

# LyriKOR: 음절을 맞춘 영한 노래 가사 번역 모델

조혜진<sup>1</sup>\*, 홍은빈<sup>1</sup>, 오지민<sup>1</sup>, 박정환<sup>2</sup>, 이병준<sup>1</sup>  
고려대학교 컴퓨터학과<sup>1</sup>  
고려대학교 사이버방학과<sup>2</sup>

{jhaenim, mulberry3280, min03, junghwark, byungjunlee}@korea.ac.kr

## LyriKOR: English to Korean Song Translation with Syllabic Alignment

Hyejin Jo<sup>1</sup>\*, Eunbeen Hong<sup>1</sup>, Jimin Oh<sup>1</sup>, Junghwan Park<sup>2</sup>, Byungjun Lee<sup>1</sup>  
Korea University, Department of Computer Science <sup>1</sup> Korea University, Department of Cyber Defense <sup>2</sup>

### 요약

세계화가 진행됨에 따라 다양한 문화의 음악을 즐기는 사람들이 늘어나고, 해외 팬들이 외국 노래를 이해하고 따라 부를 수 있는 접근성을 확보하는 것이 중요해졌다. 이를 위해 본 논문에서는 노래 가사 데이터에 특화된 영어-한국어 번역 모델 리리코(LyriKOR)를 제시한다. 리리코는 영어 노래를 한국어로 번역하여 그 의미를 담아낼 뿐만 아니라, 번역 결과물이 원곡의 선율과 리듬에 어느 정도 부합하도록 하여 한국어로 바로 따라 부를 수 있도록 하는 것을 목표로 한다. 이를 위해 번역과 음절 조정의 두 단계(two-stage)를 거쳐 제한된 데이터로 음절 정렬된 번역 모델을 훈련하는 새로운 방법을 소개한다. 모델 코드는 여기에서 볼 수 있다.

**주제어:** 노래 번역, 음절 제약, 기계 번역

## 1. 서론

글로벌리즘(globalism)이 진행되고 외국 음악의 인기가 높아지면서, 국내에서도 많은 사람들이 해외 아티스트들의 음악을 즐겨 듣고 있으며 이에 따라 가사 번역에 대한 수요가 증가하고 있다. 디즈니 노래나 아이들의 자장가와 같은 일부 분야에서 한국어 더빙의 대중화가 성공적으로 이루어졌지만, 대중 음악을 더 큰 규모로 번역하려는 시도는 제한적이었고 전문가들의 도움이 필수적이었다.

노래 가사 번역은 다음 두 가지를 충족해야 한다. 첫째, 출발어(source language)로 작성된 노래의 의미를 도착어(target language)에 정확하게 담아내야 한다. 즉, 원래의 리듬과 선율에 맞추기 위해 노래의 본질적인 의미는 건드리지 말아야 한다는 것이다. 둘째, 번역된 결과물은 원곡과 유사한 리듬적 구조를 가져야 한다. 만약 출발어가 5음절인데 도착어가 10음절이었다면 아무리 의미가 동일하더라도 잘 된 번역이라고 부르기 어렵다. 일반적인 상황에서, 리듬 구조의 정렬과 의미의 일치 사이를 적절하게 조절하는 문제는 꽤나 까다롭다[1, 2]. 따라서 노래 번역 작업을 성공적으로 완수하기 위해서는 모델이 번역의 의미와 음절 수 두 가지의 요소를 동시에 고려할 수 있어야 한다.

TV 프로그램이나 애니메이션 영화 같은 매체의 더빙 영역에서 유사한 문제를 해결하기 위한 선행 연구[1, 2]가 이루어졌지만, 노래 번역은 문장의 길이가 상대적으로 짧고 음절 정렬의 중요성이 높을 뿐더러 오디오나 시각 자료와 같은 멀티모달(multi-modal) 입력이 필요하지 않기 때문에 이와 차이가 있다. 따라서 우리는 노래 번역의 문제를 해결하기 위해, 텍스트

데이터 이외의 자료 없이 오직 음절 수와 가사만을 기반으로 하는 모델을 제시한다.

리리코는 크게 두 가지에 대해 기여한다:

- 음절 조정 모델을 훈련시키기 위한 프록시(proxy) 데이터셋을 만들어 노래 번역을 위한 페어링된 데이터의 부족을 해결했다.
- 음절 제약이 있는 노래 가사 번역 문제를 해결하기 위해 2단계의 새로운 접근 방식을 제안했다. 이 새로운 접근은 영어-한국어의 짝지어진 데이터 없이도 성공적으로 학습할 수 있으며, 평가 시간에 이전에 본 적 없던 데이터에서도 좋은 성능을 보여준다.

## 2. 관련 연구

### 2.1 가사 번역

영어-한글 기계 번역에 대한 연구는 매우 많이 행해져 왔으며, 이에 맞게 미세 조정(fine-tuned)된 사전학습(pretrained) 모델들도 많이 존재한다. 사전학습된 모델들은 다양한 분야에서 사용되었고 성능이 검증되었으며, 새로운 번역 모델을 학습시키는 경우와 달리 컴퓨팅 자원과 학습데이터를 사용하지 않아도 되므로 본 논문에서 제안한 방법 중 일부에서 사전학습된 모델이 사용되었다. 우리는 많은 양의 영한 쌍 데이터로 학습된 KoBART 영한 번역 모델[3]을 선택하였다.

### 2.2 음절 제약

본 과제에서 가장 어려웠던 부분은 모델이 번역을 수행할 때 "음절 수"라는 제약을 가지도록 만드는 것이었다. 대부분의 자연어 생성 문제에서는 출력되는 텍스트의 길이에 크게 제약은

두지 않지만 선행 연구 중 그렇지 않은 연구도 존재한다.

음절 제약과 관련된 연구는 애니메이션 더빙 분야에서 연구된 바 있다. 이전의 연구는 주로 번역 과정에서 출력 길이에 가벼운 제약을 가하는 방식으로 진행되었으며 N-best rescoring[1] 방식과 같이 번역 모델의 다변성을 이용하였다. 이런 방식은 음성학적 동기화와 더빙 최적화 번역의 번역 신뢰성에 대한 가중 평균 점수 휴리스틱(heuristic)을 사용한 Saboo와 Baumann의 연구[2]에서 언급된 바 있다. 번역 신뢰도와 출력 길이를 고려한 이런 접근은 음절에 그다지 엄격한 제약을 가하지 않으며 의미 정확도를 보존하기 위해 길이 정확도를 희생하는 결과를 낳았다.

우리의 접근은 후술할 이유들 때문에 앞서 언급한 방법들과는 궤를 달리 한다. 첫째, 노래 번역 과제는 애니메이션 더빙 문제와 많은 유사점을 가지긴 하지만, 필요한 입력 데이터와 출력 음절의 융통성에서 큰 차이가 있다. 둘째, 노래 번역에서는 음성학적 특성이나 성조는 크게 중요하지 않기에 크기가 크고 훨씬 많은 계산 자원이 필요한 음성 데이터나 시각 데이터를 사용하지 않았다. 게다가, 애니메이션 더빙의 특성상, 전술한 모델들은 노래 가사에 비해 덜 엄격한 음절 제약을 가진다.

이런 차이점 때문에 영어 가사를 먼저 한국어로 번역 한 후, 번역된 한국어를 목표 길이 토큰을 이용해 최종 출력물을 생성하는 방식을 선택했다. 이는 스타일 트랜스포머[4]에서 입력 글귀에 스타일 토큰을 함께 입력하여 해당 스타일로 수정된 출력을 생성하는 방식과 유사하며, 두 언어에서의 데이터 쌍(pair)이 존재하지 않을 때 쓰는 방법론인 역번역(back translation)[5]과도 비슷하다. 이 방식을 통해 1) 엄격한 음절 제약을 주는 모델을 만들어 번역 가사의 가창성을 높이고, 2) 쌍으로 이루어지지 않은 데이터로도 모델을 학습시켰으며, 3) 사전학습된 모델을 사용해 계산 부담과 컴퓨팅 자원의 사용을 줄일 수 있었다.

### 2.3 노래 번역

가창성을 고려한 노래 가사 번역과 관련된 선행 연구도 존재한다. 중국어 노래 자동 번역과 관련된 연구[6]는 평가 방식과 모델 구조에 대한 기초적인 방안을 제시했다. 번역 모델의 평가 기준으로 의미, 가창성, 이해도를 제안하였다. 본 논문에서도 비슷한 기준을 채택하여 리리코를 의미와 가창성 두가지 방식으로 평가하였다. 가창성은 음악에 맞추어 가사를 따라 부를 수 있는지를 평가하는 지표이다. Guo[6]가 제시한 모델의 출력은 중국어로, 성조가 있는 언어이기 때문에 더 구체적인 입력을 필요로 한다. 본 논문에서 제시한 모델은 멜로디 입력을 제외하고 오직 음절과 가사 정보만을 입력으로 사용하여 불필요한 정보를 제거하였다. 한국어는 중국어와 달리 성조가 없어 멜로디가 필요하지 않기 때문이다.

## 3. 접근

리리코는 두 단계의 모델로 나뉜다. 1) 제약을 따로 두지 않은 번역 단계와 2) 번역된 글의 음절을 조정해 음절을 맞춘 가사를 출력하는 단계이다. 간단한 모델 구조는 그림 1에서 확인할 수 있다.

그림 1의 첫 단계(1)는 사전학습된 KoBART 번역(KoBART Translator)으로, 영어 가사를 입력받아 음절 제약이 없는 한국어 번역본을 출력한다. 두 번째 단계(2)는 음절 조정 모델(Syllabic Adjustment Model)이다. 이 모델은 한국어 구절과 목표 음절 길이 토큰을 함께 입력받아, 입력과 동일한 의미를 가지면서 목표 음절 수와도 맞는 구절을 출력한다. 학습 과정에서 두 모델은 따로 학습되며, 테스트 과정에서 둘은 하나로 합쳐져 음절을 맞춘 영어-한국어 가사 번역 모델로 활용된다.

### 3.1 무제약 기계 번역

모델의 첫 단계는 사전학습된 기계 번역 모델을 이용하여 영어 가사 본래의 의미를 유지한 한국어 구절을 생성하는 것이다. 좋은 성능을 가진 가벼운 모델이 필요했으므로, 사전학습된 KoBART 번역 모델[3]을 채택하였다. 우리는 모델 학습 시에 영어-한국어 데이터 쌍을 사용하지 않았으므로, 해당 모델을 추가적으로 미세 조정 학습시키지는 않았다.

### 3.2 음절 조정

모델의 두 번째 단계는, 첫 단계에서 나온 토큰화된(tokenized) 출력물을 목표 음절수와 함께 전달하여 동일한 의미를 가지면서 음절 수가 조정된 최종 가사를 출력하는 것이다. 이를 위해 모델에 학습시킬 데이터를 가공하는 과정이 필요했는데, 데이터 생성과 모델 학습의 구조는 그림 2에서 확인할 수 있다.

그림 2의 상단 모델은 데이터셋 생성을 위한 것으로, 학습 전 데이터 전처리에서만 사용된다. 하단의 구조는 음절 조정 모델을 보여준다.

#### 3.2.1 학습 데이터 생성

한국어 가사 데이터를 영어로 번역한 후, 한국어로 재번역하여 사용하였다. 이때 KoBART Korean-English 모델[7]과 English-Korean 번역 모델[3]을 사용하였다. 목표 음절 정보를 나타내기 위해서 [START] 토큰 뒤에 [LEN<number of syllables>]을 덧붙여 다음과 같은 입력 데이터를 만들었다.

[START][LEN<number of syllables>]한국어 가사[END]

데이터 생성 과정은 4.1에서 확인할 수 있다.

#### 3.2.2 음절 조정 모델 구조

음절 조정 모델은 길이에 제약을 두며 번역까지 동시에 해야 하는 모델보다는 훨씬 간단하지만, 여전히 길이 토큰과 목표

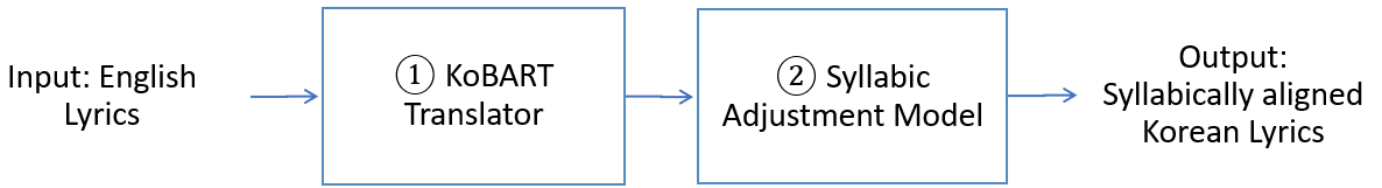


그림 1. 모델 구조

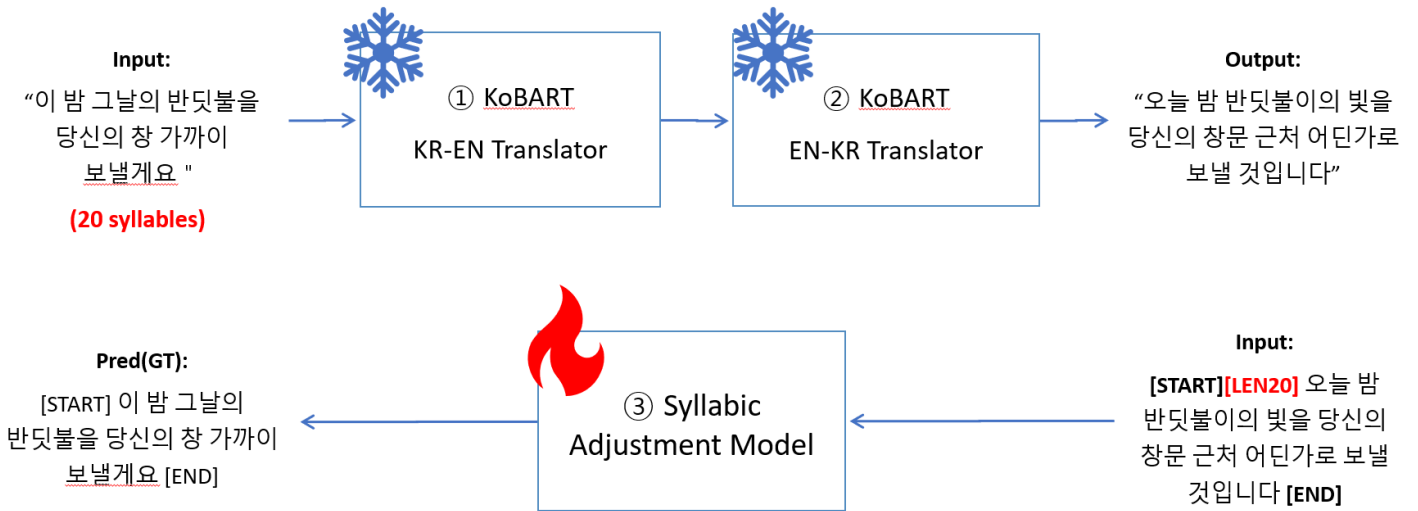


그림 2. 음절 조정 모델의 구조. ❄️: 가중치 동결, 🔥: 가중치 미세 조정.

출력물간의 관계를 학습하기 위해 많은 사전 학습이 필요하다. 따라서 바닐라 트랜스포머(Transformer)[8]를 사용하는 대신, 40GB이상의 한국어 데이터로 사전학습된 BART[9] 기반의 KoBART[10] 모델을 추가로 학습하는 방식을 선택하였다.

리리코는 인코더-디코더(Encoder-Decoder) 구조를 갖고 있으며 각각의 인코더와 디코더는 총 6계층의 트랜스포머로 이루어져 있다. 각 트랜스포머는 총 16개의 어텐션 헤드와, 3072 차원의 feed-forward 층과 768 차원의 은닉층으로 이루어져 있다. 학습에는 BART[9]에서 사용된 수식 1의 크로스 엔트로피 손실 함수를 사용하였다.

$$\mathcal{L}_{ours} = - \sum \sum p(x) \log(q(x)) \quad (1)$$

$p(x)$ 는 실제 확률 분포이고,  $q(x)$ 는 각 토큰들에 대한 예측 확률 분포이다. 구체적인 학습 방법은 4.1.2에서 확인할 수 있다.

## 4. 실험

### 4.1 실험 방법

#### 4.1.1 학습 데이터

영어 가사를 한국어로 번역하기 위해서 가장 중요한 것은 음절 수가 일치하는 영어-한국어 가사 데이터 쌍이었다. 번역된 가사가 원래 노래에 맞추어 불릴 수 있기 위해서는 음절 수가 일치하는 것이 매우 중요하다. 그러나 음절수가 일치하는 많은 양의 데이터 쌍이 존재하지 않았고, 따라서 전술했던 것처럼 우리는 새로운 접근 방식으로 모델을 두 단계로 나누고 직접 학습 데이터를 만들기로 하였다.

한국어 가사를 얻기 위해 한국 음원 스트리밍 사이트 멜론(Melon)으로 부터 가사 데이터를 모았다. 2000년 1월부터 2023년 4월까지의 Top-100 차트의 한국어 가사와 인디 음악 가사를 사용하였다. 중복을 제외하고, 약 8000개의 노래에서 약 170,000 줄의 가사 데이터를 얻을 수 있었다. 한국어가 포함되지 않은 일본어나 중국어의 노래는 제외했고, 영어가 포함된 한국어 노래는 포함하였다.

앞서 서술했듯이 무제약 번역 모델은 따로 추가적으로 학

습시키지는 않았다. 대신 음절 조정 모델을 학습시키기 위해, 수집한 가사 데이터에 대해 전처리를 수행하였다. 원본 한국어 가사는 학습 과정에서 정답(ground truth)으로 사용되었고, 이 가사를 사전학습된 번역 모델을 이용해 영어로 번역한 후, 다시 한국어로 번역하는 과정을 거쳐 입력 데이터를 만들었다. 추가적으로, 정답 데이터의 음절 수를 목표 음절 수로 두어 입력 데이터 앞부분에 [LEN<number of syllables>] 토큰으로 두었다. 이러한 과정을 통해 우리는 한국어로 (재)번역된 데이터-원본 한국어 데이터 쌍을 얻을 수 있었다. 이 데이터를 이용하여 학습된 음절 조정 모델은 입력된 한국어 가사를 의미를 유지하며 목표 음절에 맞게 조정해준다.

#### 4.1.2 학습 방법

리리코의 앞 부분(무제약 번역 모델)은 가중치를 동결한 사전 학습된 번역 모델이다. 뒷 부분(음절 조정 모델)의 학습 방법은 아래와 같다.

음절 조정 모델은 배치사이즈 512로 30 에폭 학습되었다. 최적화 알고리즘으로는 학습률  $3e-5$ 의 AdamW[11]를 사용하였다. 코사인 스케줄러(coine scheduler)과 gradient clipping을 사용하여 파라미터가 쉽게 국소 최적해(local optima)에 수렴하는 것을 방지하였다.

리리코는 구글 코랩(Colab) 환경에서 학습되었으며 GPU로는 NVIDIA A100이 사용되었다. 170,000 줄의 데이터셋에서 한 에폭의 학습은 약 4분이 소요되었으며 30 에폭 학습에는 총 2시간이 소요되었는데, 해당 테스트와 결과를 고려했을 때 효율적이다.

## 4.2 평가 방법

모델의 성능을 평가 하기 위해서 양적 평가와 질적 평가 두 가지 방식을 모두 이용했다. 양적 평가는 가사의 번역과 음절 정렬에 대한 견고한 평가를 할 수 있게 해준다. 다만 음악은 주관적이고 예술적인 특성을 갖고 있으므로, 사람이 직접 평가하는 질적인 평가 또한 가창성과 번역의 적절성을 판단하는 데 필수적인 요소이다.

### 4.2.1 양적 평가

우리의 모델을 평가하기 위해 음절 정확도와 의미 정확도 두 가지 관점에서 양적 평가를 실행했다. 두 가지를 동시에 평가하려면 짝지어진 데이터셋이 있어야 하는데 이러한 데이터 쌍이 없으므로, 우리는 두 관점에 대한 평가 방식을 다르게 하여 개별적으로 평가를 진행하였다.

음절 정확도를 측정하기 위해 원문 영어 가사의 음절 수와 모델 출력 가사의 음절 수를 비교하였다. 음절 정확도는 수식 2와 같이 목표 음절수에 대한 음절 수의 절대적 값으로 계산하였다. 이 지표는 주어진 가사를 멜로디에 맞게 모델이 얼마나

잘 변환시키는지 나타낸다.

$$\frac{|syll(English) - syll(translated Korean)|}{syll(English)} \quad (2)$$

번역 정확도를 평가하기 위해 일반적으로 사용되는 방식은 BLEU 점수[2][1]이다. 그러나 BLEU를 평가하기 위해서는 모델의 출력과 비교할 한국어 정답이 필요했기에 짝지어진 데이터가 없는 우리 모델에서는 사용할 수 없었다. 따라서 각 구절의 토큰의 코사인 유사도를 이용해 의미 유사도를 평가하는 BERT 점수[12]가 가장 적합했다. 테스트 시간에는 약 600개의 영어 노래의 가사를 추출하여 입력과 출력의 BERT 점수를 계산했다.

### 4.2.2 질적 평가

양적 평가만으로는 주관적인 측면을 완전히 평가할 수 없기 때문에 추가로 질적 평가를 진행하였다. 이 질적 평가는 번역 정확도와 일반적인 가창도를 더 전체적으로 평가하는 것을 목적으로 하며 인간의 선호도와 해석을 평가한다. 질적 평가는 번역 정확도와 가창도의 두 부분으로 진행되었다. 무작위로 선정된 20명이 평가에 참가하였으며, 각각 무작위로 선정된 팝송 가사와 모델이 번역한 한국 가사가 주어졌다.

## 4.3 결과

평가 시간에서 결과를 평가하기 위해, 모델이 훈련에 사용되지 않은 영어 가사를 모델에 입력해 음절이 조정된 한국어 가사를 생성했다. 그림 3은 테스트 시간에 번역된 영어 가사의 예시이다.

### 4.3.1 양적 평가

그림 5에서 평가 결과를 확인할 수 있다. BERT 스코어에서 75% 이상이 0.65점 이상을 얻었다. "Look at this stuff"와 "이것 좀 봐"의 BERT 스코어가 0.7인 것을 감안할 때, 0.65점은 꽤나 정확한 결과이며 번역 모델이 가사의 의미를 잘 유지하며 번역했다고 해석할 수 있다. 음절 평가에서는 출력값 대부분의 음절수가 정확히 동일했다. 이는 음절 조정 모델이 매우 정확한 결과를 생성할 수 있다는 것을 보여준다. 음절 수의 차이가 10 이상인 이상치들이 몇 있었는데, 이는 무제약 번역 단계에서의 오류이거나 목표 길이의 조정이 실패해 빈 출력을 반환한 경우였다.

### 4.3.2 질적 평가

위의 양적 평가는 리리코가 의미와 음절적으로 얼마나 잘 번역하는지 나타내주지만, 의미나 문맥적인 부분에서 질적 평가가 더 좋은 평가 지표가 될 것이다. 20명의 평가자 중에 15명이 번역 내용이 만족할 만하다고 답변했으며 3명은 잘 모르겠음, 2명은 별로라고 답변하였다. 모델의 번역 정확도는 평균 9/10,

LyriKOR

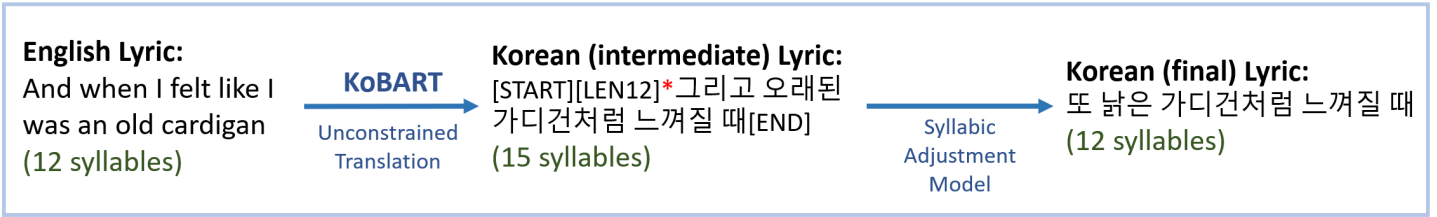


그림 3. 테스트 데이터셋 결과 예시. \*[LEN12] 토큰은 12음절을 의미함.



그림 4. 악보와 맞는 가사 예시

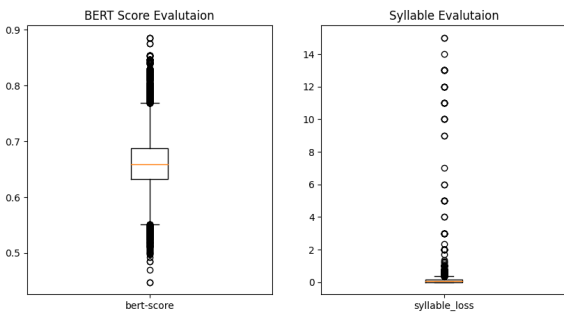
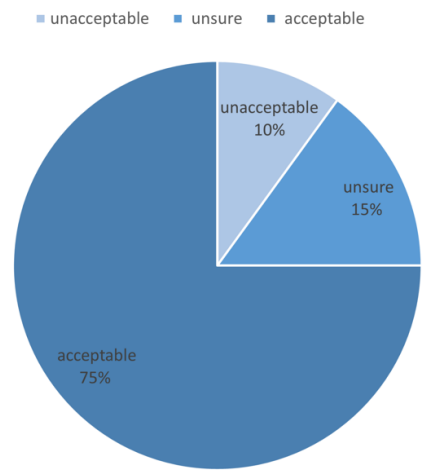


그림 5. (좌) BERTScore 결과, (우) 음절 평가 결과.

가창성은 평균 8.7/10점이었다. 두 지표 모두 의미 정확도와 가창성에서 좋은 결과를 얻었다.

참가자들에게 추가적인 질적 평가를 요청했을 때 대부분 음절 제약 덕분에 멜로디에 따라 노래를 부르기 쉬워지지만, 멜로디의 높낮이에 가사가 어색한 부분이 있다는 의견이 있었다. 특정 음이 길게 지속되는 경우 리리코는 추가적인 음성 정보가 없 이 음의 길이를 고려할 수 없기 때문에 해당 가사의 가창성이 저해되는 경우도 있었다. 추가적으로, 음절 수를 맞추기 위해 한국어 단어가 연속적으로 반복되는 경우도 있었다. 또한 번역된 의미는 전반적으로 매우 정확하지만 몇몇 경우에 어색하거나 문법적으로 부정확한 경우가 있었다. 이는 예술적이고 더 자유로운 문법을 가진 가사 번역에서 흔하게 보이는 한계점으로 보인다.

Overall Translation Quality Rating



Subtask Scores

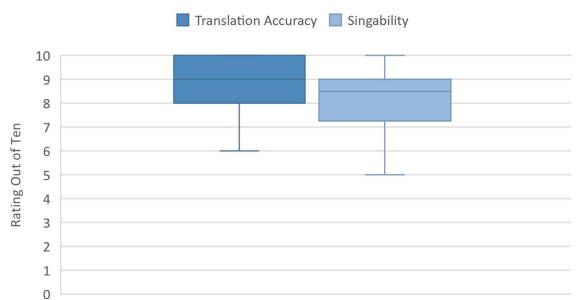


그림 6. (좌) 20명의 참가자가 전반적인 가사의 품질을 acceptable, unsure, unacceptable로 평가한 결과. (우) Translation Accuracy와 Singability에 대한 평가 분포.

5. 분석

5.1 이전에 사용된 방법들과의 차이

이전 연구들은 음절 정확도와 의미 정확도 사이에 상충 관계에 초점을 맞추어, 이 관계를 하이퍼 파라미터의 형태로 직접적으로 손실 함수에 사용하여 둘의 우선순위를 조절하였다. 이러한 음절과 의미 정확도의 충돌은 전문 번역가도 피할 수 없는 문제이지만, 우리는 서로 다른 두 모델이 두 역할을 연속적으로 수행하게 하여 음절과 의미 정확도의 충돌을 상대적

으로 감소시켰다. 또한 의미 번역과 음절 조정에 각각 모델을 사용하였기에, 모델이 수행하는 작업은 번역과 음절을 조정을 동시에 수행하는 방식보다 더 단순화되었다. 리리코는 하나의 복잡한 작업을 두개의 단순 작업으로 나누어 해결함으로써 성공적으로 번역을 수행하였고, 음절과 의미 정확도 모두에서 더 좋은 결과를 얻을 수 있었다.

## 5.2 한계점

그러나 위 접근 방식에는 몇 가지 한계가 있다. 첫째, 번역과 길이 조정을 위해 두 개의 개별 모델에 의존하기 때문에 첫 번째 모델의 정확성이 두 번째 모델 결과에도 큰 영향을 준다. 결과적으로 부분적인 의미를 가진 가사의 경우, 번역 모델이 전체 맥락과 다른 의미로 번역을 하게 되면 최종 출력되는 가사도 부정확해진다. 본 논문에서 사용된 번역 모델의 성능이 좋았기 때문에 테스트에서 큰 성능 저하가 있지는 않았지만, 이후에 잠재적으로 문제가 될 수 있다.

또한, 모델 학습에 이용된 훈련 데이터와 테스트 데이터의 분포 차이가 있다. 음절 조정 모델의 입력으로는 한국어-영어-한국어의 번역을 거친 의미가 같고 음절수는 다른 한국어 번역 데이터, 정답으로는 원본 한국어 가사가 사용되었다. 실제 시험 데이터에 가능한 한 가까운 분포를 갖는 데이터를 사용하고 싶었지만, 훈련 중에 사용되는 기계 번역 영어 가사의 데이터 분포와 특성은 인간이 번역한 영어 가사의 데이터 분포와 다를 수밖에 없다. 이 분포 차이의 정도에 따라 모델에 부정적인 영향이 있을 수 있다.

## 6. 결론

본 논문에서, 우리는 영어 노래를 한국어로 번역하는 새로운 방법을 제안하였다. 과거의 다른 방법들은 번역과 장황성 (verbosity) 제어를 결합했지만, 우리는 먼저 번역한 후, 제약 조건에 맞게 음절을 조정하는 분할 및 정복 방법을 사용하였다. 정확한 음절 수를 가진 영어-한국어 데이터셋이 없었기에 이 문제를 해결하기 위해 새로운 방식으로 데이터셋을 직접 만들어서 사용하였다. 한국어 가사를 번역해 의미가 동일한 영어 가사를 만들어 이 쌍을 데이터셋으로 만들었다. 음절 제약이 이미 있는 한국어 가사의 의미를 보전한 영어 가사를 만들었지만, 몇가지 한계점도 존재한다. 데이터셋을 만들 때 이용된 사전 훈련된 번역 모델의 성능이 본 모델의 성능에 영향을 주고, 기계번역으로 만든 훈련 데이터셋과 사람이 번역한 실제 테스트 데이터셋의 분포가 잠재적으로 차이가 있을 것이라는 점이다. 향후 연구에서는 훈련 데이터와 테스트 데이터의 분포를 최소화 하기 위해 사람이 번역한 영어-한국어 데이터셋을 훈련에서도 사용하고, 더 많은 양의 데이터로 모델을 훈련시키고 싶다. 또한, 현재 모델은 노래 가사를 한 줄씩 번역하기

때문에 노래 전체의 맥락을 이해하지 못해 일부 가사의 번역에 어려움이 있다. 따라서 전체 노래에 대한 의미 맥락을 활용할 수 있도록 모델이 전체 가사를 한번에 처리할 수 있도록 크기를 키우면 성능이 더 향상될 것이라고 생각한다.

## 참고문헌

- [1] S. M. Lakew, M. Federico, Y. Wang, C. Hoang, Y. Virkar, R. Barra-Chicote, and R. Enyedi, "Machine translation verbosity control for automatic dubbing," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- [2] A. Saboo and T. Baumann, "Integration of dubbing constraints into machine translation," *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pp. 94–101, Aug. 2019. [Online]. Available: <https://aclanthology.org/W19-5210>
- [3] Circulus, "circulus/kobart-trans-en-ko-v2," <https://huggingface.co/circulus/kobart-trans-en-ko-v2/tree/main>, 2023.
- [4] N. Dai, J. Liang, X. Qiu, and X. Huang, "Style transformer: Unpaired text style transfer without disentangled latent representation," 2019.
- [5] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," 2016.
- [6] F. Guo, C. Zhang, Z. Zhang, Q. He, K. Zhang, J. Xie, and J. Boyd-Graber, "Automatic song translation for tonal languages," *Findings of the Association for Computational Linguistics: ACL 2022*, 2022.
- [7] Circulus, "circulus/kobart-trans-ko-en-v2," <https://huggingface.co/circulus/kobart-trans-ko-en-v2/tree/main>, 2023.
- [8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.
- [9] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," 2019.
- [10] skt ai, "skt-ai/kobart," <https://github.com/SKT-AI/KoBART>, 2020.

- [11] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” 2019.
- [12] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “Bertscore: Evaluating text generation with bert,” 2020.