NHN FORWARD

오픈소스로 쉽게 만드는 기계번역기

다룰 내용

1. 시작하기
2. 준비물
3. 만들기
4. 마무리

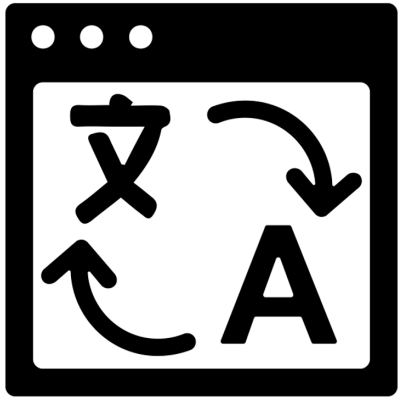
1. 시작하기

기계번역은?

NLP는 컴퓨터가 사람의 언어를 이해하는 목표를 가진다.

자연어는 생략과 중의성이 많아 컴퓨터가 이해하기 어렵다.

기계번역은 NLP에서 어려운 테스크에 속한다.



NLPは、コンピュータが人の言語を理解するという目標を持っています。

自然語は省略や中義性が多く、コンピュータに理解されにくい。

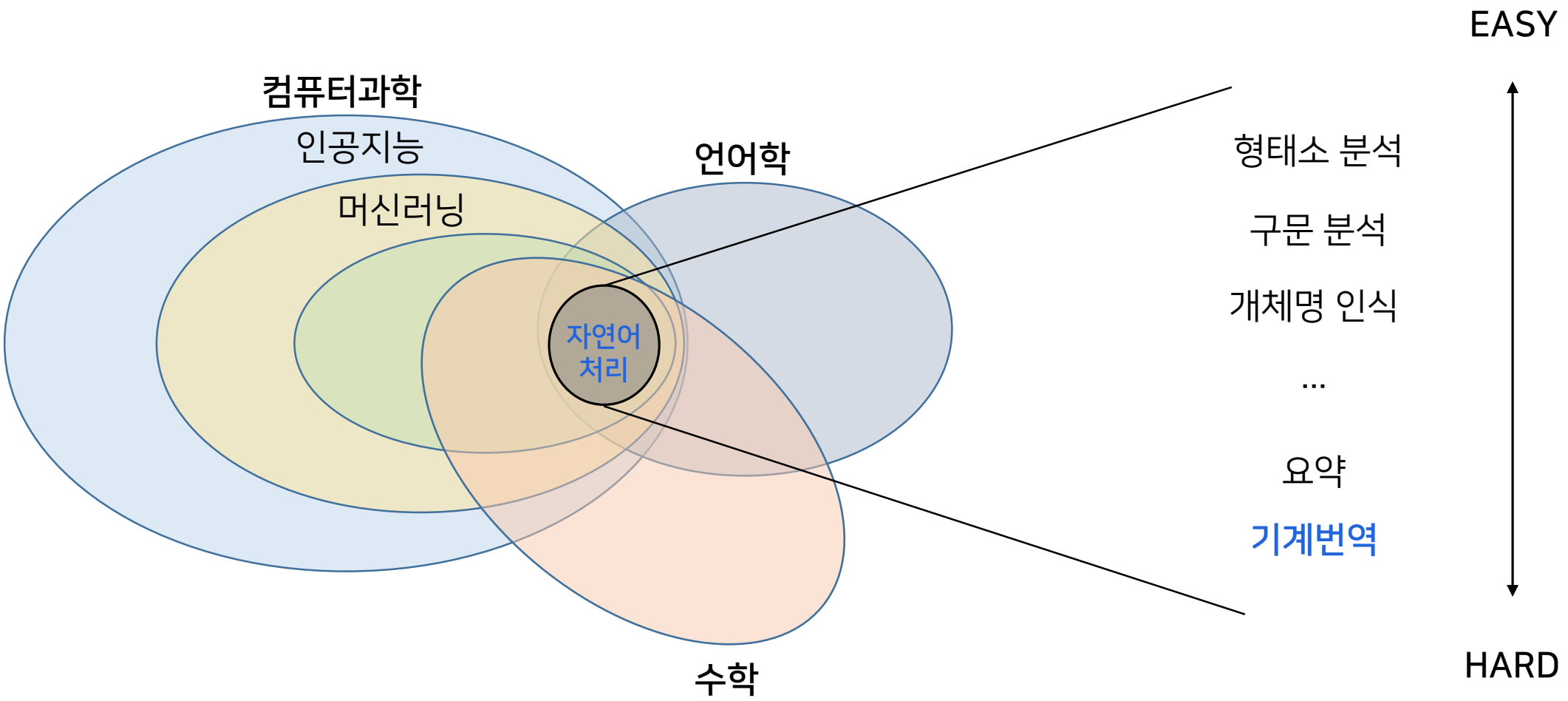
機械翻訳はNLPでは難しいテスクに属する。

Source

Target

그림출처: https://thenounproject.com/search/?q=translator&i=143961

기계번역은?


컴퓨터과학
인공지능
머신러닝
언어학
자연어
처리
수학
EASY
형태소 분석
구문 분석
개체명 인식
...
요약
기계번역
HARD

2. 준비물

오픈소스 – Fairseq

Name	Company	Framework	Github Stars
Transformers	HuggingFace	Pytorch, Tensorflow	52.6 k
Fairseq	Facebook	Pytorch	14.1 k
Tensor2tensor	Google	Tensorflow	11.6 k
OpenNMT-py	OpenNMT	Pytorch	5.3 k
Sockeye	Amazon	MXNet	1 k
Marian	Microsoft	C++	0.8 k

- WMT19 (WMT: Workshop on Statistical Machine Translation)
  Marian(30%), Fairseq(18%), OpenNMT-py(16%),
  Tensor2tensor(14%), Sockeye(14%), ···

학습 데이터

병렬 데이터 << 단일 데이터

한국어 문장 + 탭 + 일본어 문장 ...

bitext.ko-ja.tsv

- AI Hub 개방 데이터

한국어 문장 ...

mono.ko

- Wikipedia

일본어 문장 ...

mono.ja

3. 만들기

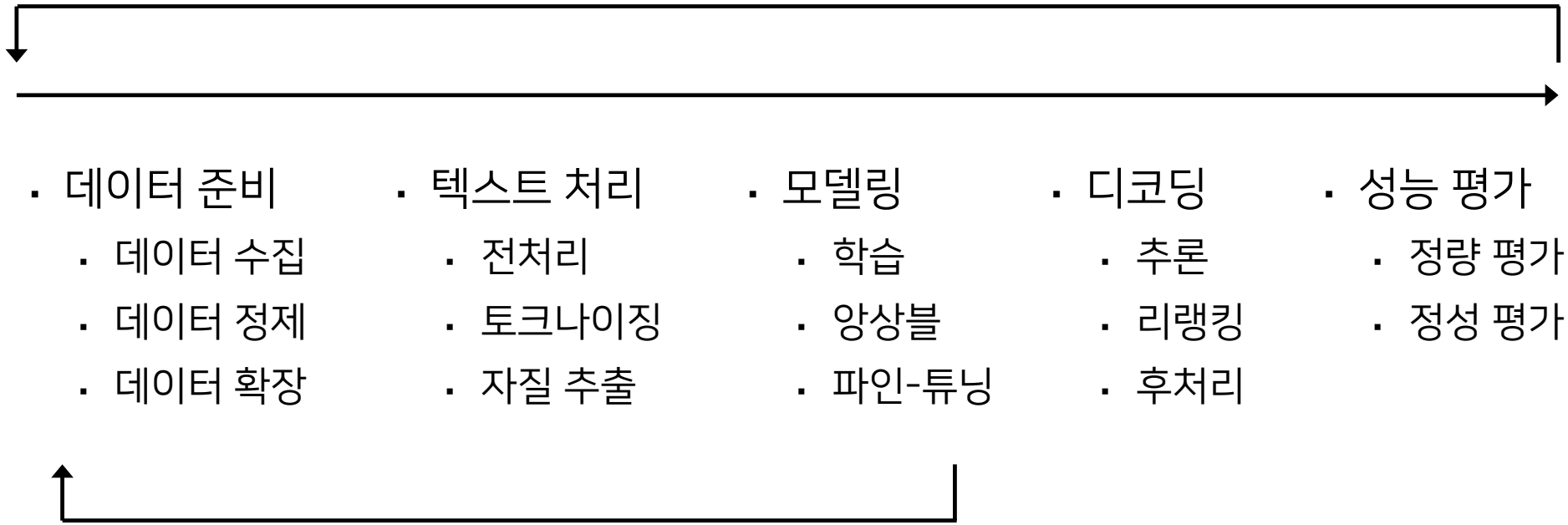
3.1. 개요

- 데이터 준비
  - 데이터 수집
  - 데이터 정제
  - 데이터 확장
- 텍스트 처리
  - 전처리
  - 토크나이징
  - 자질 추출
- 모델링
  - 학습
  - 앙상블
  - 파인-튜닝
- 디코딩
  - 추론
  - 리랭킹
  - 후처리
- 성능 평가
  - 정량 평가
  - 정성 평가

3.1. 개요

- 데이터 준비
  - 데이터 수집
  - 데이터 정제
  - 데이터 확장
- 텍스트 처리
  - 전처리
  - 토크나이징
  - 자질 추출
- 모델링
  - 학습
  - 앙상블
  - 파인-튜닝
- 디코딩
  - 추론
  - 리랭킹
  - 후처리
- 성능 평가
  - 정량 평가
  - 정성 평가

3.1. 개요


· 데이터 준비
· 데이터 수집
· 데이터 정제
· 데이터 확장
· 텍스트 처리
· 전처리
· 토크나이징
· 자질 추출
· 모델링
· 학습
· 앙상블
· 파인-튜닝
· 디코딩
· 추론
· 리랭킹
· 후처리
· 성능 평가
· 정량 평가
· 정성 평가

3.1. 개요

3.2. 준비	3.3. 학습	3.4. 결과

- 데이터 준비
  - 데이터 수집
  - 데이터 정제
  - 데이터 확장
- 텍스트 처리
  - 전처리
  - 토크나이징
  - 자질 추출
- 모델링
  - 학습
  - 앙상블
  - 파인-튜닝
- 디코딩
  - 추론
  - 리랭킹
  - 후처리
- 성능 평가
  - 정량 평가
  - 정성 평가

3.5 심화

3.2. 준비(1/2)

fairseq 설치

```
$ cd /home/

$ git clone https://github.com/pytorch/fairseq

$ cd fairseq

$ pip install --editable ./
```

Requirements

- python / pytorch

3.2. 준비(1/2)

데이터 준비

```
$ vi /home/raw_bitexta.to-ja.tsv

안녕          じゃあね
만나서 반가워          会えて嬉しいよ
한일 기계번역기입니다.          日韓の
機械翻訳機です。

...
```

bitext.ko-ja.tsv

텍스트 처리

✓ Train/Valid 분할

```
$ vi /home/train.ko

안 녕
만 나 서 반 가 워
...

$ vi /home/train.ja

じ ゃ あ ね
会 え て 嬉 し い よ
...
```

train.ko , train.ja

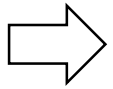
```
$ vi /home/valid.ko

뉴 스
...

$ vi /home/valid.ja

ニ ュ ー ス
...
```

valid.ko , valid.ja

3.2. 준비(2/2)

`fairseq-preprocess`

```
$ fairseq-preprocess \
        --source-lang ko \
        --target-lang ja \
        --trainpref /home/train \
        --validpref /home/valid \
        --thresholdtgt 3 \
        --thresholdsrc 3 \
        --destdir /home/data-bin \
        --workers 8
```



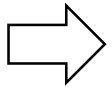
```
$ ls -1a /home/data-bin/
.
..
dict.ja.txt
dict.ko.txt
preprocess.log
train.ko-ja.ja.bin
train.ko-ja.ja.idx
train.ko-ja.ko.bin
train.ko-ja.ko.idx
valid.ko-ja.ja.bin
valid.ko-ja.ja.idx
valid.ko-ja.ko.bin
valid.ko-ja.ko.idx
```

- 데이터를 바이너리로 변환. 딕셔너리 구축.
- 빠른 로딩 속도와 압축 효과

3.3. 학습(1/3)

`fairseq-train`

```
$ fairseq-train \
        /home/data-bin \
        --source-lang ko \
        --target-lang ja \
        --arch transformer \
        --max-sentences 50 \
        --optimizer adam
        --save-dir /home/my-model \
```



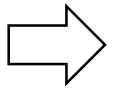
```
$ ls -1a /home/my-model/
.
..
checkpoint1.pt
checkpoint2pt
checkpoint3.pt
      ...
checkpoint64.pt
checkpoint65.pt
checkpoint66.pt
checkpoint_last.pt
```

- 학습 파라미터: https://fairseq.readthedocs.io/en/latest/command_line_tools.html#fairseq-train
- 모델 아키텍쳐(--arch): https://github.com/pytorch/fairseq/tree/main/fairseq/models

3.3. 학습(2/3)

앙상블(Ensemble) – Checkpoint Averaging

```
$ python /home/fairseq/scripts/average_checkpoints.py \
        --inputs /home/my-model/checkpoint* \
        --outputs /home/my-model/checkpoint.avg10.pt
```



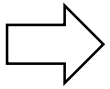
```
$ ls -1a /home/my-model/
.
..
checkpoint1.pt
checkpoint2pt
checkpoint3.pt
      ...
checkpoint64.pt
checkpoint65.pt
checkpoint66.pt
checkpoint_last.pt
checkpoint.avg10.pt
```

- 마지막 n개 checkpoint의 weight들을 평균화.
- https://github.com/pytorch/fairseq/blob/main/scripts/average_checkpoints.py

3.3. 학습(3/3)

파인-튜닝(Fine-tuning)

```
$ fairseq-train \
        /home/ft-data-bin \
        --source-lang ko \
        --target-lang ja \
        --arch transformer \
        --max-sentences 50 \
        --optimizer adam
        --save-dir /home/my-model/ft
        --reset-dataloader \
        --reset-optimizer \
        --restore-file /home/my-model/checkpoint.avg10.pt
```



```
$ ls -1a /home/my-model/ft
.
..
checkpoint1.pt
checkpoint2pt
checkpoint3.pt
      ...
checkpoint12.pt
checkpoint13.pt
checkpoint14.pt
checkpoint_last.pt
```

- checkpoint.avg10.pt 에서 재학습.

3.3. 결과(1/2)

`fairseq-interactive`

- Raw Text 번역

```
$ fairseq-interactive \
        --path /home/my-model/checkpoint.avg10.pt /home/my-model \
        --source-lang ko --target-lang ja \
        --beam 5 \
        --bpe characters
```

`fairseq-generate`

- Pre-processed Text 번역 & 배치 모드

```
$ fairseq-generate \
        --path /home/my-model/checkpoint.avg10.pt \
        --source-lang ko --target-lang ja \
        --max-sentences 50 \
        --beam 5
```

3.3. 결과(2/2)

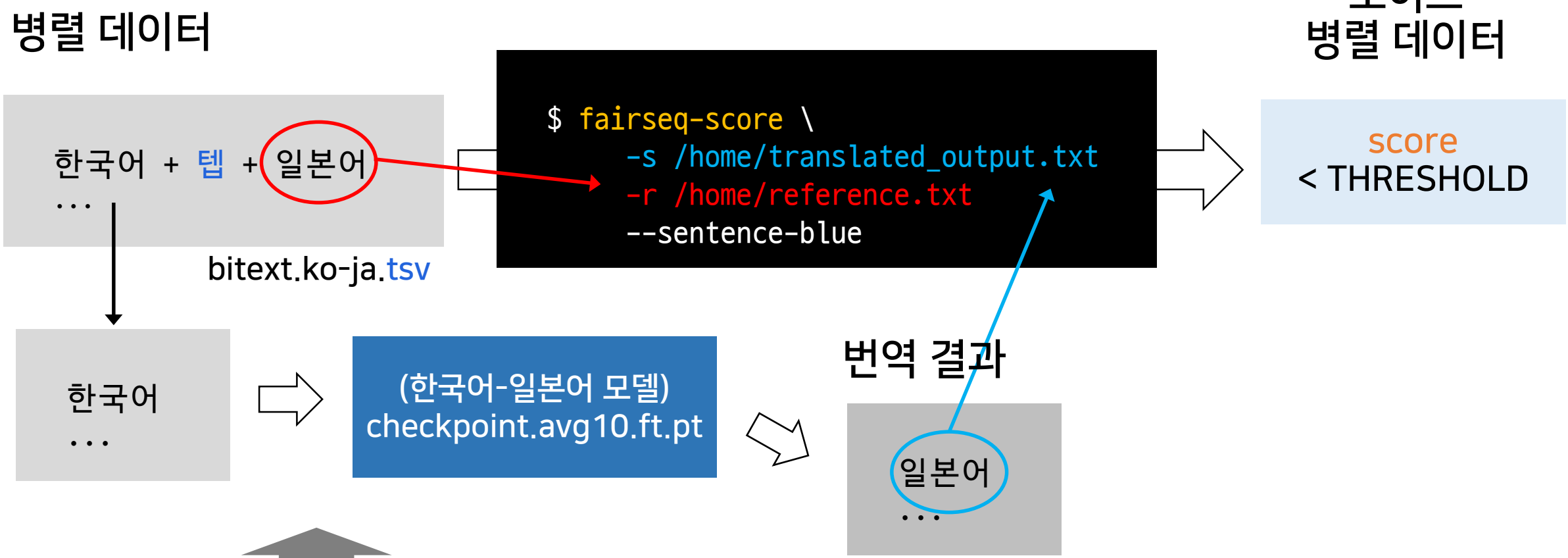
`fairseq-score`

```
$ fairseq-score \
        -s /home/translated_output.txt
        -r /home/reference.txt
        --sentence-blue
```

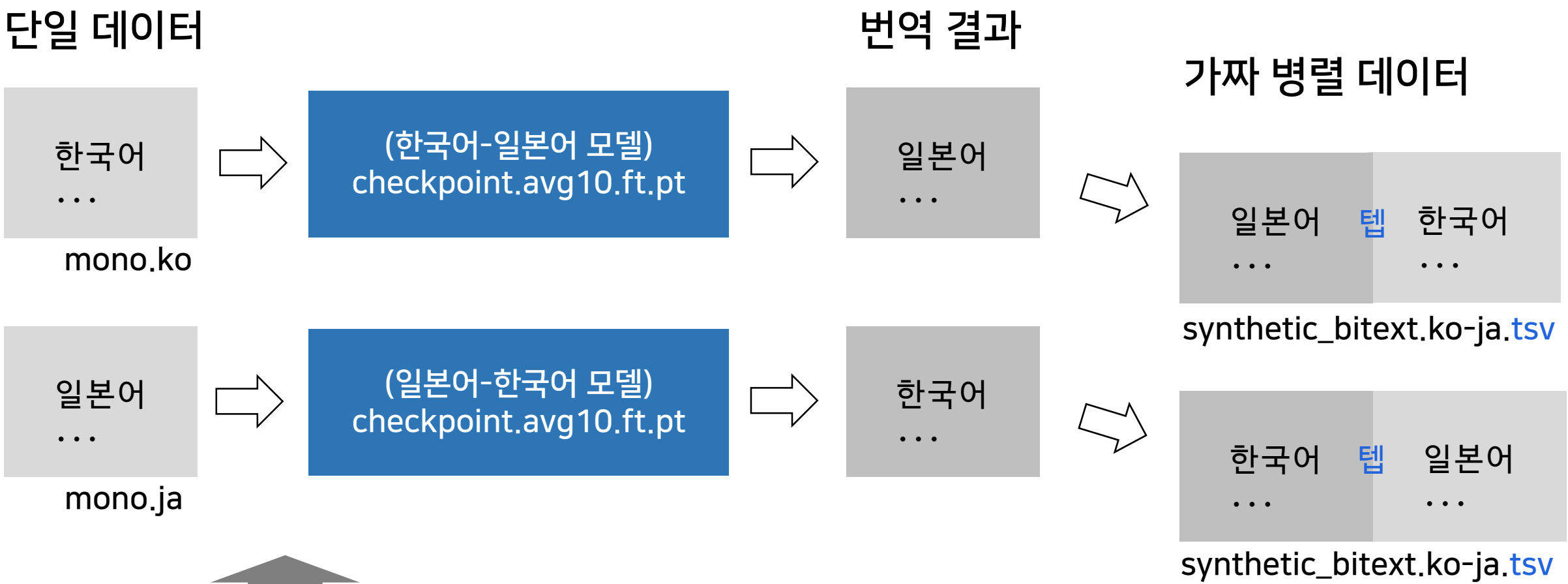
BLEU Score

- BLEU(Bilingual Evaluation Understudy)
- 기계번역 정량 평가 지표
- n-gram 기반 점수 측정 & 페널티(중복 토큰, 짧은 문장 길이)

3.4. 심화(1/2) – 데이터 정제



3.4. 심화(2/2) – 데이터 확장(back-translation)



4. 마무리

팁

- 주요 Fairseq 링크
  - 명령어 사용법: https://fairseq.readthedocs.io/en/latest/command_line_tools.html
  - 명령어 파이썬 코드: https://github.com/pytorch/fairseq/tree/main/fairseq_cli
  - 모델 리스트: https://github.com/pytorch/fairseq/tree/main/examples
- 체크사항
  - (학습) 멀티 & 분산 GPU 지원 ,(추론) 단일 GPU
  - GPU 메모리 관련 파라미터 설정 주의(사전 크기, 배치 크기 등)
  - Fairseq 버전 확인
  - Torchscript 지원
- 번역 품질
  1) 데이터 수집 + 데이터 정제
  2) 모델 + 규칙
  3) 정량 평가 + 정성 평가

정리

기계번역은 자연어처리 중에서 어려운 태스크

오픈소스로 누구나 쉽게 개발 시작

Fairseq는 기계번역에서 활발히 사용되는 오픈소스

CLI만으로도 충분히 기본형 기계번역 개발 가능

NHN FORWARD