케줄러인 lr\_scheduler 등이 있다 그리하여 우리가 이번 학습에 사용한 Hyper Parameter는 다음과 같다.

<표3 - 학습에 사용된 Hyper Parameter>

Hyper Paramiter	
batch size	32
epochs	13
max_length	1024
lr	3.00E-05
lr_scheduler	CosineAnnealingLR

### 2-4-2. 학습 과정

koBART 모델의 base 버전으로 학습한 결과 성능이 좋지 않았다. 우리는 그 이유가 koBART라는 모델에서의 encoder는 원래 문자를 받아서 학습 받는 구조였지만 우리는 점자, 즉 바이너리의 형태를 encoder에 들어가 학습하다 보니 문제가 생긴 것 같아 몇 가지 작업을 진행하였다. 첫 번째로 기존의 koBART의 Tokenizer의 vocabulary에는 한글, 영어, 이모티콘 등 다양한 단어들이 추가되어 있지만 점자가 없기에 우리가 사용하는 64가지의 점자 형태를 vocabulary에 추가하였다. 그 후 encoder의 layer를 6개가 아닌 4개로 2개를 줄여 학습을 진행하였다.

결과는 성공적이었다. 처음 시도했던 base 모델보다 훨씬 좋은 성능을 보였다, 우리는 이 성능의 발전이 encoder의 개수를 줄여서인지, 우리의 바이너리를 tokenizer vocab에 추가하여 발전했

10) SKT, GitHub, URL: <https://github.com/SKT-AI/KoBART>

는지 확인하기 위해 마지막 모델인 encoder와 decoder는 6개의 layer를 사용하고 tokenizer에 추가한 형태로 학습을 진행하였다.

총 우리가 진행한 학습 모델을 정리하면 아래와 같다.

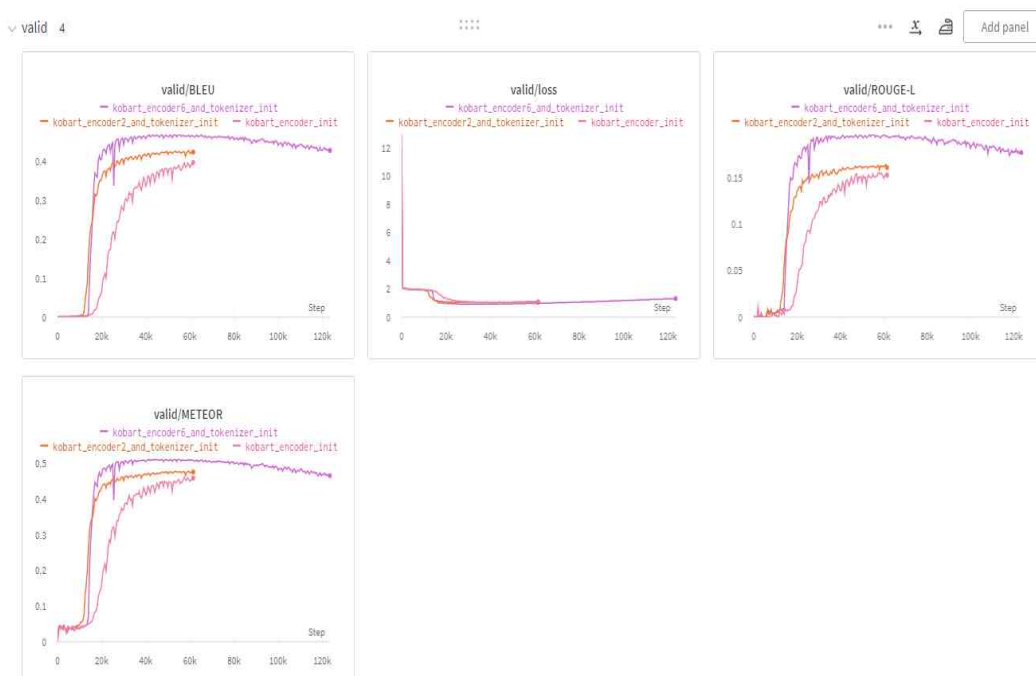
**학습 모델 1.** 기존의 koBART base 모델

**학습 모델 2.** koBART에서 encoder layer를 2개 사용하고, 점자 바이너리 tokenizer에 vocab 추가

**학습 모델 3.** koBART에서 encoder layer를 6개 사용하고, 점자 바이너리 tokenizer에 vocab 추가

## 2-4-2. 학습 모델 성능 비교

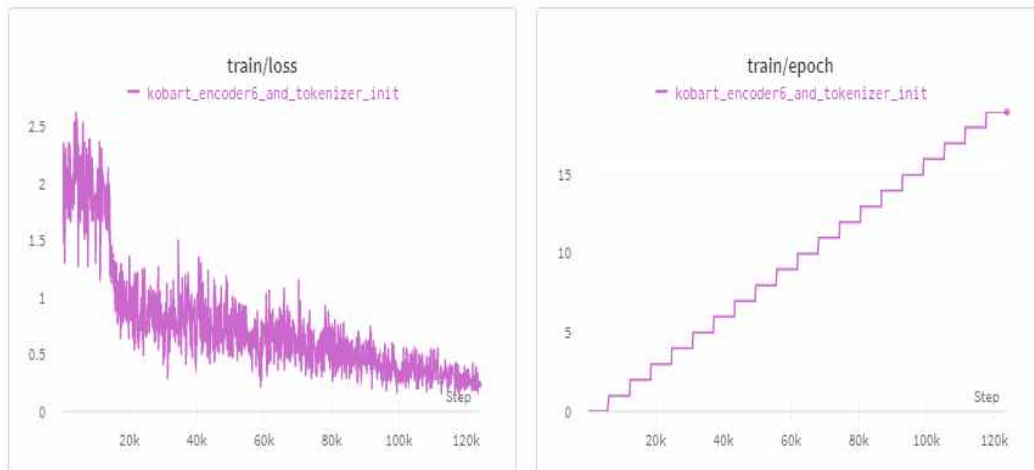
우리는 총 3가지의 모델을 학습시켰고, 그 중 가장 성능이 좋은 모델을 최종 프로젝트의 결과물로 선정하고자 했다. 모델의 학습 성능 지표로 선정한 것들은 다음과 같다. 우리의 bart 모델은 번역과 흡사한 작업으로 볼 수 있으며 자동 생성 된 요약문이 얼마나 reference summary와 일치하는지에 대한 근사치를 제공하는 3가지의 Evaluation Metric을 사용했다. 첫 번째로는 가장 대표적인 BLEU (Bilingual Evaluation Understudy), 두 번째로는 ROUGE (Recall Oriented Understudy for Gisting Evaluation), 마지막으로 METEOR (Metric for Evaluation of Translation with Explicit Ordering)를 사용하여 각 모델들의 score를 비교하여 최종 모델을 선정하고자 한다. 다음은 성능 비교 결과의 monitoring 자료이다.



<그림 17 - 학습 모델 성능 평가 지표>11)

11) wandb, URL: <https://wandb.ai/>

윗 그림에서 보이는 보라색은 encoder & decoder가 6개의 layer이고 tokenizer에 vocab을 수정한 모델이고, 주황색은 encoder 2개 decoder 6개에 tokenizer에 vocab을 수정한 모델이다, 마지막으로 분홍색은 기존의 koBART base 모델을 사용한 것이다. 이 그림을 통하여 알 수 있듯이 모든 성능 지표에서 학습이 지남에 따라 보라색이 앞서는 것을 확인할 수 있었다. 이때 사용한 validation data는 train data의 1%만 사용하였다. 그리하여 우리는 최종 학습 결과 모델을 encoder & decoder가 6개의 layer이고 tokenizer에 vocab을 수정한 모델로 선정하였고 학습을 끝까지 진행하였다.



<그림18 - 최종 선정된 모델 학습>

시간이 지남에 따라 train/loss가 0에 수렴하였으므로 학습이 잘 되었음을 알 수 있었다.

## 2-5. 기존 프로그램과의 성능 비교

역점역 기능을 제공하는 기존 프로그램인 점사랑과 우리 팀의 프로그램 성능을 비교하기 위해, 총 4가지 조건과 테스트할 문장을 아래와 같이 설정하였다.

조건 1. 한글로만 구성된 문장.

조건 2. 한글과 영어로 구성된 문장.

조건 3. 한글, 영어 숫자가 모두 포함된 문장.

조건 4. 축약법칙이 반영된 문장.

조건 1을 시작으로, 각각의 조건에 대한 점자 문장의 역점역 결과를 기존 프로그램인 하상브레일, 점사랑과 서로 비교하여 상호 간의 유의적인 차이가 있는지 검토하였다. 또한, 검토 결과를 바탕으로 기존 프로그램 대비 우리 팀이 제작한 프로그램의 차별화 된 장점을 각각의 표 아래 부분에 제시하였다.



<표 4 - 조건 1의 대한 성능 비교>

조건 1	한글로만 구성된 문장.
테스트 문장	대검관계자는 피의 사실도 공표하면 안 되는데 감찰은 일종의 내사 단계여서 더욱 말하기 어렵다고 말했다.
테스트 문장의 점역	대검관계자는 피의 사실도 공표하면 안 되는데 감찰은 일종의 내사 단계여서 더욱 말하기 어렵다고 말했다.
기존 프로그램	대검관계자는 피의 사실도 공표하면 안 되는데 감찰은 일종의 내사 단계여서 더욱 말하기 어렵다고 말했다.
우리 팀의 프로그램	그는 피의 사실도 공표하면 안 되는데 감찰은 일종의 내사 단계여서 더욱 말하기 어렵다고 말했다.

기존 프로그램의 경우, 마지막 온점을 제외하고는 완벽한 역점역 결과를 보여주고 있음이 확인되었다. 하지만 우리 팀의 경우, 온점은 정확하게 반영했지만 ‘대검관계자’와 같은 고유명사를 반영하지 못했다. BART 모델의 특성상 문장 생성 및 요약의 성질을 가진 모델이기에 정확한 직책 대신 ‘그는’으로 반영되어 역점역시 정확도가 높게 나오지 못함을 알 수 있다.

<표 5 - 조건 2의 대한 성능 비교>

조건 2	한글과 영어로 구성된 문장.
테스트 문장	경기 지역에서의 A형용 백신 접종은 유전자 분석과 검토가 끝나면 바로 실시할 것이라고 말했다.
테스트 문장의 점역	경기 지역에서의 A형용 백신 접종은 유전자 분석과 검토가 끝나면 바로 실시할 것이라고 말했다.
기존 프로그램	경기 지역에서의 A형용 백신 접종은 유전자 분석과 검토가 끝나면 바로 실시할 것이라고 말했다.
우리 팀의 프로그램	이 A형 백신 접종은 유전자 분석과 검토가 끝나면 바로 실시할 것이라고 말했다.

기존 프로그램의 경우, 한글과 영어가 혼합되어 있을 시에 영어 텍스트를 아예 반영하지 못한다. 또한, 마지막 온점도 조건 1에서의 결과와 같이 종성 ‘ㅂ’으로 역점역 한다. 우리 팀의 프로그램 결과는 온점을 정확하게 반영하였고, 영어 단어도 정확히 역점역이 가능했다. 하지만, BART 모델의 특성상 경기 지역을 ‘이’로 요약 하는 오역이 발생하는 역점역의 한계를 볼 수 있다.

<표 6 - 조건 3의 대한 성능 비교>

조건 3	한글, 영어 숫자가 모두 포함된 문장.
테스트 문장	IBK투자증권 연구원은 갤럭시의 조로화가 예상보다 빠르게 진행 중이라고 진단하면서 3분기 IM부문 영업이익은 2조 7000억원으로 악화될 것이라고 추정했다.
테스트 문장의 점역	IBK투자증권 연구원은 갤럭시의 조로화가 예상보다 빠르게 진행 중이라고 진단하면서 3분기 IM부문 영업이익은 2조 7000억원으로 악화될 것이라고 추정했다.
기존 프로그램	IBK투자증권 연구원은 갤럭시의 조로화가 예상보다 빠르게 진행 중이라고 진단하면서 3분기 IM부문 영업이익은 2조 7000억원으로 악화될 것이라고 추정했다.
우리 팀의 프로그램	증권 연구원은 갤럭시의 조로화가 예상보다 빠르게 진행 중이라고 진단하면서 3분기 IM부문이익은 2조 7000억원으로 악화될 것이라고 추정했다.

기존 프로그램의 경우, 조건 2와 같이 한글과 영어가 혼합되어 있는 경우 영어를 아예 반영하지 못함을 볼 수 있고, 조건 1과 같이 문장부호인 마침표도 역점역에 반영하지 못함을 알 수 있다. 우리 팀의 프로그램은 한글, 영어, 숫자 모든 부분을 정확하게 역점역했으나 BART 모델의 특성인 'IBK투자'라는 고유명사를 '증권'으로 요약하는 우리 팀의 프로그램의 한계를 확인할 수 있었다.

<표 7 - 조건 4의 대한 성능 비교>

조건 4	축약법칙이 반영된 문장.
테스트 문장	1895년 일본 청나라가 맺은 조약은 뎡진 조약이다.
테스트 문장의 점역	1895년 일본 청나라가 맺은 조약은 뎡진 조약이다.
기존 프로그램	1895년 일본 청나라가 맺은 조약은 뎡진 조약이다.
우리 팀의 프로그램	1895년 일본 청나라가 맺은 조약은 뎡진 조약이다.

마지막으로 조건 4는 우리가 해결하고자 했던 축약법칙에 대한 성능 비교 자료이다. 다음은 한글 점자 규정 해설서 제3절 7.8항의 내용이다.

### 제3절 모음자

#### 제7항

기본 모음자 'ㅏ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㅣ'는 다음과 같이 적는다.

ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ
ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ

제8항

그 밖의 모음자 ‘ㅐ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㅣ’는 다음과 같이 적는다.

ㅐ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ
ㅐ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ

<그림19 - 한글 점자 규정 해설서 제3절 7,8항 내용>

축약법칙 중 모음자에 대한 내용으로 정해진 규정에 따라 위 그림에서 볼 수 있는 모음자는 해당되는 점자로 표기해야한다는 규정의 내용이다. 조건 4에 해당하는 성능 비교 자료를 보면 기존 프로그램의 경우 축약법칙이 반영되어 “텐진 조약”이라는 단어를 정확하게 역점역 하지 못함을 볼 수 있다. 하지만 우리의 프로그램은 축약법칙 또한 학습했기 때문에 테스트 문장과 동일한 결과를 도출 할 수 있었다.

## 2-6. 성능 비교 결과 평가

위의 2-5의 비교 결과를 종합하여 평가하면, 영어 텍스트와 문장 부호 그리고 축약법칙이 반영된 문장의 역점역 정확도는 기존 프로그램보다 확실히 높다. 하지만 기존의 문장과 의미가 완벽하게 일치하는 역점역은 현재 달성하지 못하고 있다. 이와 관련해서 다음과 같은 3가지 이유를 들어 추정했다.

일반적으로 BART같은 Transformer 모델은 많은 양의 데이터를 요구한다. 더욱이 본 프로젝트에서는 한글/영문/숫자 같은 다양한 언어가 모두 포함되어 있다. 더불어 본 학습에 사용된 한글-점자 데이터 쌍은 프로젝트를 위해 만든 데이터로, 완벽한 성능의 프로그램을 제작하기에는 20만 문장이라는 데이터의 양이 부족했음을 첫 번째 이유로 추정한다. 더욱 좋은 성능을 위해서는 우리가 사용한 한국 기사 데이터뿐만 아니라 초성, 중성, 종성으로 나누어진 말뭉치 데이터, 영어가 더욱 많이 포함된 말뭉치 데이터 등을 사용하면 더욱 더 높은 성능을 기대할 수 있다고 예측한다.

두 번째로는 역점역을 해결하기 위해서는 먼저 점역의 과정을 이해해야 한다. 하나의 단어가 점자로 변환되는 과정에서 예를 들어 “안녕”이라는 단어를 점자로 표현하면 “ㅇㅏㄴㄴㅇ”으로 자음 모음을 분리한 후 해당하는 점자로 1대1 대응하는 방식이다. 그러나 모든 점역 과정이 이렇게 되는 것이 아니라 문장의 길이가 길어짐을 방지하기 위한 축약법칙이 사용된다. BART와 같은 deep learning 모델의 학습에서 단어, 문장 단위로 학습을 진행하였기 때문에 임의의 단어가 몇 자리의 바이너리로 변환되는 규칙성을 반영하지 못하여 완벽한 성능의 역점역이 구현되지 않았다고 추정한다. 보다 높은 정확도의 프로그램을 제작하기 위해서는 위에서 제시한 방법을 반영해야 한다고 생각한다.

마지막으로 2-5에서 볼 수 있듯이 역점역의 세부적인 정확도는 떨어지지만, 전체적인 내용은 알맞게 역점역 되었다는 점을 확인할 수 있다. 이는 BART 모델의 특성상 noising 함수를 통해 텍스트를 오염시키고 Sequence to Sequence 모델이 원래의 텍스트를 복원하기 위해 학습하면서 문장을 예측 및 생성하며 요약하는 특징 때문에 나타나는 오류를 해결하기 위해서는 BART 모델의 구조를 점자 번역에 특화 시켜 모델의 encoder와 decoder의 layer와 parameter를 적절히 수정하는 과정을 거쳐야 한다고 예상한다.

### 3. 결론

#### 3-1. 프로젝트의 요약

본 프로젝트는 기존 프로그램의 1:1 대응 방식 역점역의 한계를 극복하고자, 새로운 메커니즘인 “Deep learning(기계학습)을 활용한 역점역 프로그램” 개발을 목표로 하였다. 한글, 문장 부호, 영어, 숫자, 그리고 축약법칙이 반영된 문장의 역점역 정확도를 개선하기 위해 BART 모델을 통한 학습을 진행하였고, 학습에 필요한 [문자 - 점자] 데이터 쌍은 직접 제작하여 학습에 사용했다. epoch에 따른 train/loss는 0에 수렴함을 보아 성공적으로 학습은 진행되었으며, 기존 프로그램과의 성능 비교를 통해 한글, 영어, 문장부호 그리고 축약법칙이 반영된 문장에서 기존 프로그램보다 확실히 높은 정확도를 가지는 프로그램 개발에 성공했다.

하지만 학습 모델의 구조적 한계와 질 좋은 데이터 양 부족 등의 요소로 인해 100% 정확한 역점역 수행은 불가능하였다. 후에 더 많은 데이터 확보와 모델의 특징이 가지는 부분 때문에 생기는 요약 부분, 기존 프로그램이 한글 문장에서는 정확도를 보이지만 영어나 문장부호가 나오는 부분에서의 한계를 가진다. 우리는 그 점을 Deep learning을 활용하여 해결했지만 학습을 할 때 모든 부분에 대한 학습이 아닌 문제가 생기는 부분만 지정해서 학습을 한다면 더 작은 용량으로 정확도 100%의 역점역 프로그램 개발할 수 있을 것이라 생각한다.

#### 3-2. 프로그램의 활용방안

우리는 기존의 역점역 프로그램보다 높은 성능의 프로그램을 개발하였고 보다 쉽게 사용하기 위해 web을 제작하였다. Python에서 제공하는 오픈 소스 라이브러리 streamlit<sup>12)</sup>을 사용하여 web을 제작하였고, 우리의 학습 데이터와 연결하여 점자에 해당하는 바이너리를 기입하면 그에 해당하는 한글로 번역되는 구조로 구성되어있다.

## 점자 번역기

모델 선택

trained\_bart\_braille\_6\_and\_toknizer\_init

Enter your binary braille:

```
110001010010100100010110000001101110001101 101100000010101010100100101011  
00010010110010000010001011110000110000010101110101010010000  
101010110000100100101010010100010011
```

## Result

우리는 국민브레일입니다.

<그림20 - 역점역 프로그램을 사용하기 위한 web>

별도의 하드웨어를 따로 구매하여 사용해야 하는 것이 아닌 소프트웨어로서 구현했기 때문에 기존의 노트북, 스마트폰 등에서 쉽게 사용할 수 있다. 소프트웨어 업데이트를 통해 잘못된 번역이나 오류 등을 쉽고 빠르게 수정할 수 있으며, 새로 추가해야 하는 외국어, 신조어 등에도 빠르게 대처할 수 있다.

12) streamlit URL: <https://streamlit.io/>

### 3-3. 사회적 측면에서의 기대 효과

후천적 시각장애인의 경우 시각 이외의 감각이 선천적인 장애인에 비해 예민하지 않아 점자를 새로 익히는 데에 큰 어려움을 겪고 있는 상황이다. 여기에 우리 팀의 역점역 프로그램의 기능에 점자를 촬영하여 점자를 인식하는 프로그램과 번역된 문자를 음성으로 송출해주는 TTS(Text to Speech) 기능을 추가한다면, 점자를 스스로 학습하는 데에 큰 도움이 될 수 있을 것으로 예상된다. 궁극적으로 후천적 시각장애인들의 빠른 사회생활 복귀에 도움이 되는 프로그램으로 자리 잡을 수 있을 것이다.

또한, 시각장애인들을 위해 작성한 창작물에 대한 접근이 쉬워지며, 시각장애인들은 자신의 생각을 더욱 자유롭고 편하게 표현할 수 있다. 우리 팀이 제작한 프로그램을 통해 시각장애인들의 사회 진출이 장려될 수 있으며, 궁극적으로 비장애인과 시각장애인 간의 사회적 융화를 이룰 수 있을 것으로 기대한다.