

Neural Machine Translation with mT5

TEAM : 2팀 나는 자연인이다.

김태웅, 설재민, 이재형, 이세린

CONTENTS

1

**Project
concept**

Introduction

-

Analysis
process

2

**Project
Team**

3

**Project
Process**

4

**Project
Result**

EDA

-

Model
Selection

-

Model
Improvement

5

**자체 평가
및 보완**

1

**Project
Concept**

Project concept

Introduction

영어 → 한국어 번역을 수행하는 Neural machine translation Project

사용 모델 : mT5 모델

기존 T5 모델을 다국어 코퍼스로 확장한 모델

Framework

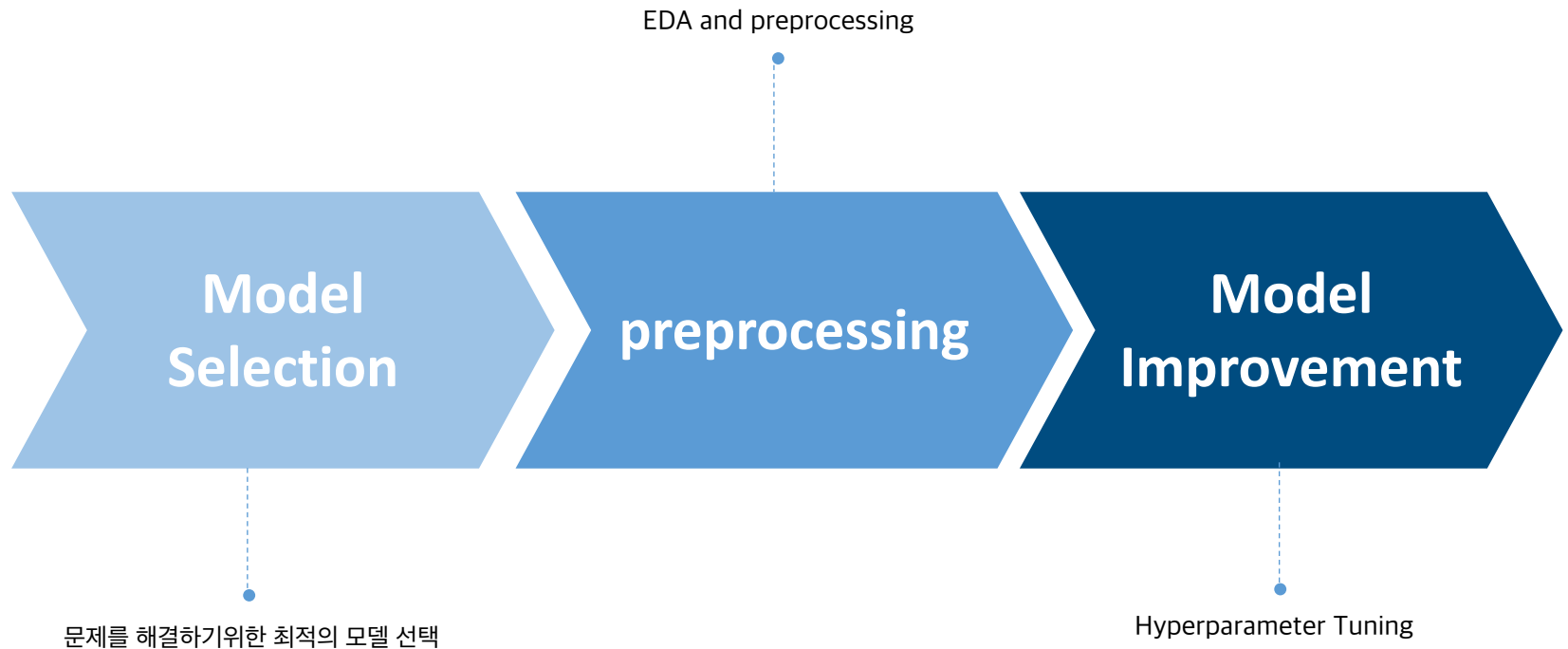


Library



Project concept

analysis process



2

**Project
Team**

Project Team

김태웅(팀장)

Project process 설계, base line code 작성 ,mT5, 시각화

설재민(팀원)

baseline code 수정, Bart 모델 적용

이재형(팀원)

baseline code 수정, Bart 모델 적용

이세린(팀원)

baseline code 수정, Bart 모델 적용

3

**Project
process**

Project Process

구분	기간	활동
Base Line 코드 및 데이터 분석	11/17 ~ 11/20	Base Line 코드 및 데이터 파악
새 Base Line 코드 작성 & 분석	11/21 ~ 11/23	새로운 Base Line 작성 및 분석
각자 모델 적용 & 수정	11/23 ~ 11/30	각각 모델 적용 및 수정
발표 자료 시각화 및 작성	11/30 ~ 12/01	최종 발표 자료 작성
총 프로젝트 진행 기간	11/17 ~ 12/01	

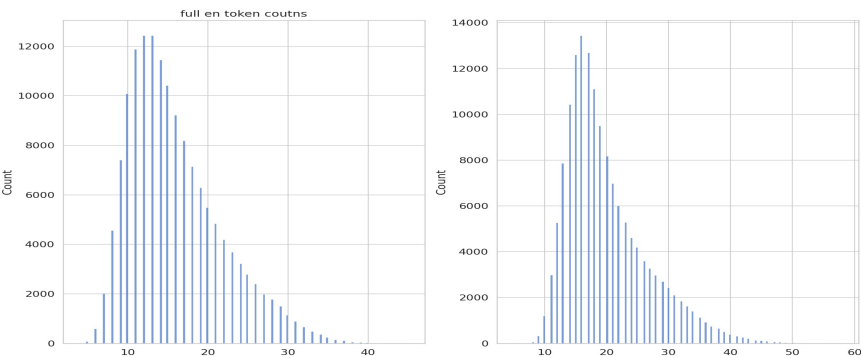
4

**Project
Result**

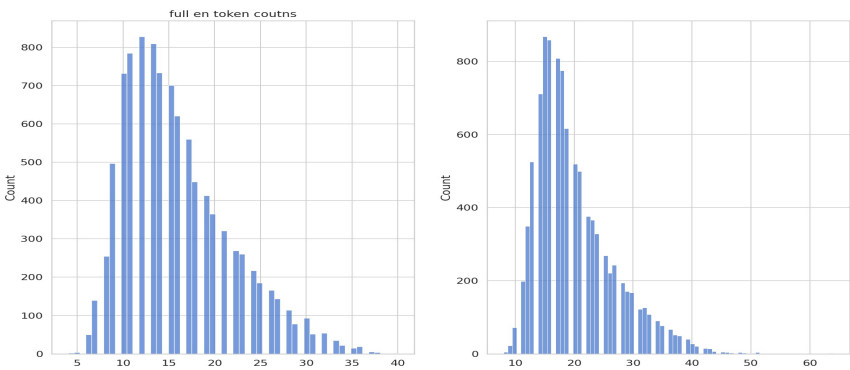
Project Result

EDA

Train Data



Dev Data



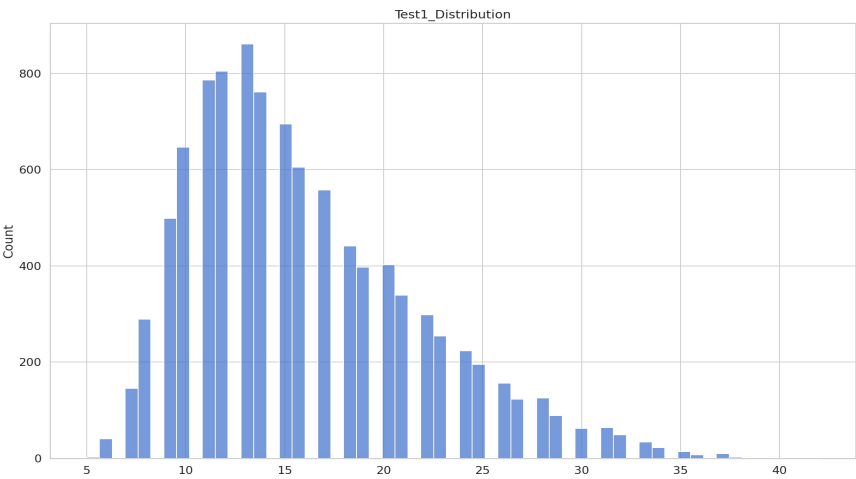
	sid	en	ko
count	150000	150000	150000
unique	150000	149901	149993

	sid	en	ko
count	10000	10000	10000
unique	10000	10000	10000

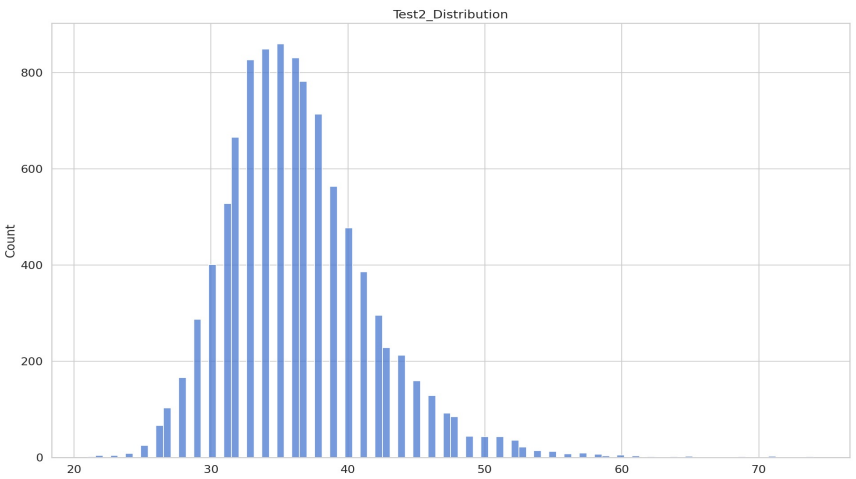
Project Result

EDA

Test1 Data



Test2 Data

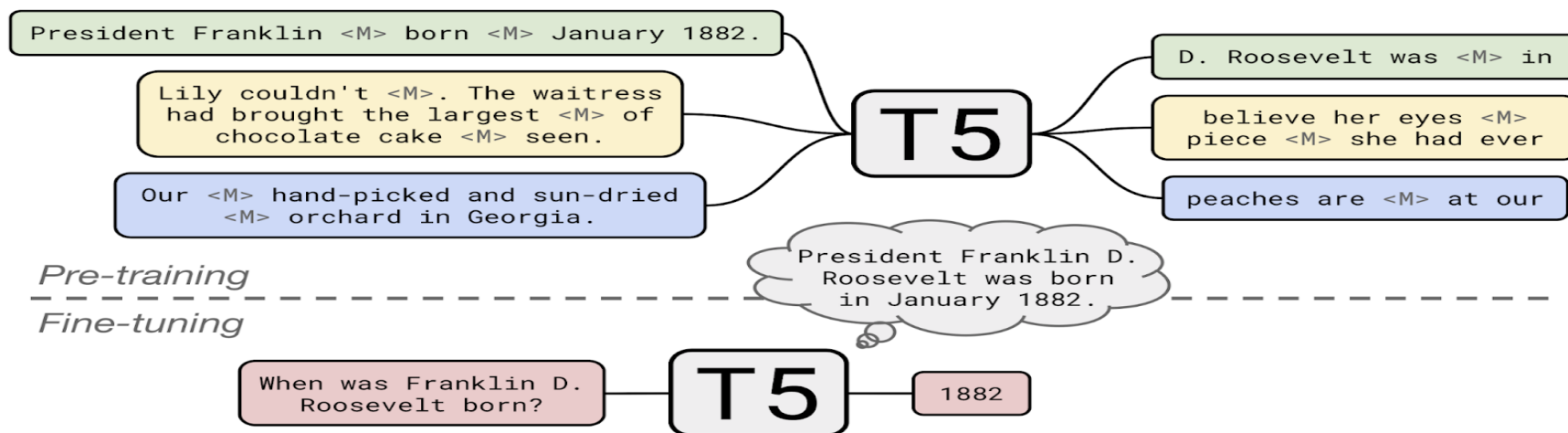


	sid	en
count	10000	10000
unique	10000	10000

	sid	en
count	10000	10000
unique	10000	9999

Project Result

Model selection(mT5) and analysis



기존 영어 위주로 연구되던 언어 모델 및 번역 모델을 다국어로 확장하려는 시도가 많이 이루어지고 있습니다. Google의 mT5는 기존 T5 (text-to-text transfer transformer)를 다국어 코퍼스로 확장한 연구로서, 총 101개국 언어를 포함한 데이터를 수집하고, 이를 이용하여 학습함으로써 cross-lingual language task에서 성능 개선을 이루었습니다.

Project Result

Model selection(mT5) and analysis

- **Hyperparameter**

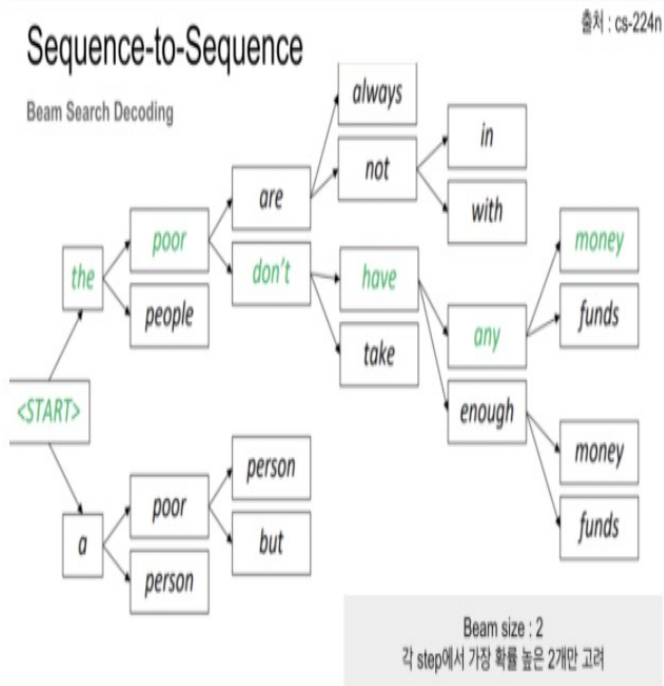
- data : train, dev, test, test2
- model : mT5
- parameter
 - input max_length : 60
 - output max_length : 60
 - learning_rate : 0.0005
 - epoch : 10, 13 (**early_stopping_callback**)
 - batch_size : 8 (**accumulate grad batches**)

Test Review

- Epoch 13에서 validation loss가 최저를 보임
- Colab의 메모리 부족으로 초기 batch_size의 크기는 8이며, 이를 개선하기 위해 accumulate grad batches를 사용함
- Learning_rate는 하이퍼파라미터 튜닝을 진행하지 못함
- EDA 결과 input max length, output max length를 60으로 설정

Project Result

Model selection(mT5) and analysis



Post-processing

Greedy Search: decision making을 early stage에서 하므로 초반에 틀리면 다 틀릴 지도 모른다는 단점

Exhaustive Search: 모든 가능성을 다 봄 → 가능한 가짓수가 time step에 비례하므로 문장이 조금만 길어져도, vocab size가 조금만 커져도 불가능

Beam Search: exhaustive search를 하되, 각 time step마다 가장 점수가 높은 top-k만 유지 → 가장 많이 쓰는 방식

Project Result

Model selection(mT5) and analysis

Model	num_beam	test1	test2
mT5-base	2	22.48	16.77
mT5-base	5	23.24	17.81
mT5-base (dev data add)	5	23.34	17.74
mT5-base	10	22.69	16.86

Test Review

- Num_beam은 5일 때 가장 성능이 좋음.
- Dev data를 train data에 추가해서 학습했을 때 test1의 성능이 향상됨. 하지만 test2의 성능은 좋아지지 않음
 - dev 데이터의 input sequence의 길이가 대부분 40 정도여서 test2의 성능 향상에 영향을 못 주는 것 같다.

Project Result

Model selection(mT5) and analysis

sid	en	ko	Predict
dev-9000	Therefore, the Hashima Island has a tragic his...	그렇기에 지옥 섬으로 불릴 정도로 하시 마섬은 우리나라의 아픈 역사를 가지고 있는 곳...	따라서 하사마섬은 우리나라의 비 극적인 역사를 가지고 있습니다.
dev-9001	I've been suffering in gastritis.	제가 위염에 걸려 고생을 좀 했습니다.	저는 위염으로 고생하고 있어요.
dev-9002	The school ground is used as the filming Site.	학교 운동장은 영화 촬영지로 이용되었다.	학교 땅은 촬영지로 사용됩니다.
dev-9003	This business is really important to me.	이 일은 저에게 정말 매우 중요해요.	이 사업은 저에게 매우 중요합니다.
dev-9004	A dog is sitting in front of the door.	개 한 마리가 문 앞에 앉아 있어요.	개가 문 앞에 앉아 있습니다.

Test Review

- 전체적으로 ko와 model이 예측한 결과값의 의미가 같다.
- Ko 와 model이 예측한 결과값의 의미는 같지만 어순, 어미 등에서 미세한 차이를 보인다.

Project Result

Model selection(T5) and analysis

- **Predict-Ko Training**

- 모델이 예측한 Predict와 ko의 어순 어미의 미세한 차이를 조정하기 위해 재학습
- Train data의 모델이 예측한 Predict 값을 추가
- Encoder에 Predict data, decoder에 ko 데이터로 t5 모델 학습
- 성능이 좋아지지 않음
 - train data의 전부를 predict data로 활용하지 못함

5

자체 평가
및 보완

자체 평가 및 보완

다양한 모델 활용하지 못함

이번 프로젝트에서는 mT5 외의 다른 모델을 테스트 해보지 못해 아쉬움이 남는다.

Gpu의 메모리 용량 제한

이전 프로젝트에서는 accumulate grad batch를 사용하지않아 batch size의 제약을 크게 받았는데 이번에는 accumulate grad batch의 사용으로 적은 memory에서 효율적으로 학습할 수 있었다.

HyperParameter Tuning

Learning_rate, epoch, Regularizaztion 등의 파라미터 튜닝을 다양하게 진행하지 못했다.

Back Translation 적용

시간상의 문제로 강의 시간에 배운 back translation을 적용해보지 못한 아쉬움이 남는다.

**THANK
YOU!**