Journal of The Korea Convergence Society
Vol. 13. No. 3, pp. 43-54, 2022
e-ISSN 2713-6353
DOI : https://doi.org/10.15207/JKCS.2022.13.03.043

실시간 동시통번역의 정책기반 성능 비교 연구

이정섭[1], 문현석[1], 박찬준[1], 서재형[1], 어수경[1], 이승준[1], 구선민[1], 임희석[2*]
[1]고려대학교 컴퓨터학과 석·박사통합과정, [2]고려대학교 컴퓨터학과 교수

Policy-based performance comparison study of Real-time Simultaneous Translation

Jungseob Lee[1], Hyeonseok Moon[1], Chanjun Park[1], Jaehyung Seo[1], Sugyeong Eo[1], Seungjun Lee[1], Seonmin Koo[1], Heuiseok Lim[2*]
[1]Master & Ph.D. Combined Student, Department of Computer Science and Engineering, Korea University
[2]Professor, Department of Computer Science and Engineering, Korea University

요 약 동시통번역은 문장의 일부만으로 번역을 시작하는 온라인 디코딩으로 지연 대비 번역 성능을 평가 지표로 사용한다. 동시통번역 연구의 공통의 목적은 지연 대비 번역 성능을 높이는 것으로, 지연과 번역 성능 사이의 적절한 절충점을 찾는 것이다. 본 논문은 이러한 동시통번역의 현재 연구 흐름을 반영하여 한국어에서 고정 정책 기반 동시통번역의 비교 실험을 진행하였다. 또한, 한국어에서 동시통번역은 토큰화 과정에서 많은 분절이 발생하여 다른 언어 대비 불필요한 지연이 발생하게 되고, 이를 해결하기 위한 n-gram 토큰화 방안 등의 후속 연구의 필요성에 대해 제시하였다.

주제어 : 동시통번역, 기계번역, 온라인 디코딩, 음성번역, 온라인 정책, 언어융합

Abstract Simultaneous translation is online decoding to translates with only subsentence. The goal of simultaneous translation research is to improve translation performance against delay. For this reason, most studies find trade-off performance between delays. We studied the experiments of the fixed policy-based simultaneous translation in Korean. Our experiments suggest that Korean tokenization causes many fragments, resulting in delay compared to other languages. We suggest follow-up studies such as n-gram tokenization to solve the problems.

Key Words : Simultaneous translation, Machine translation, Speech translation, Online policy, Language Convergence

*This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques) and supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2021R1A6A1A03045425).

*Corresponding Author : Heuiseok Lim(limhseok@korea.ac.kr)

Received January 10, 2022
Revised February 17, 2022
Accepted March 20, 2022
Published March 28, 2022

1. 서론

동시통번역(Simultaneous Machine Translation, SiMT)은 문장 중 일부의 단어를 짧은 지연 간격마다 다른 언어의 단어로 번역하는 것을 목표로 한다. 이는 번역을 실시간으로 수행해야 하는 상황에서 필요성이 존재한다. 예를 들어, 동시통역은 국제회의, 정상회담 혹은 실시간 라이브 스트리밍에서 널리 사용된다. 동시통역은 두 언어를 동시에 이해하고 즉각적으로 번역문을 전달하기 때문에 매우 어렵고 많은 에너지를 요구한다. 실제로, 동시통역사는 두 언어를 모국어 수준으로 이해하고 있어야 하며, 대게 30분의 가용 시간을 가지고 있다. 따라서, 언어 간의 접근성과 경제성을 부각하기 위하여 동시통번역을 연구할 필요가 있다.

동시통번역은 기존의 기계번역(Machine Translation, MT)에서 번역의 성능뿐 아니라 지연(Lagging)을 줄여야 하는 목적이 더해져, 이를 위해 때로는 번역 시스템이 화자가 말하는 것을 예측해야 하므로 상당히 까다로운 것으로 알려져 있다. 더하여, 한국어의 토큰화 (Tokenization) 방식은 다른 언어의 토큰화에 비해 많은 분절을 요구하므로 상대적으로 지연이 높아진다. 이러한 문제로 인해 한국어에 대한 동시 번역 연구의 중요성이 부각된다. 더불어, 한국어와 관련된 동시통번역에 대한 연구는 거의 진행되지 않고 있어, 다양한 지연 대비 번역 성능을 최대화하는 한국어 동시 번역 연구의 필요성에서 본 연구의 동기를 얻을 수 있다.

트랜스포머(Transformer)[1] 이래로 기계번역은 많은 발전을 이루었다. 트랜스포머 구조의 번역 모델에 소스(Source) 문장을 인코더에 입력하면, 자기 회귀(Auto-regressive) 방식으로 디코더에서는 입력 문장의 문맥을 반영하여 번역(Target) 문장을 생성한다. 하지만, 이는 동시통번역에서 적용하기 어려운 부분이 있다. 동시통번역은 문장 전체를 인코더에 입력하지 않고, 문장의 일부를 통해 번역을 진행한다. 이는, 입력 언어와 번역된 언어의 어순 차이로 큰 어려움을 야기한다. 이러한 이유로, 최근 동시통번역 연구는 입력 단어를 더 기다릴 것인지 (READ), 혹은 타겟 단어를 생성할 것인지 (WRITE)의 행동 (Action) 여부를 결정하는 정책 (Policy)을 정의하는 것에 집중되어 있다. 이러한 동시통번역의 정책을 연구하는 최근의 논문에서 정책은 크게 (1) 고정 지연정책 (Fixed latency policy)[2,3], (2) 동적 지연정책 (Adaptive latency policy)[4-6]의 두 가지로 분류된다. 특히, 동적 지연정책은 기존의 고정 지연정책에 비해 더 나은 성능을 보였으며, 다양한 동적 지연정책 학습 방법 연구에 많은 연구를 촉진했다. 대표적으로, 인코더 기반의 언어모델을 이용하여 단어 단위 정책을 생성하는 연구[7-8], 인코더 어텐션의 구조에 단조 어텐션 메커니즘 (Monotonic attention mechanism)을 적용해 동적 정책을 학습하는 연구[9-11], 강화학습(Reinforcement Learning)이나 모방 학습(Imitation Learning)을 통해 정책을 학습하는 연구[12-14] 등의 다양한 연구가 진행되었다.

동시통번역 모델은 어순이 다른 언어 간의 번역에서 과감한 예측을 해야 좋은 성능을 얻을 수 있다. 동시 번역에서 SOV → SVO 번역은 지연을 낮추기 위해 동사를 예측하는 것이 필수적이다. 이는 동사를 예측해야 하는 것 때문에 동시통번역에서 가장 어려운 언어 간의 번역으로 여겨진다. Fig. 1은 SOV형 언어인 한국어에서 SVO형 언어인 영어로의 동시통번역 예시이다. 동시통번역 모델은 "우리는 이전 립스틱 제품들보다 더"를 통해 동사인 "tried"를 예측해야 한다. 이는 즉, 예측 필요 여부가 기존 기계번역과 동시 번역에서의 큰 차이로 해석할 수 있으며, 언어 간의 어순 차이를 극복할 모델링 기법에 따라 성능의 차이를 나타낼 수 있음을 알 수 있음을 암시한다.

이러한 이유로, 본 논문은 동시통번역에 대한 연구 동향과 평가지표에 대해 설명하고, 각 연구에서의 문제점에 대해 제시하고, 실험을 통해 동시통번역의 문제와 한국어에서 영어로의 동시통번역에서 추가로 발생하는 문제점을 실험을 통해 제시한다. 또한, 한국어의 언어학적 특성을 토대로 문제점의 원인을 제안하고, 이에 대한 해결방안을 제안한다.

본 논문의 구성은 다음과 같다. 2절에서, 동시통번역의 시작점인 기계번역의 학습 방법에 대해 간단히 설명하고, 궁극적인 목표인 동시 음성번역과 동시통번역 연구의 흐름에 관해 서술한다. 3절에서, 동시 번역의 평가지표인 번역 품질 평가지표와 지연 평가지표에 대해 서술한다. 4절에서, 고정 정책 기반 동시통번역 연구에 관련된 동향을 서술하고, 동시 번역 정책 학습 과정에 대해 자세히 서술한다. 5절에서, 동적 정책 기반 동시통번역의 연구 동향에 관해 서술한다. 마지막으로, 한국어에서 고정 정책 기반 동시통번역 실험 결과를 제시하고, 문제의 원인에 대해 분석한다. 또한, 이를 바탕으로 한국어 동시 번역 연구의 필요성을 강조하고, 개선을 위한 해결방안 대해 서술한다.

Fig 1. Example of the wait-6 simultaneous system on Korean → English translation

2. 배경지식

2.1 기계번역

최근 기계번역은 연구는 신경망을 기반으로 하는 신경망 기계번역 (Neural Machine Translation, NMT)을 중심으로 연구되고 있다. 번역문을 생성하는 합리적인 모델은 인코더-디코더 구조의 모델로, 인코더에서 입력받은 소스 문장을 바탕으로 컨텍스트 벡터 (Context vector)를 디코더에 전달하고 디코더는 해당 벡터를 반영하여 타겟 문장을 생성하게 된다.

기존 신경망 기계번역의 인코더는 문장 $\mathrm{x}=(x_1,x_2,...,x_n)$ 전체를 입력으로 받아, 인코더의 은닉 상태(hidden state) $\mathbf{h}$를 출력하며, 디코더는 인코더의 은닉 상태 $\mathbf{h}$를 디코더의 입력으로 확률 $P(y_i;\mathbf{x},\mathbf{y}_{<i};\theta)$를 최대화하는 타겟 단어 y_i를 선택하여 생성한다. 이러한 과정은 자기 회귀 방식으로 〈eos〉 토큰을 출력할 때까지 반복하게 되며, 최종 번역 생성 문장 $\mathbf{y}=(y_1,y_2,...,<\mathrm{eos}>)$ 를 얻는다. 즉, 최종 번역 문장 $\mathbf{y}$의 확률은 식 (1)과 같다.

$$P(\mathrm{y}|\mathrm{x};\theta)=\Pi_{i=1}^{|\mathrm{y}|}\ P(y_i|\mathrm{x},\mathrm{y}_{<i};\theta) \qquad (1)$$

따라서, 병렬 문장 $(\mathbf{x},\mathbf{y}^*)$에 대해 Cross-entropy를 적용한 음의 조건부 확률(Negative conditional probability)의 손실 함수는 식 (2)와 같다.

$$\mathcal{L}(D)=-\sum_{(\mathbf{x},\mathbf{y}^*)\in D}\log P(\mathbf{y}^*|\mathbf{x};\theta) \qquad (2)$$

신경망 기계번역은 식 (2)를 최소화하는 방향으로 학습이 이루어진다.

2.2 음성번역

음성번역 모델은 cascaded 구조와 end-to-end 구조로 구분할 수 있다. cascaded 구조는 음성번역 모델에 음성을 텍스트로 변환하는 모듈인 자동 음성 인식 (Automatic Speech Recognition, ASR)을 부착하여 변환된 텍스트를 번역하는 모델이다. cascaded 구조는 텍스트 간의 번역을 학습하는 것으로, 음성번역 연구의 관점에 포함되지 않는 경우가 있다. 해당 구조는 자동 음성 인식 모듈의 성능에 따라 모델의 성능에 큰 영향을 준다. 또한, 모듈에 의해 변환된 텍스트를 기반으로 번역을 학습하기 때문에, 최종 번역이 텍스트인지 음성인지에 따라 text-to-speech, text-to-text로 구분할 수 있다. 반면, end-to-end 구조는 자동 음성 인식 모듈 없이 음성을 단방향으로 번역하는 speech-to-speech 혹은 speech-to-text 모델이다. 해당 구조는 노이즈나 어투에 민감하여 비교적 번역 성능이 낮지만, 음성 인식 모듈의 에러를 전파받지 않는다. 또한, cascaded 모듈이 두 번의 추론을 진행하는 반면, end-to-end 모듈은 단방향으로 학습을 진행하기 때문에 한 번의 추론을 진행하여 빠른 서비스 속도를 제공할 수 있다는 장점이 있다[15]. 이러한 이유로, 음성번역 최근 연구는 대부분 end-to-end의 형식으로 이루어진다[16-19]. Fig. 2는 음성번역의 두 가지 구조를 그림으로 나타낸 것이다. 왼쪽 그림은 자동 음성 인식이 부착된 cascaded 구조

이고, 오른쪽은 음성을 단방향으로 처리하는 end-to-end의 구조이다.

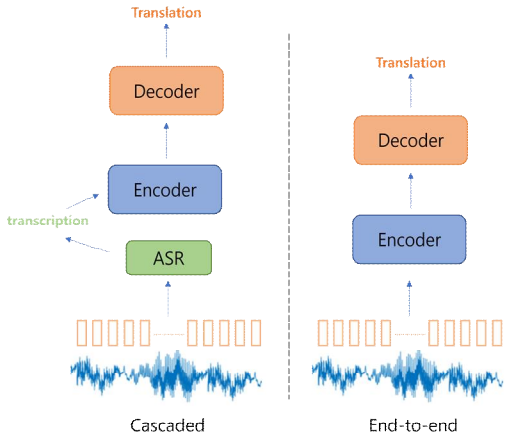

Fig. 2. Speech translation architectures

2.3 동시통번역

기계번역 모델은 식 (1)과 같이 인코더에서 문장 전체를 읽은 후에, 디코더가 타겟 토큰 생성을 시작한다. 이와는 다르게, 동시통번역 모델은 인코더가 문장 전체를 읽기 전에 디코더가 타겟 토큰을 생성하기 시작한다. 인코더에서 소스 단어를 얼마큼 읽고 타겟 토큰 단어를 생성할지 결정하는 것이 동시통번역에서 말하는 정책을 정의하는 것이다.

동시통번역 모델은 정책에 따라 번역 성능과 지연시간의 절충점(Trade-off)이 존재한다. 많은 소스 단어를 읽은 후에 디코딩을 시작하면 번역 성능이 높아지게 되지만, 동시통번역에서 중요히 여기는 지연시간에 악영향을 주게 된다. 반대로, 충분한 소스 단어를 읽지 않고 디코딩을 시작하면 지연시간 성능은 좋아지지만, 번역 과정에 예측이 포함되어 충분한 번역 성능을 낼 수 없다. 이러한 이유로, 동시통번역 시스템은 번역 성능과 지연시간의 절충점을 맞추기 위하여 정책을 신중히 정의해야 한다.

동시통번역의 궁극적인 목표는 언어적 장벽을 완화하는 동시 음성번역 모델을 구축하는 것이다[20]. 기존 동시통번역의 대부분의 연구는 자동 음성 인식 시스템을 사용하여, 음성을 텍스트로 변환했다는 가정에서 이루어졌다. [2,3] 등의 연구는 일정한 규칙의 스텝마다 읽기와 번역을 번갈아 하는 고정 지연정책을 제안하고, [9-11] 등의 연구는 단조 어텐션을 활용한 학습 가능한 동적 지연정책을 제시한다. 또한, [12-14] 등의 연구는 디코딩을 조절하는 강화학습(Reinforcement Learning) 혹은 모방학습(Imitation Learning) 전략을 반영한 동적 지연정책을 제시한다. 동시통번역 정책에 관한 최근의 연구는 효율적이고 지연을 유연하게 조정할 수 있는 동적 지연정책을 학습하는 것에 초점이 있다.

동시통번역에서 기존 연구는 음성이 텍스트로 변환됐다는 가정하에 이루어졌다. 하지만 cascaded 음성번역 시스템과 동일하게, 자동 음성 인식을 부착한 동시통번역 시스템은 모델을 학습하는 과정에서 음성이 완벽한 텍스트로 변환되지 않았을 때의 에러가 함께 전파되어 모델의 성능을 낮출 뿐 아니라, 음성이 텍스트로 변환하는 과정에서 불필요한 서비스상 지연을 야기한다는 문제점이 있다. 이러한 이유로, 음성 정보를 인코더의 입력으로 직접 번역하는 많은 연구 사례가 있다[21-24].

따라서, 동시 음성번역은 직관적인 추론 결과를 확인할 수 있는 cascaded 구조에서 text-to-text에 관한 지연 대비 품질의 절충점을 가지는 정책을 학습하는 연구에서, 이러한 정책을 end-to-end 구조의 동시 음성번역 모델에 적용하는 연구로 발전하는 중이다.

3. 동시통번역 평가지표

번역 품질과 지연 성능은 동시통번역 모델을 평가하는 중요한 요소이다. 동시통번역 시스템은 번역 품질과 지연 간의 균형을 유지해야 한다. 동시통번역 시스템의 목표는 번역 품질을 최대화하고, 지연 품질을 최소화하는 것이 목표이다. 해당 절에서는 동시통번역 시스템에서 사용하는 번역 품질 평가지표와 지연 평가지표에 대해 소개한다.

3.1 번역 품질 평가지표

번역의 품질 평가를 위해서 가장 대중적으로 사용하는 방법은 단어 혹은 n-gram의 오버래핑 (Overlapping)을 측정하는 것이다. 동시통번역에서 번역 품질 평가지표는 기존 기계번역에서 사용하는 동일한 평가지표를 모두 사용할 수 있으며 추가적인 제약이 없다.

3.1.1 데이터 정제과정 부족의 문제

기계번역 시스템과 마찬가지로, 동시통번역 시스템

또한 대중적으로 BLEU[25]를 번역 품질 성능지표로 사용한다. BLEU 지표는 기계번역 분야에서 가장 빈번히 사용되는 품질 성능지표로, 각 문장에 대해 n-gram 기반 오버래핑을 측정하고 평균화한다.

3.1.2 METEOR

METEOR[26] 지표는 BLEU와는 다르게 유니그램 정밀도 (Precision)와 재현율 (Recall)의 조화 평균을 기반으로 측정되며, BLEU 지표에서 발생하는 동의어 문제를 WordNet 등의 동의어 사전 매칭을 전략 추가로 고려할 수 있는 품질 평가 지표이다.

3.2 지연 평가지표

지연을 측정하는 평가지표는 주로 Consecutive Wait[13], Average Proportion[2], Average Lagging [3], Differential Average Lagging[27]의 네 가지 평가지표를 사용하여 지연 성능을 측정한다.

본 절에서, 입력으로 사용되는 입력 소스 문장을 $\mathbf{x}$로 표기하고, 번역 생성 문장을 $\mathbf{y}$로 표기한다. 또한, 디코딩 스텝 t에서, y_t를 생성할 때 인코더에 입력된 소스 문장의 수를 delay로 정의하며 $\mathbf{g} = (g_1, \ldots, g_k)$로 표기한다.

3.2.1 Consecutive Wait, CW

[13]의 연구에서 정의된 Consecutive Wait (CW)는 연속된 디코딩 스텝 t, $t-1$에서, 타겟 토큰 생성 사이에 몇 개의 입력 토큰이 대기했는지 수치로 나타낸 지연 평가지표로 식 (3)와 같다.

$$c_t = \begin{cases} c_{t-1} + 1 & \text{action}_t = \text{READ} \\ 0 & \text{action}_t = \text{WRITE} \end{cases} \tag{3}$$

식 (3)은 delay g에 관한 식으로 표기할 수 있으며, 이는 식 (4)와 같다

$$c_{g_t} = g_t - g_{t-1} \tag{4}$$

Consecutive Wait은 연속으로 생성된 두 토큰을 기준으로 지연을 계산하기 때문에, 국소 지연 (local latency)만을 측정한다는 단점이 있다[3].

3.2.2 Average Proportion, AP

Average Proportion (AP)[2]는 각 토큰을 번역할 때 필요한 평균 소스 토큰의 수치로 정의하며, 식 (5)로 표기할 수 있다.

$$AP = \frac{1}{|\mathbf{x}||\mathbf{y}|} \sum_{i=1}^{|\mathbf{y}|} g_i \tag{5}$$

AP는 크게 두 가지 문제점을 가지고 있다. 1) 수치가 직관적이지 않다: 이상적인 지연 정책인 단어 간격 번역 (word-by-word translation)에서 $AP = 0.5$의 값을 얻을 수 있다. 이는, 0.5의 값이 가장 이상적인 AP 값임을 의미하며, AP 값의 범위가 $0 \leq AP \leq 1$이라는 점을 고려했을 때 이는 직관적으로 해석되지 않을 수 있다. 2) 입력 문장의 크기 $|\mathbf{x}|$와 타겟 문장의 크기 $|\mathbf{y}|$에 민감하다: AP 지표에서 $|\mathbf{x}|$와 $|\mathbf{y}|$가 무수히 커지면, AP 값은 가장 이상적인 값 0.5에 가까워진다. 또한, $|\mathbf{y}|$가 작아지면, AP의 값은 1에 가까워진다.

3.2.3 Average Lagging, AL

Average Lagging (AL)은 '지연은 이상적인 정책 (Ideal policy) 이후에 발생한다'라는 아이디어를 동기로 제안되었다[3]. 여기에서 표현하는 이상적인 정책은 입력 문장의 크기 $|\mathbf{x}|$와 타겟 문장의 크기 $|\mathbf{y}|$가 동일할 때, 디코더가 인코더보다 한 스텝이 앞서 있는 단계를 의미하며, 이상적인 정책에서 $AL = 0$을 만족하도록 식을 제안한다. 즉, AL 지표는 입력 문장 AL=0가 모두 입력으로 들어오기 이전의 단계에서의 지연만을 측정한다. AL의 식은 식 (6)와 같다.

$$AL = \frac{1}{\tau} \sum_{i=1}^{\tau} g_i - \frac{i-1}{|\mathbf{y}|/|\mathbf{x}|} \tag{6}$$
$$\text{where } \tau = \text{argmin}_i (g_i = |\mathbf{x}|)$$

$|\mathbf{x}| = |\mathbf{y}|$이면서 인코더 스텝이 디코더 스텝보다 k만큼 앞서 있을 경우, AL의 값은 k로 직관적인 해석을 줄 수 있다. 또한, AL 지표는 문장의 길이에 민감하게 반응하지 않아, 서로 다른 동시 시스템에 대한 비교를 용이하게 한다.

3.2.4 Differential Average Lagging, DAL

Average Lagging은 지연을 측정하기에 굉장히 좋은 지표이다. 하지만, AL은 식 (6)의 τ로 인하여 미분 불가능하여, 손실 함수를 정의해 최적화하는 것이 불가능하다. 이러한 문제점을 극복하기 위해 미분 가능한 Differential Average Lagging (DAL)[27]이 제안되었다. DAL의 식은 식 (7)와 같다.

$$DAL = \sum_{i=1}^{|\mathbf{y}|} g_i' - \frac{i-1}{|\mathbf{y}|/|\mathbf{x}|} \tag{7}$$

$$\text{where } g_i' = \begin{cases} g_i & i = 1 \\ \max_i(g_i, g'_{i-1} + \frac{|\mathbf{y}|}{|\mathbf{x}|}) & i > 1 \end{cases}$$

DAL에서는 이전 발생한 최소 지연 g'을 지속적으로 추적한다. 이를 통해, $|\mathbf{x}| = |\mathbf{y}|$ 이면서 인코더 스텝이 디코더 스텝보다 k만큼 앞서 있을 경우, AL과 동일한 k 값을 반환할 수 있도록 설계되었다.

4. 고정 정책 기반 동시통번역

고정 정책 기반 동시통번역은 학습하지 않는 규칙 기반 정책을 뜻한다. 해당 정책은 매우 간단하면서, 원하는 지연 성능으로 조절할 수 있으며, 동적 정책 기반 번역보다 해석이 용이하다는 장점이 있다. 또한, 동적 정책 기반에서 발생하는 학습의 오류를 받지 않으면서 빠른 학습을 진행할 수 있다.

더하여, 고정 정책 기반 시스템은 원하는 지연 성능으로 조정이 가능하다는 장점으로 지연을 자유롭게 조절할 수 없는 동적 정책 기반의 지연 대비 품질 성능 비교 모델로 용이하다.

4.1 wait-if-* 정책 기반 동시통번역

[2]의 연구는 학습하지 않는 고정 정책인 wait-if-* 정책을 제안한다. 이는 규칙 기반의 에이전트를 이용하여 정의된 일부 규칙을 통해 각 스텝에서 읽기/쓰기를 결정한다.

wait-if-* 연구에서는 두 가지 기준 Wait-If-Worse (Λ_{W})와, Wait-If-Diff (Λ_{D})를 정의한다.

Wait-If-Worse란 주어진 입력 토큰의 수가 증가했을 경우, 가장 높은 확률을 가지고 있는 토큰의 로그 확률 변화에 따라 읽기/쓰기의 행동을 결정하는 정책이다. 만약 로그 확률이 감소했다면, 이는 입력 토큰의 수가 증가했을 때의 타겟 토큰 예측 확률이 더 높다는 것을 의미하므로, 추가 입력 토큰을 더 기다리는 읽기 행동을 한다. 반대로, 로그 확률이 증가했다면, 이는 해당 스텝이 적절한 입력 토큰의 수를 결정한 것으로 판단하여 쓰기 행동을 한다. Wait-If-Worse 정책 Λ_{W}을 스텝 t에 대한 식으로 나타내면 식 (8)와 같다

$$\Lambda_{\mathrm{W}} = \begin{cases} R & \log P(\hat{y}_t | \hat{y}_{<j}, \mathbf{x}_{<i}) < \log P(\hat{y}_t | \hat{y}_{<j}, \mathbf{x}_{<i+1}) \\ \mathrm{W} & \log P(\hat{y}_t | \hat{y}_{<j}, \mathbf{x}_{<i}) \geq \log P(\hat{y}_t | \hat{y}_{<j}, \mathbf{x}_{<i+1}) \end{cases} \tag{8}$$

$$\text{where} \hat{y_t} = \arg max_y P(y | \hat{y}_{<j}, \mathbf{x}_{<i+1})$$

식 (8)에서 R, W는 각각 READ와 WRITE 행동을 의미하며, i는 READ 행동의 수, j는 WRITE 행동의 수이다.

Wait-If-Diff란 더 많은 입력 토큰을 읽어도, 생성한 타겟 토큰 $\hat{y}$이 변경되지 않는 경우에만 토큰을 생성하는 것이다. Wait-If-Diff 정책 Λ_{D}을 스텝 t에 대한 식으로 나타내면 식 (9)와 같다.

$$\Lambda_{\mathrm{D}}(t) = \begin{cases} \mathrm{READ} & \hat{y_t} \neq \hat{y}_{t+1} \\ \mathrm{WRITE} & \hat{y_t} = \hat{y}_{t+1} \end{cases} \tag{9}$$

$$\text{where} \hat{y_t} = \arg max_y P(y | \hat{y}_{<j}, \mathbf{x}_{i+1})$$

Wait-If-* 정책은 학습이 없는 정책을 사용한다는 점에서 고정 정책 기반 시스템으로 분리할 수 있으나, 일정한 규칙으로 행동을 결정하지 않는다는 점에서는 동적 정책 기반 시스템의 시발점으로 볼 수 있다.

4.2 wait-k 정책 기반 동시통번역

[3] 연구는 기존 기계번역 모델을 prefix-to-prefix로 학습하는 대표적인 고정 정책 중 하나인 wait-k 전략을 제안한다. 기존 신경망 기계번역과는 다르게, 초기에 인코더는 오직 k개의 입력 토큰을 받고 연속적으로 입력 토큰 읽기와 타겟 토큰 쓰기를 반복한다. 이 과정은 기존 기계번역 모델과 마찬가지로 〈eos〉 토큰을 생성할 때까지 반복하며, 매 스텝마다 인코더의 입력이 변하기 때문에 자기 회귀 방식으로 생성하지 않는다는 차이점이

있다. wait-k 정책에서 인코더는 소스 문장 $\mathbf{x}=(x_1,...,x_n)$을 입력으로 받아, 타겟 문장 $\mathbf{y}=(y_1,...,<\text{eos}>)$를 얻는다. 이때, 최종 타겟 문장 $\mathbf{y}$의 확률은 식 (10)와 같다.

$$P_{\text{wait}-k}(\mathbf{y}|\mathbf{x};k;\theta)=\Pi_{t=1}^{|\mathbf{y}|}P(y_i|\mathbf{x}_{<i+k},\mathbf{y}_{<i};\theta) \quad (10)$$

따라서, 병렬 문장 $(\mathbf{x},\mathbf{y}^*)$에 대한 Cross-entropy를 적용한 음의 조건부 확률의 손실 함수는 식 (11)와 같다.

$$\mathcal{L}_{\text{wait}-k}(D)=-\sum_{(\mathbf{x},\mathbf{y}^*)\in D}\log P_{\text{wait}-k}(\mathbf{y}^*|\mathbf{x};\theta) \quad (11)$$

wait-k 정책에서, 모델은 식 (11)을 최소화하는 방향으로 학습이 이루어진다. 즉, 동시통번역은 학습을 진행할 때, 입력 문장의 일부만으로 타겟 문장의 일부를 학습하는 구조를 가진다. 이러한 이유로, 동시통번역 시스템의 구조는 시퀀스-대-시퀀스 (Sequence-to-sequence) 구조가 아닌, 접두사-대-접두사 (Prefix-to-prefix) 구조로 표현한다.

wait-k 정책을 적용한 모델의 성능은 $k\approx\infty$ 경우, 기존 기계번역 모델의 성능과 동일해지며 이는 동시통번역 모델이 낼 수 있는 최대의 번역 성능을 암시한다. 또한 $k=0$ 경우, 입력 문장 없이 생성과 읽기를 반복하는 정책으로 모델의 지연 성능 관점으로 지연을 최소화할 수 있는 이상적인 정책 (Ideal policy)으로 불린다.

5. 동적 지연 정책 기반 동시통번역

wait-k 정책 등의 고정 정책은 k를 설정하여 지연을 자유롭게 조절할 수 있으나, 미래의 토큰을 공격적으로 예측한다는 문제점이 있다. 이와 반대로, 동적 정책은 지연을 자유롭게 조절할 수 없지만, 문장마다 효율적인 정책을 제시하여 낮은 지연에서도 높은 번역 성능으로 모델을 개선하기 위해 제안되었다.

5.1 행동 시퀀스 생성 기반 동적 정책

행동 시퀀스 (Action sequence)란, 병렬 문장 쌍 $(\mathbf{s},\mathbf{t})$에 대하여 $\mathbf{s}$를 $\mathbf{t}$로 번역하는 읽기/쓰기 (READ/WRITE)를 정의한 일련의 시퀀스이다.

[7] 연구는 사전학습된 번역 모델을 사용하여 각 문장 쌍에 대한 행동 시퀀스를 정의한다. 행동 시퀀스를 생성할 때, 사전학습된 번역 모델 M이 주어진 m개의 입력 토큰 $x_1,...,x_m$만으로 정답 타겟 토큰 y_t를 유추할 수 있다면, 이는 동시통번역 모델이 주어진 입력 토큰이 정답 토큰을 예측하기에 충분한 정보를 가지고 있다는 것으로 가정한다.

이러한 가정으로, 사전학습된 번역 모델에 입력 단어를 순차적으로 입력하여 타겟 단어를 예측하고 예측한 타겟 단어를 확률 순으로 정렬한다. 그런 다음, 실제 정답 토큰이 사전학습 번역 모델 M이 예측한 타겟 단어 중 상위 랭크에 속한다면 쓰기 행동 WRITE을 행동 시퀀스에 추가하고, 실제 정답 토큰이 사전학습 번역 모델 M이 예측한 타겟 단어가 상위 랭크에 속하지 않는다면 읽기 행동 READ을 행동 시퀀스에 추가한다. 병렬 문장 쌍 $(\mathbf{s},\mathbf{t})$에 대한 행동 시퀀스를 생성하는 알고리즘은 Fig. 3에 있다.

생성된 행동 시퀀스를 반영하여 정책을 학습할 경우, 동일한 지연 수치에서 고정 정책 wait-k 대비 좋은 번역 성능의 동적 정책을 학습할 수 있다. 이러한 기존 사전학습 기계번역 모델을 이용한 행동 시퀀스 생성 연구를 바탕으로, [8] 등의 사전학습 모델을 이용한 후속 정책 학습 연구들이 제안되었다.

Algorithm 1 Action sequence generation

Input: sentence pair $(\boldsymbol{s},\boldsymbol{t})$, model M, rank r
Initialize $\text{idx}_s \leftarrow 1$, $\text{idx}_t \leftarrow 0$, $\text{action}_0 = \text{READ}$
while $\text{idx}_t < |\mathbf{t}|$ **do**
  if $\text{idx}_s = |\mathbf{s}|$ or $rank_M(t_{\text{idx}_t}|\boldsymbol{s}_{\leq idx_s}) \leq r$ **then**
    $\text{action}_{\text{idx}_s+\text{idx}_t} = \text{WRITE}$
    $\text{idx}_t \leftarrow \text{idx}_t + 1$
  else
    $\text{action}_{i+j} = \text{READ}$
    $\text{idx}_s \leftarrow \text{idx}_s + 1$
return action

Fig. 3. The algorithm of the generating action sequences

5.2 어텐션 기반 동적 정책

어텐션 기반 동적 정책은 인코더의 어텐션에 단조 어텐션 메커니즘을 적용하고 학습하여 정책을 조절하는 것이다.

hard monotonic attention 메커니즘은 RNN 기반 인코더-디코더 모델에 대한 온라인 선형 시간 디코딩

(Online linear time decoding)을 달성하기 위해 처음 도입되었고, 해당 어텐션 메커니즘을 이용한 초기 어텐션 기반 동시통번역 시스템의 연구가 활발히 이루어졌다[28].

초기의 어텐션 기반 동시통번역 시스템은 순환신경망(Recurrent Neural Network)를 기반으로 연구되어, 단일 어텐션으로 정책을 조절하는 Monotonic Chunk Attention (MoCha)[9], Monotonic Infinite Lookback Attention (MILk)[10] 등의 단조 어텐션 기반 동적 정책 학습 연구로 당시 기계번역 시스템의 최고 성능을 기록한 트랜스포머의 동시통번역 성능을 능가하였다. 최근의 연구에는 [11] 등의 멀티헤드에 적용할 수 있는 단조 어텐션 연구로 트랜스포머에 단조 어텐션 메커니즘을 적용하여 더 좋은 지연 대비 품질 절충점을 달성한다.

5.2.1 Monotonic Multihead Attention

이전에 동시통번역과 같은 온라인 환경에서 단조 어텐션을 활용하여 행동을 결정하는 대표적인 연구로 Monotonic chunk attention[9], Monotonic infinite lookback attention[10] 등의 연구가 진행되었다. 이러한 연구는 모두 순환신경망에서 사용 가능한 단일 어텐션을 기반으로 연구되어 트랜스포머 구조에 적용하지 못한다는 문제를 가지고 있다. 이러한 문제에 착안하여, 기계번역에서 최상의 성능을 가지는 트랜스포머 구조에 적용하기 위해 멀티헤드 어텐션을 기반으로 진행된 대표적인 단조 어텐션 메커니즘이 단조 멀티헤드 어텐션 (Monotonic multihead attention) 이다[11].

멀티헤드 어텐션은 모든 디코더 레이어가 여러 개의 헤드를 가질 수 있도록, 각 헤드에 서로 다른 어텐션 분포 (Attention distribution)를 계산할 수 있도록 한다. 쿼리 Q, 키 K, 벨류 V 행렬에 주어졌을 때, scaled dot product attention 을 적용한 멀티헤드 어텐션 $\mathrm{MultiHead}(Q, K, V)$는 다음과 같이 정의된다.

$$\mathrm{Attention}(Q, K, V) = \mathrm{Softmax}(\frac{QK^T}{\sqrt{d_k}}V) \quad (12)$$

$$\mathrm{head}_h = \mathrm{Attention}(QW_h^Q, KW_h^K, VW_h^V) \quad (13)$$

$$\mathrm{MultiHead}\ (Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head}_H)W^O \quad (14)$$

입력 문장 토큰 개수가 T이고 각 토큰에 해당하는 인코더 state가 $\mathbf{m} = m_1, ..., m_T$이면, i번째 디코딩 스텝에서 토큰 y_i를 생성할 때, 디코더는 $t_i = j$에 해당하는 H개의 어텐션 헤드를 가진 L개의 디코더 레이어에 대한 MMA의 h번째 헤드의 인코더-디코더 어텐션의 선택 프로세스는 다음과 같다.

$$e_{i,j}^{l,h} = (\frac{m_j W_{l,h}^K (s_{i-1} W_{l,h}^Q)^T}{\sqrt{d_k}})i,j \quad (15)$$

$$p_{i,j}^{l,h} = \sigma(e_{i,j}) \quad (16)$$

$$z_{i,j}^{l,h} = \mathrm{Beroulli}(p_{i,j}) \quad (17)$$

여기서, $\sigma(\cdot)$은 로지스틱 시그모이드 함수이다. 이전 MoCha, MILk와는 다르게, 이러한 선택 프로세스를 적용하면 모든 레이어의 각 헤드는 독립적인 선택을 할 수 있다. 즉, 이전 단조 어텐션 메커니즘은 단일 헤드만을 사용하여 새로운 정보가 입력되면 이전 정보를 유지할 수 없지만, 멀티헤드 단조 어텐션 메커니즘은 일부의 헤드는 새로운 입력 토큰에 어텐션을 하는 동시에, 다른 일부의 헤드는 이전에 입력된 토큰에 어텐션하여 이전의 정보를 유지할 수 있다.

인코더 스테이트에 헤드를 할당하는 선택 프로세스의 과정은 MMA-Hard (MMA-H)와 MMA-Infinite Lookback (MMA-IL)의 두 가지로 나눌 수 있다.

MMA-H는 멀티헤드에서 각 헤드에 대해 예상 정렬 (Expected alignment) α을 계산한다. 예상 정렬의 계산 식은 아래와 같다.

$$\alpha_{i,j} = p_{i,j}(\frac{(1-p_{i,j-1})\alpha_{i,j-1}}{p_{i,j-1}} + \alpha_{i-1,j}) \quad (18)$$

MMA-IL는 각 레이어의 각 헤드에 대해 소프트맥스 에너지 $u_{i,j}^{l,h}$를 계산한다. 소프트맥스 에너지의 계산 식은 다음과 같다.

$$u_{i,j}^{l,h} = (\frac{m_j \widehat{W}_{l,h}^K (s_{i-1} \widehat{W}_{l,h}^Q)^T}{\sqrt{d_k}})i,j \quad (19)$$

MMA-H의 경우, 각 어텐션 헤드는 하나의 인코더 스테이트에 어텐션한다. 이는 지연 사항을 자유롭게 조정할 수 있음을 의미한다. 반면에, MMA-IL의 각 어텐션 헤드는 광범위한 이전 인코더 스테이트에 어텐션을 하

여 많은 정보를 활용할 수 있지만, MMA-H와 같이 지연 사항을 자유롭게 조정할 수 없다는 제약이 있다.

6. 실험 및 실험결과

6.1 데이터셋

학습, 검증 및 테스트 데이터셋으로 AI hub[1] 에서 제공하는 한국어-영어 번역 병렬 말뭉치를 사용하였다. 해당 데이터셋은 뉴스, 간행물, 구어체, 대화체 등의 모든 도메인을 포함하고 있고 대략 160K개의 문장으로 이루어져 있다. 여기에서 우리는 158K개의 문장을 학습 데이터셋으로, 1K개의 문장을 검증 데이터셋으로, 나머지 1K개를 테스트 데이터셋으로 사용하였다. 한국어와 영어 데이터셋 모두 센텐스피스(Sentencepiece)[29]를 사용하여 영어, 한국어 5만개 BPE 어휘 사전을 구축하였다[30].

6.2 모델 및 정책

고정 정책인 wait-k 모델은 지연을 조정할 수 있으므로, 동적 정책 대비 지연 발생 지점과 지연의 문제점을 정확하게 파악할 수 있다. 한국어에서 발생하는 문제점을 확인하기 위해, 본 절에서는 지연을 조정할 수 있는 고정 정책인 wait-k 정책을 사용하여 모델을 학습하였다. 모든 모델은 fairseq[2]을 이용한 재구현을 통해 학습하였다. wait-3, 5, 7, 9, 11의 총 5개의 k값에 따른 모델을 학습하여 성능을 평가하고, [11]의 max token을 제외한 하이퍼 파라미터를 동일하게 사용하였으며, 본 연구에서는 max token 대신 512 크기의 batch size를 사용하였다. 또한, 한국어와 영어의 임베딩 레이어를 공유하였으며, 5만 개의 어휘 사전을 구축하여 학습하였다.

6.3 평가 방법

지연 지표로 AP, AL, DAL을 사용하였고, 번역 성능은 BLEU score를 사용하였다.

6.4 실험 결과

Fig. 4, Fig. 5, Fig. 6은 wait-3부터 wait-11까지의 모델과 전체 데이터를 기존의 기계번역 방법으로 번역한 것의 지연 대비 번역 성능이다.

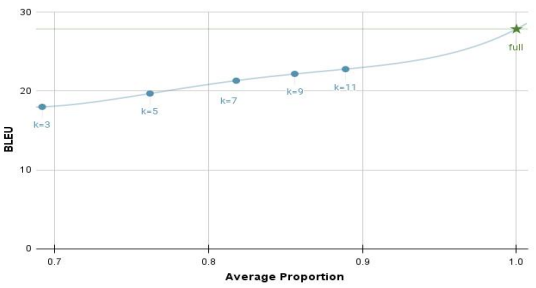

Fig. 4. Translation quality (BLEU) against average proportion on Korean → English simultaneous system

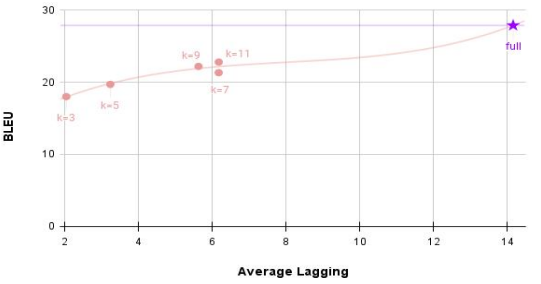

Fig. 5. Translation quality (BLEU) against average lagging on Korean → English simultaneous system

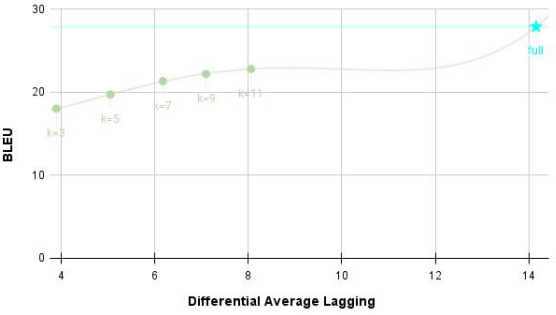

Fig. 6. Translation quality (BLEU) against differential average lagging on Korean → English simultaneous system

실험결과, 동시통번역에서 한국어 → 영어 번역은 다른 언어에 비해 불필요한 지연과 번역 성능에 문제를 겪는다. 한국어는 지연 값이 비교적 높은 것을 알 수 있다. wait-3 모델은 비교적 낮은 지연을 위한 정책임에도, 지연 지표가 비교적 높은 수치를 기록한다. 이는, 한국

1) https://aihub.or.kr/

2) https://github.com/pytorch/fairseq

어가 토큰화 과정에서 토큰의 길이가 상당히 길어지게 되어 발생하는 지연 문제임을 예상할 수 있다. 또한, 번역 지표인 BLEU 점수도 번역 성능에서도 기존 번역 모델에 비해 성능이 많이 부족하다는 문제가 있다.

이러한 문제는 이전에 예상했듯이 SOV → SVO 번역에서 동사를 예측해야 하는 문제로 인하여 발생하는 것이라 예상한다. 기존 다른 언어에 대한 동시통번역 연구 결과와 본 논문의 한국어에 대한 동시통번역 연구 결과에서의 차이점은 오직 어휘사전의 구성이다. 이는, 한국어에서 불필요한 지연은 어휘사전을 참조하여 토큰을 생성하는 과정에서 발생한다는 것을 의미하는 것으로, 토크나이징 방법에 따라 출력 문장의 길이가 입력 문장의 길이에 비해 상당히 길어져 발생하는 것으로 볼 수 있다. 다른 언어와 비슷한 지연성능을 달성하기 위해서는, 실험 설정의 유일한 차이점인 어휘사전을 효율적인 기법으로 토큰화 방법을 적용하여 다른 언어와 마찬가지로 생성 단계의 수를 맞춰야 한다. 대표적인 해결책으로, 토큰별 등장 횟수가 높은 순으로 n-gram을 구성하여 어휘 사전에 추가하여 학습하는 방법으로 출력 문장의 길이를 줄이는 방법을 고려해볼 필요가 있다[31]. 혹은, 고정 정책이 아닌 동적 정책을 적용하여 한국어 번역에서 지연을 줄일 수 있는 정책을 학습하는 방법이 필요할 것으로 보인다.

또한, 비교적 낮은 번역 성능의 원인으로 SOV 언어인 한국어를 SVO 언어인 영어로 번역할 때 과감히 동사를 예측해야 하는 문제로 인해 발생한다. 이전 문장의 문맥을 반영하는 모델로 구조를 수정하거나, 문장에 노이즈를 추가하여 학습한다면[32], 이전 문장들의 문맥을 통해서 동사를 예측하는 것에 도움을 줄 수 있을 것으로 생각하며 번역 성능과 지연 성능을 동시에 향상할 수 있을 것으로 기대된다.

이러한 동기들을 바탕으로, 추후 다른 언어의 동시 음성번역과 마찬가지로 한국어 또한 직관적인 추론 결과를 확인할 수 있는 cascaded 구조에서 text-to-text에 관한 지연 대비 품질의 절충점을 가지는 정책을 학습하는 연구를 해야 한다. 또한, 이를 바탕으로 지연 대비 품질 성능을 개선해야 한다. 더하여, 정책을 학습하는 전략을 end-to-end 동시 음성번역에 적용하는 연구를 진행하고 end-to-end의 성능 향상에 관한 연구가 추가적으로 이루어져야 한다.

7. 결론

동시 통역은 언어적 장벽을 없앨 수 있는 게임 체인저(Game changer) 기술이다. 동시통번역 연구는 후에 매우 중요시하는 기술이 될 것이 자명하나, 현재 국내에서 동시통번역 연구자의 수는 극히 드물다. 더욱이, 기존 기계번역 내에서 발생하지 않는 한국어 고유의 문제점이 동시통번역 내에서 발생하므로 연구의 필요성이 더욱더 강조된다. 본 연구에서, 한국어 동시통번역은 다른 언어의 동시통번역에 비해 많은 지연 발생으로 한국어에 적합한 연구가 필요하다는 것을 강조한다. 본 연구가 한국어에서 동시 번역의 연구가 도움이 되기를 희망한다.

REFERENCES

[1] Vaswani et al. (2017). Attention is all you need. *Advances in neural information processing systems,* pp. 5998-6008.

[2] K. Cho & M. Esipova. (2016). Can neural machine translation do simultaneous translation? *arXiv preprint arXiv:1606.02012.*

[3] Ma et al. (2018) Stacl: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. *arXiv preprint arXiv:1810.08398.*

[4] B. Zheng, K. Liu, R. Zheng, M. Ma, H. Liu & L. Huang. (2020). Simultaneous translation policies: From fixed to adaptive. a*rXiv preprint arXiv:2004.13169.*

[5] N. Arivazhagan, C. Cherry, W. Macherey & G. Foster. (2020). Retranslation versus streaming for simultaneous translation. *arXiv preprint arXiv:2004.03643.*

[6] M. Elbayad, L. Besacier & J. Verbeek. (2020). Efficient wait-k models for simultaneous machine translation. *arXiv preprint arXiv:2005.08595.*

[7] B. Zheng, R. Zheng, M. Ma & L. Huang, "Simpler and faster learning of adaptive policies for simultaneous translation," *arXiv preprint arXiv:1909.01559,* 2019.

[8] S. Li, J. Hu, B. Wang, X. Shi & Y. Chen. (2021). Xmu's simultaneous translation system at naacl 2021. *Proceedings of the Second Workshop on Automatic Simultaneous Translation,* pp. 19-23.

[9] C.-C. Chiu & C. Raffel. (2017). Monotonic chunkwise attention," arXiv preprint arXiv:1712.05382.

[10] Arivazhagan et al. (2019). Monotonic infinite lookback attention for simultaneous machine translation. *arXiv preprint arXiv:1906.05218*.

[11] X. Ma, J. Pino, J. Cross, L. Puzon & J. Gu. (2019). Monotonic multihead attention. a*rXiv preprint arXiv:1909.12406*.

[12] H. Satija & J. Pineau. (2016). Simultaneous machine translation using deep reinforcement learning. *ICML 2016 Workshop on Abstraction in Reinforcement Learning*.

[13] J. Gu, G. Neubig, K. Cho & V. O. Li. (2016). Learning to translate in real-time with neural machine translation. *arXiv preprint arXiv:1610.00388*.

[14] B. Zheng, R. Zheng, M. Ma & L. Huang. (2019). Simultaneous translation with flexible policy via restricted imitation learning. *arXiv preprint arXiv:1906.01135*.

[15] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu & Z. Chen. (2017). Sequence-to-sequence models can directly translate foreign speech. a*rXiv preprint arXiv:1703.08581*.

[16] R. Ye, M. Wang & L. Li. (2021). End-to-end speech translation via cross-modal progressive training. *arXiv preprint arXiv:2104.10380*.

[17] C. Wang, Y. Wu, S. Liu, M. Zhou & Z. Yang. (2020). Curriculum pre-training for end-to-end speech translation. *arXiv preprint arXiv:2004.10093*.

[18] B. Zhang, I. Titov, B. Haddow & R. (2020). Sennrich. Adaptive feature selection for end-to-end speech translation. *arXiv preprint arXiv:2010.08518*.

[19] H. Nguyen, F. Bougares, N. Tomashenko, Y. Est`eve & L. Besacier. (2020). Investigating self-supervised pre-training for end-to-end speech translation. *Interspeech 2020*.

[20] Ren et al. (2020). Simulspeech: End-to-end simultaneous speech to text translation. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* pp. 3787-3796.

[21] X. Ma, J. Pino & P. Koehn. (2020). Simulmt to simulst: Adapting simultaneous text translation to end-to-end simultaneous speech translation. *arXiv preprint arXiv:2011.02048*.

[22] A. Karakanta, S. Papi, M. Negri & M. Turchi. (2021). Simultaneous speech translation for live subtitling: from delay to display," *arXiv preprint arXiv:2107.08807*.

[23] X. Ma, Y.Wang, M. J. Dousti, P. Koehn & J. Pino. (2021). Streaming simultaneous speech translation with augmented memory transformer. I*CASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 7523-7527.

[24] D. Liu, M. Du, X. Li, Y. Hu & L. Dai. (2021). The ustc-nelslip systems for simultaneous speech translation task at iwslt 2021. *arXiv preprint arXiv:2107.00279*.

[25] K. Papineni, S. Roukos, T. Ward & W.-J. Zhu. (2002). Bleu: a method for automatic evaluation of machine translation," *Proceedings of the 40th annual meeting of the Association for Computational Linguistics,* pp. 311-318.

[26] S. Banerjee & A. Lavie. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization,* pp. 65-72.

[27] C. Cherry & G. Foster. (2019). Thinking slow about latency evaluation for simultaneous machine translation. *arXiv preprint arXiv:1906.00048*.

[28] C. Raffel, M.-T. Luong, P. J. Liu, R. J. Weiss & D. Eck. (2017). Online and linear-time attention by enforcing monotonic alignments. *International Conference on Machine Learning,* pp. 2837-2846.

[29] T. Kudo & J. Richardson. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

[30] G. Kim. J. Son. J. Kim, H. Lee & H. Lim. (2021). Enhancing Korean Named Entity Recognition With Linguistic Tokenization Strategies. *IEEE Access, 9,* 151814-151823.

[31] C, Park. S. Eo. H. Moon & H. Lim. (2021, June). Should we find another model?: Improving Neural Machine Translation Performance with ONE-Piece Tokenization Method without Model Modification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers* (pp. 97-104).

[32] C. Park. K. Kim, Y. Yang, M. Kang & H. Lim. (2021). Neural spelling correction: translating incorrect sentences to correct sentences for multimedia. *Multimedia Tools and Applications, 80(26),* 34591-34608.

이 정 섭(Jungseob Lee) [학생회원]

· 2021년 8월 : 동국대학교 정보통신공학전공 (공학사)
· 2021년 10월 ~ 현재 : 고려대학교 Human-Inspired AI 연구소

· 관심분야 : Simultaneous Translation, Dialogue System, Machine Translation, Speech Translation
· E-Mail : cy951011@gmail.com

문 현 석(Hyeonseok Moon) [학생회원]

· 2021년 2월 : 고려대학교 수학과 및 인공지능학과(이학사, 공학사)
· 2021년 3월 ~ 현재 : 고려대학교 컴퓨터학과 석박사통합과정

· 관심분야 : Neural Machine Translation, Natural Language Processing
· E-Mail : glee889@korea.ac.kr

박 찬 준(Chanjun Park) [학생회원]

· 2019년 2월 : 부산외국어대학교 언어처리창의융합전공 (공학사)
· 2018년 6월 ~ 2019년 7월 : SYSTRAN Research Engineer
· 2019년 9월 ~ 현재 : 고려대학교 컴퓨터학과 석박사통합과정

· 관심분야 : Machine Translation, Data-centric AI, Grammar Error Correction, Deep Learning
· E-Mail : bcj1210@naver.com

서 재 형(Jaehyung Seo) [학생회원]

· 2020년 8월 : 고려대학교 영어영문학과 및 경영학과(문학사, 경영학사)
· 2020년 9월 ~ 현재 : 고려대학교 컴퓨터학과 석박사통합과정
· 관심분야 : Graph Encoder, Commense Reasoning
· E-Mail : seojae777@korea.ac.kr

어 수 경(Sugyeong Eo) [학생회원]

· 2020년 8월 : 한국외국어대학교 언어인지과학과, 언어와공학전공 (문학사, 언어공학사)
· 2020년 9월 ~ 현재 : 고려대학교 컴퓨터학과 석박사통합과정

· 관심분야 : Neural Machine Translation, Quality Estimation
· E-Mail : djtnrud@korea.ac.kr

이 승 준(Seungjun Lee) [학생회원]

· 2021년 2월 : 한국외국어대학교 산업경영공학과 (공학사)
· 2021년 7월 ~ 현재 : Human-inspired AI 연구소

· 관심분야 : Natural Language Processing, AI in Education, Text-Mining
· E-Mail : dzzy6505@gmail.com

구 선 민(Seonmin Koo) [학생회원]

· 2018년 3월 ~ 현재 : 건국대학교 컴퓨터공학과 (공학사)
· 2021년 9월 ~ 현재 : 고려대학교 Human-Inspired AI 연구소

· 관심분야 : AI, Machine Translation, Grammar Error Correction, Deep Learning, Natural Language Processing
· E-Mail : djtnrud@korea.ac.kr

임 희 석(Heuiseok Lim) [종신회원]

· 1992년 : 고려대학교 컴퓨터학과 (이학학사)
· 1994년 : 고려대학교 컴퓨터학과 (이학석사)
· 1997년 : 고려대학교 컴퓨터학과 (이학박사)

· 2008년 ~ 현재 : 고려대학교 컴퓨터학과 교수
· 관심분야 : 자연어처리, 기계학습, 인공지능
· E-Mail : limhseok@korea.ac.kr