신경망 번역 모델의 진화 과정

통계 기반 번역기가 end-to-end 방식의 신경망 기반 기계번역(neural machine translation, NMT)으로 바뀌고 실제 서비스에 적용되기 시작한 것은 불과 1년 전의 일이다. 인공지능(artificial intelligence, AI)은 'AI winter' 시기를 견뎌낸 후, GPU 성능의 성장과 함께 꽃을 피웠다. 다양한 문제에 적합한 뉴럴 네트워크들이 나왔고, 단순한 뉴럴 네트워크들이 다양한 방식으로 연결되어 더 크고 복잡한 구조를 만들어 내고 있다.

네트워크가 더 크고 복잡해지는 이유는 각 단위 뉴럴 네트워크들이 어려운 문제를 해결하는데 상보적으로 작용하기 때문이다. 이는 마치 생명체가 세포들이 모여서 기관을 이루고, 기관들이 모여 온전한 개체로 완성되는 것과 유사하다. NMT도 가장 전형적인 복합 구조를 가지는 뉴럴 네트워크 중 하나이며 이미지넷(ImageNet)의 영상 인식 기술과 유사하게 매우 짧은 시기에 다양한 진화 단계를 겪어왔다. 물론 지금도 계속 진화중이다. 이번 글에서는 번역에 사용되는 모델의 진화 과정을 통해 각 모델의 핵심 구조와 아이디어들이 어떻게 발전해 왔는지 살펴보도록 하겠다.

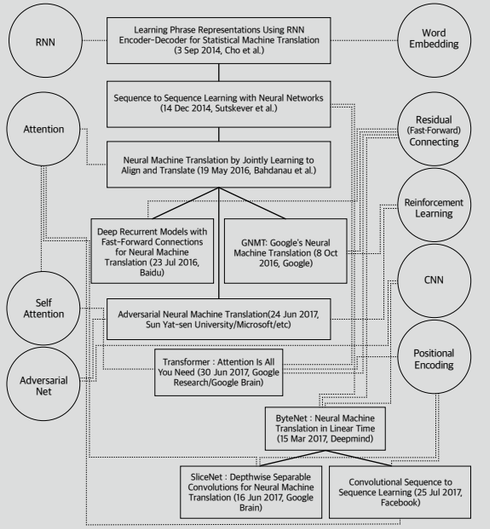
글 | 배재경 storm.bae@kakaocorp.com

짧은 지식으로 NMT의 전반적인 기술 발전 과정에 대하여 글을 쓰자니 부끄럼이 앞섭니다. 검색엔진 개발만 6~7년 하다가 AI쪽으로 넘어온 지 1년도 안되었지만 긍정적인 마인드로 열심히 하고 있습니다. 모든 것들이 공유되는 문화라서 마음만 먹는다면 누구든 빨리 따라잡을 수 있는 분야인 것 같습니다.

NMT 모델의 진화 과정

지금까지 이루어진 NMT 연구 결과를 한 장의 도표로 표현하는 것은 쉽지 않다. 그렇지만 주요 기반 뉴럴 네트워크 및 모델들 간의 상관 관계를 시간 축으로 그려 보면 많은 정보를 얻을 수 있다.

[그림 1] NMT 모델의 진화 과정

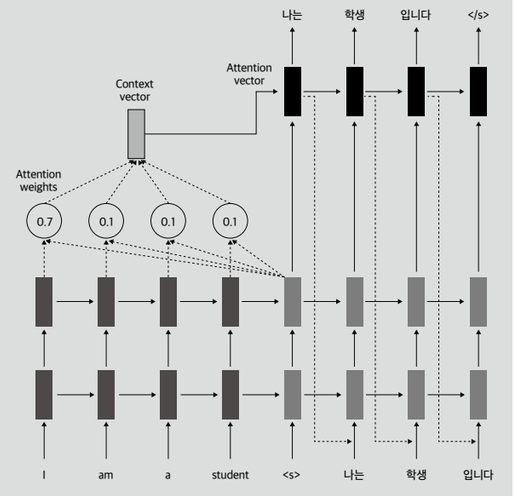


[그림 1] 에서 원은 주요 기반 모듈이고, 사각형은 NMT 모델이다. 각 모델은 기존 모델을 토대로 만들어지기도 하고 어느 정도 독립적으로 생성되기도 하는데, 그림에서 굵은 실선은 강한 영향 관계 또는 기존 모델을 토대로 했다는 의미이고 얇은 점선은 약한 영향 관계 또는 내부 모듈로 사용된 경우를 표현하고 있다.

첫번째 NMT 모델은 2014년 12월에 발표된 'Sequence to sequence learning with neural networks' 이다[*1]. 하지만 이 모델은 인코더-디코더(encoder-decoder)모델을 토대로 확장된 것이기 때문에 'Learning phrase representations using RNN Encoder-Decoder for statistical machine translation' 을 NMT의 시발점이라고 봐도 될듯 하다[*2]. Encoder-decoder 모델은 단위 정보(word 또는 token)의 시퀀스(sequence)를 입력값으로 받아서 고정 길이 vector representation을 생성한 후 이를 이용하여 또 다른 단위 정보의 sequence를 생성하는 모델이다. Sequence를 주로 다루기 때문에 최근에는 encoder-decoder 대신 sequence-sequence(seq2seq) 라는 용어를 많이 쓰고 있다. Sutskever의 모델에서 encoder와 decoder 각각은 RNN(recurrent neural network)으로 구현되며 단위 정보는 word embedding 을 통해 continuous value로 변환되어 사용된다.

Seq2seq 모델을 NMT에서만 사용하는 것은 아니다. Sequence 형태로 표현될 수 있는 정보를 다루는 어떤 곳이든 사용 가능한데, 예를 들어, 문서 요약, QA(Question Answering), Dialog 등이 모두 포함된다. 따라서 NMT의 구조를 파악하는 것은 자연어를 다루는 많은 문제들, 특히 문맥(context) 정보를 파악해야 하는 과제를 풀어나가는 가장 좋은 출발점이라고 볼 수도 있다.

[그림 2] RNN 기반 Attentional Seq2seq 구조



가장 단순한 형태의 seq2seq 모델은 성능이 그렇게 만족스럽지는 못했고 뉴럴 네트워크의 가능성을 확인한 정도였다. 기존의 가장 좋은 통계 기반 기계번역(statistical machine translation, SMT)과 경쟁할 만한 성과를 보여준 모델은 'Neural machine translation by jointly learning to align and translate' 이라 할 수 있다. 이 모델은 attention net을 활용하는 좀 더 복잡한 decoder를 사용한다[*3].

Attentional decoder는 decoding 시 매 time-step 별로 새로 생성될 토큰을 결정할 때 source sequence에서 가장 가까운 관계의 token을 결정한 후 이 정보를 활용하는 구조이다. 마치 두가지 언어를 구사할 수 있는 사람이 번역을 할 때 단어 별로 원문과 번역문을 매칭해 가면서 번역하는 것과 유사하다.

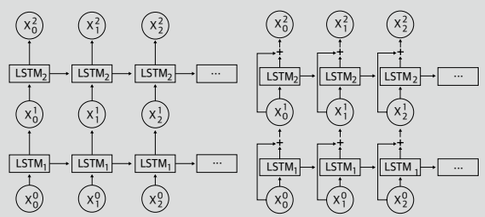
Attention을 도입함으로써 encoder의 결과를 고정 길이 벡터(vector)에 담아야 하는 문제도 해소되었다. 짧은 문장에 비해 긴 문장의 경우 더 많은 정보가 함축될 수밖에 없는데, 길이에 상관없이 고정 길이 벡터를 사용하는 것은 불합리하다는 것이다. 이 모델에서는 encoder의 매 time-step 시에 생성되는 벡터가 attention에 사용되므로 sequence 길이에 비례하여 더 많은 정보가 활용된다. 논문에서는 장문 번역의 성능이 높아진 결과를 attention 도입의 효과로 서술하고 있다.

Encoder, decoder의 각 RNN은 동일 구조가 반복적으로 쌓인 구조(multi-RNN)인데, 이렇게 할 경우 단일 layer에 비하여 좀 더 복잡하고 다양한 특징(feature)을 추출할 수 있다. RNN의 각 셀은 LSTM(long short-term memory) 또는 GRU(gated recurrent unit)를 사용한다. 논문에서는 추가적인 아이디어로 bidirectional RNN encoder를 제안하고 있는데, 이는 양방향의 이력(history) 정보를 모두 활용하여 놓치는 정보를 최소화려는 의도이다. [그림 2]는 이 모델의 구조를 보여준다. 이 그림에서는 복잡도를 줄이기 위하여 unidirectional RNN을 가정하였다.

드즈미트리 바다나우(Dzmitry Bahdanau)의 논문 이전에도 여러 기업에서 NMT 연구가 활발이 이루어졌지만, 이 논문을 계기로 좀 더 적극적으로 바뀌었다. 특히 구글(Google)과 바이두(Baidu)의 물밑 경쟁은 눈여겨 볼만하다. 먼저 바이두에서 추가적인 아이디어를 통해 번역 성능을 높인 논문(Deep recurrent models with fast-forward connections for neural machine translation)을 발표했다[4]. 핵심적인 내용은 fast-forward connection(구글에서는 residual connection으로 명명)의 도입인데 기존 encoder/decoder에서 Multi-RNN을 사용할 때 layer가 3~4개 이상인 경우에 학습이 잘 안되던 문제를, fast-forward connection을 통해 8개 layer 이상도 학습이 가능하도록 만든 것이다. Layer를 깊게 가져 가려는 이유는 더 풍부한 특징을 추출하여 성능을 높이기 위함인데, 너무 깊을 경우 기울기 값이 소실되는 문제(gradient vanishing)로 학습이 안되는 경우가 있었다. Fast-forward connection을 통해 이 문제를 해소하였고, 그 결과 드디어 번역 성능이 SMT를 능가하게 되었다. Fast-forward connection은 복잡한 뉴럴 네트워크가 아니라 n번째 layer의 입력이 n+1 번째 layer의 입력에 같이 들어가도록 추가 connection을 하나 두는 방식인데 CNN에서 먼저 사용되어 효과를 본 것을 RNN에도 유사하게 적용한 것이다.

아래 그림을 보면 단순 RNN과 fast-forward(residual) connection 이 있는 RNN의 차이를 확인할 수 있다.

[그림 3] Residual Connection의 형태



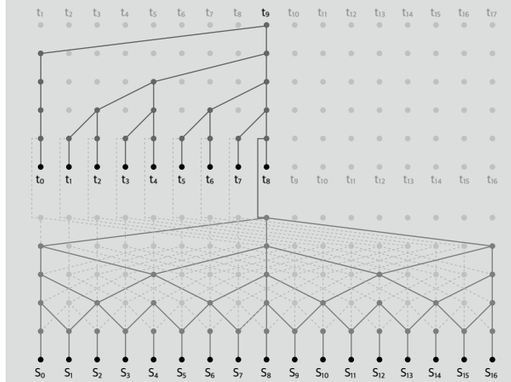
구글이 NMT 핵심 연구 분야에서 앞서가고 있었음에도 불구하고, 바이두가 한발 먼저 최고 성능의 번역 모델을 발표한 상황이 되었고, 이에 대한 대응으로 구글은 빠르게 실제 서비스를 런칭하는 방향으로 전략을 수정하였다. 논문으로 실험 결과를 발표하는 것과 실제 서비스화 하는 것에는 큰 차이가 있는데, 학습 시간도 이슈지만 응답 속도 및 처리량에 대한 고려를 하기 위해서는 많은 추가 작업이 필요하기 때문이다.

구글은 당시 유효하다고 판단했던 여러가지 기반 뉴럴 네트워크를 NMT 모델에 적용하였고, 특히 빠른 학습 속도를 위하여 하드웨어에 최적화된 model parallelism을 구현하였다. 빠른 응답 속도를 위해 양자화(quantization)를 도입하였고, 번역 품질을 극대화 하기 위해 길이 정규화(length normalization), coverage penalty 등의 몇가지 예측 알고리듬(prediction algorithm)을 도입하였다. 전략은 성공했고 필자도 당시 GNMT 논문을 처음 접했을 때 공학적인 측면에서 감탄하지 않을 수 없었다. 구글에서 발표한 'Google's neural machine translation system' 이라는 논문은 'Bridging the gap between human and machine translation' 이라는 부제를 가지고 있는데 그만큼 성능에 자신이 있었기 때문이었을 것이다[5].

하지만 어느 정도 시간이 지난 후 필자는 GNMT 모델의 구조가 당시 하드웨어 스펙에 맞추느라 다소 부자연스러운 점이 있다고 생각했고, 이 구조가 많은 알고리듬을 조합해 낸 결과이기 때문에 장기적인 개선 작업 또한 쉽지는 않을 것이라고 판단했다. 아니나 다를까 구글은 RNN 기반이 아닌 새로운 구조의 모델을 조만간 발표하게 된다. 이들에 대해서는 잠시 후에 다룰 예정이다. 사실 그 사이 네이버가 구글보다 먼저 NMT를 번역 서비스에 적용했는데 발빠른 행보가 참으로 돋보였다. 그렇지만 모델이 공개되지 않아 그 구조를 파악할 수 없었고 글자 수 200자 제한을 꽤 오랜 기간 유지한 것도 아쉬운 부분이었다. 네이버, 구글 이외에도 여러 업체에서 NMT 기반 번역 서비스를 시작하게 되면서 번역은 비젼(vision) 분야와 함께 딥러닝의 주요 화두가 되었고 2017년 초부터 또다른 연구 성과들이 경쟁적으로 공개되었다. 그 중 가장 주목할 만한 것은 CNN 기반의 모델인 ByteNet이다[6].

딥마인드에서 개발한 이 모델은 논문의 제목이 'Neural machine translation in linear time'인 것에서 알수 있듯이 학습 시간을 선형 시간(linear time)에 가능하게 하는 것이 목적이다. 지금까지의 NMT 모델은 RNN 기반의 attentional seq2seq를 거의 정석처럼 사용했는데 attention net 때문에 학습 시간이 quadratic time(source sequence size * target sequence size)을 가지게 된다. 반면, ByteNet에서는 attention net을 사용하지 않으므로 linear time(c * source sequence size + c * target sequence size)에 학습이 가능하다(여기서 c는 constant value). ByteNet은 [그림 4]와 같이 CNN을 사용하여 encoder 위에 decoder가 스택처럼 쌓이는 네트워크 구조를 만들고 dynamic unfolding 이라는 기법을 통해 가변 길이 sequence를 생성해 낸다.

[그림 4] ByteNet 구조도

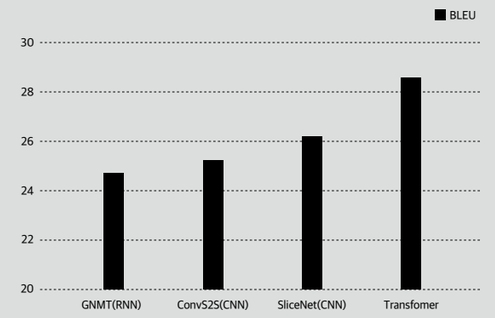


RNN에서는 필수적인 time-step과 step 간 정보의 기억(memorization)이 필요없게 되는데, 구조적인 특성상 병렬화의 여지가 훨씬 크고 멀리 떨어진 단위 정보 사이의 관계 특성(feature)을 더 잘 찾아낼 수 있다. 이 모델은 성능에 있어 기존 RNN 모델과 비교할 수준은 아니었지만, character to character 번역(단위 정보로 워드나 토큰이 아니라 character를 사용)에서는 최고의 성능을 보여주었다.

ByteNet 처럼 CNN 기반의 NMT 모델도 짧은 기간 동안 연구가 활발히 이루어져 RNN 기반 모델의 성능을 능가하는 모델이 나오기 시작했다. 그 중 두드러진 두가지 모델은 페이스북에서 공개한 Convolutional Sequence to Sequence Learning과 구글브레인에서 공개한 SliceNet이다[7][8]. 두 모델 모두 convolution net을 사용하고 positional encoding 과 attention net을 적용하였다. SliceNet이 약간 먼저 나오긴 했지만 convolution net의 구조와 attention net의 적용 형태가 약간 다를 뿐 서로 상당히 유사하다. 결국 attention net이 RNN에 적용되어 극적인 성능 향상이 이루어진 것처럼 CNN 방식에서도 유사한 과정이 진행되었다고 볼수 있다. CNN에서는 time-step이 없으므로 단위 정보의 위치 정보를 표현하기 위한 다른 방법이 필요한데 이를 위해 positional encoding을 사용한다.

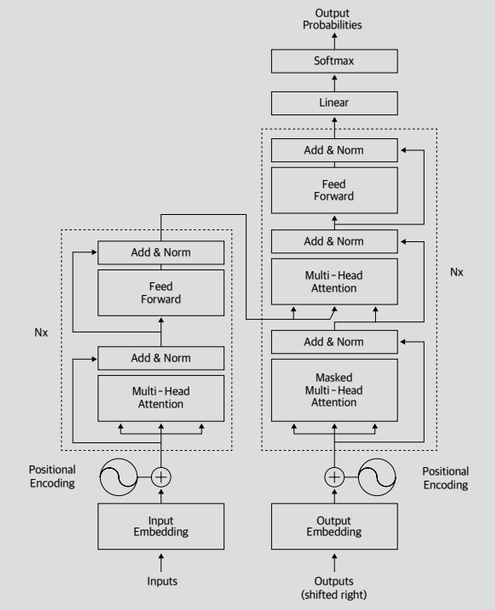
[그림 5]는 주요 모델들의 성능을 비교한 것이다[9]. 그동안 자연어 텍스트 처리에는 RNN이 적합하다는 관점이 우세했지만 이를 뒤엎는 결과가 나온 것을 확인할 수 있다. 필자는 이 결과를 보고 RNN의 시대가 벌써 저무는것 아닌가 하고 생각했는데 과연 그럴지는 두고 볼 일이다.

[그림 5] 모델별 영어와 독일어 번역 품질



비슷한 시기에 RNN, CNN 기반 모델 이외에 압도적인 성능을 보여준 모델이 하나 더 공개되었다. Transformer라 불리는 이 모델은 부제가 'Attention is all you need' 인데 RNN, CNN 모두 필요 없다는 말이다[10]. 뉴럴 네크워크는 attention net과 normalization, feed-forward net의 반복적인 구조로 이루어진 매우 매우 단순한 형태이다. 대신 attention net이 기존에 decoder의 sequence와 encoder의 sequence 간에 align을 맞춰주는 용도로 사용되었다면, 여기서는 추가로 encoder/decoder 각 layer의 입력 정보를 함축하는데 사용되는 방식으로 확장되었다. 이 때문에 Transformer에서 추가로 적용된 attention 방식을 self-attention이라고 부르기도 한다. [그림 6]은 Transformer의 구조를 보여준다.

[그림 6] Transformer 구조도



전형적인 encoder-decoder 모델의 동작 방식을 다시 되새겨 보자. Encoder가 input 정보를 vector representation으로 함축하고 decoder에서 이 정보를 바탕으로 최종 output을 생성하는데 지금까지의 모델은 input 정보의 함축을 위해 RNN이나 CNN을 사용했던 반면, Transformer에서는 단위 정보 각각의 상관 관계를 attention net 구조로 풀어내면서 정보를 함축한다. 이러한 구조만으로도 feature 정보를 충분히 잘 추출해내어 RNN과 CNN 보다 오히려 더 나은 성능을 보여준 점은 새로운 발전이 고정 관념을 깨는 것으로부터 출발하다는 좋은 실례를 보여주는 것이라 할 수 있다.

결국 RNN/CNN에 이어 self-attention 이라는 기반 뉴럴 네트워크가 가세하면서, encoder-decoder라는 큰 구조를 제외하면 이에 대한 구현체들은 얼마든지 다양한 방식으로 결정될 수 있다는 생각이 더 자연스럽게 이루어지게 되었다. Seq2seq는 RNN2RNN, RNN2CNN, CNN2CNN 뿐만 아니라 Any2Any로 고민될 수 있다. 사실 개발자 입장에서는 선택의 폭이 넓어진 것이 썩 달갑지 만은 않다. 특정 문제에 어떤 뉴럴 네트워크를 사용하는 것이 적합한지에 대한 명확한 근거가 없는 경우가 대부분이고 따라서 대부분 실험적으로 접근할 수밖에 없는데, 실험은 수많은 hyperparameter 최적화 과정뿐만 아니라 기반 뉴럴 네트워크의 다양한 조합들이 모두 고려되어야하기 때문에 부담이 가중된다고 볼 수 있다. 그렇지만 좀 더 큰 틀에서 본다면 특징 추출을 위한 가장 적합한 구조의 뉴럴 네트워크들이 다양하게 나오고 많은 연구자들에 의해 이들의 장단점, 특징들이 파악되어 감에 따라 AI가 완전한 black box에서 어느 정도 투명하고 제어 가능한 모습을 가지게 될 것이란 기대를 해본다. 그리고 추가로 AutoML이라는 분야의 연구도 활발히 진행되고 있는데 이는 뉴럴 네트워크 선택과 hyperparameter 튜닝의 자동화에 대한 연구로 AI 개발자들의 가려운 곳을 많이 긁어줄 수 있을 것으로 기대된다.

어떻든 self-attention net의 성능에 대해서는 어느 정도 증명이 되었다고 볼 수 있고, 그렇다면 이를 능가하는 새로운 기반 뉴럴 네트워크가 또 나올 것인지를 예측해 보는 것도 흥미로울 것 같아서 개인적인 의견을 달아 본다.

어떤 지도 학습(supervised learning) 모델(모델 A)의 power를 측정하기 위한 간단한 접근 방식 중 하나는 샘플을 충분히 많이 확보하여 기준 모델(모델 B)에 적용한 후 그 결과로 만들어진 데이터를 학습셋으로 이용하여 모델 A를 다시 학습하고 그 성능이 모델 B와 유사한지 아닌지 보는 것이다. 만일 성능이 서로 유사하다면 모델 A는 모델 B 보다는 약하지 않다고 판단할 수 있다. 그리고 위의 과정을 역으로 진행했을 때 성능이 유사하지 않다면 모델 A가 모델 B보다 나은 성능을 가진다고 말할 수 있을 것이다.

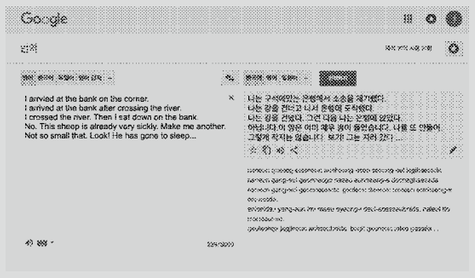
Transformer는 거의 복사기 수준으로 기존 모델을 모사해 낸다. 이런 측면에서 Transformer는 충분히 강력하다고 판단되고 이를 월등히 능가하는 모델은 쉽게 나오기 힘들 것이라는 예상을 해본다. 모델 간의 경쟁 과정은 거의 수렴 단계로 보이며 따라서 번역 관련해서는 전혀 새로운 형태의 강자가 나타가기 보다는 기존 뉴럴 네트워크를 토대로 더 넓은 문맥을 다루는 모델로 진화해 나가지 않을까 생각된다.

결론으로 넘어가기 전에 'Adversarial neural machine translation'에 대해서도 살짝 언급해야 할 것 같다"[11]. CNN 기반 모델들의 연구가 활발히 이루어지고 있는 동안 전혀 다른 학습 방식을 사용하는 접근도 이루어졌다. 지금도 여전히 활발한 연구가 이루어지고 있는 GAN(Generative Adversarial Networks)의 접근 방식을 NMT에 유사하게 적용한 모델인데, 여기서는 사람의 번역과의 유사도를 극대화하는 기존 학습 방식 대신 NMT 모델과, 이와 사람의 번역을 구별해 내는 CNN 기반의 adversary net을 도입하여 둘 사이를 경쟁 관계로 두고 서로 발전해 나가는 모델이다. RNN 기반의 seq2seq 모델을 토대로 하긴 했지만, 새로운 학습 방법을 통해 기존 모델의 성능을 개선한 것이다.

이처럼 딥러닝의 많은 기본 아이디어들은 그 사용성이 일반적인 경우가 많다. 어떤 아이디어가 유효하다면 그 쓰임새가 특정 모델에 국한되지 않는다는 의미이다. 지금까지 살펴 보았던 attention net, residual connection, positional encoding 등도 모두 그러한 예이다.

NMT가 짧은 기간 동안 큰 발전을 이루어 왔지만 아직 갈 길이 멀다. 예를 들어 다음 구글의 번역 결과를 살펴보자.

[그림 7] 구글 번역 결과 예제



[그림 7]의 예에서 'bank'라는 단어가 'river'와 같이 쓰일 때는 강둑이 더 적합하다. 또한 'Make me another.' 처럼 대명사가 들어간 문장을 제대로 번역해 내기도 힘들며, 구어체에서 많이 나타나는 짧은 어구나 문장들은 앞뒤 문맥을 더 넓게 봐야 정확한 번역이 이뤄질 수 있다.

카카오에서도 완전히 새로운 형태의 뉴럴 네트워크를 연구하기보다는 기존에 잘 동작하는 모델을 기반으로 문체나 더 넓은 문맥에 초점을 맞춰 모델을 연구 중이다. 해당 문장들을 카카오에서 실험 중인 모델로 다시 번역하면 다음과 같은 결과가 나온다.

I arrvied at the bank on the corner.	나는 모퉁이에 있는 은행에 도착했다.
I arrived at the bank after crossing the river.	나는 강을 건넌 후 강둑에 도착했다.
I crossed the river.	강을 건너 강둑에 앉았다.
Then I sat down on the bank.	
No. This sheep is already very sickly. Make me another.	아니, 이 양은 이미 병약해. 다른 양으로 만들어 줘.
Not so small that.	그렇게 작긴 않아요,
Look! He has gone to sleep…	보세요! 그는 잠들었어요.

짧은 예제라서 성급한 판단은 이르지만 가능성은 확인할 수 있다. 첫 두 문장의 'bank'의 번역은 RNN에 비하여 attention 기반 모델이 문맥을 더 잘 활용하는 예이고, 나머지 번역은 카카오에서 연구 중인 larger context 모델로 더 적절한 번역문을 만들어 낸 예제이다.

마치며

기계 번역이 인간의 수준을 따라잡기는 쉽지 않을 것으로 보인다. 언어는 수천 년간 독립적으로 형성된 문화를 반영하므로 언어간 1:1 매칭이 되지 않는 번역 규칙이 수없이 존재한다. 따라서 정확한 번역을 위해서는 문화를 이해해야 하고 이와 함께 역사/경제/과학/예술 등의 도메인 지식이 있어야 적절한 번역문을 생성해 낼 수 있다. 결국 사람은 단순히 텍스트 정보로만 번역하는 것이 아니라 수많은 추가 정보를 토대로 논리적인 유추의 과정을 거치면서 번역을 하게되는 것이다.

다행히 언어는 각 언어 별로 공통적인 규칙이 매우 많기 때문에 현재 기술로도 놀라운 성과를 내고 있다. 하지만 궁극의 번역 기술은 general AI 영역에 속한다고 볼 수 있다. 따라서 수년 내에 완벽한 번역을 해내는 AI 기술을 기대하는 것은 무리다. General AI도 언젠가는 탄생할 것이고 지구적 진화 과정의 시간 관점으로는 찰나에 해당하겠지만 기껏해야 한 세기를 살수 있는 인간의 관점에서는 긴 시간일 거라 추측된다. 그렇지만 언제가 될지 모르는 이 시점에 대한 두려움과 기대감이 교차하기도 한다.

***1** 논문 | Cho, K. et al. (2014). Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation, doi : arXiv:1406.1078. ***2** 논문 | Sutskever, I. et al. (2014). Sequence to Sequence Learning with Neural Networks, doi : arXiv:1409.3215. ***3** 논문 | Bahdanau, D. et al. (2016). Neural Machine Translation by Jointly Learning to Align and Translate, doi : arXiv:1409.0473. ***4** 논문 | Zhou, J. et al. (2016). Deep Recurrent Models with Fast-Forward Connections for Neural Machine Translation, doi : arXiv:1606.04199. ***5** 논문 | Wu, Y. et al. (2016). Google's Neural Machine Translation:Bridging the Gap between Human and Machine Translation, doi : arXiv:1609.08144 ***6** 논문 | Kalchbrenner, N. et al. (2017). Neural Machine Translation in Linear Time, doi : arXiv:1706.03059, 2017 ***7** 논문 | Kaiser, L. et al. (2017). Depthwise Separable Convolutions for Neural Machine Translation, doi : arXiv:1706.03059 ***8** 논문 | Gehring, J. et al. (2017). Convolutional Sequence to Sequence Learning, doi : arXiv:1609.08144, 2017 ***9** 참고 | https://research.googleblog.com/2017/08/transformer-novel-neural-network.html - Google Research Blog ***10** 논문 | Vaswani A. et al. (2017). Attention Is All You Need, doi : arXiv:1706.03762, 2017 ***11** 논문 | Wu, L. et al. (2017). Adversarial Neural Machine Translation, doi : arXiv:1704.06933