Transformer 기반 영문 번역기 시스템

English Translation System based on Transformer

Abstract

지난 몇 년간 Transformer 모델은 자연어 처리 모델로서 다양한 분야에서 발전을 일궈냈다. 특히 기계 번역을 통한 번역 분야의 발전 가능성이 여전히 높은 상태이다. 본 논문은 Transformer 기반 영문 분역기에 대해 표준 베이스라인을 기준으로 데이터 전처리, 데이터 증강, 빔 서치 디코딩을 단계적으로 도입해 희귀어, 복합구문, 장문에서의 취약성을 실무적으로 완화하는 방법론에 대해서 제시한다. 해당 실험을 통해 베이스라인에 비해 에포크 당 손실을 줄이며 히트맵 상에서도 번역 문장의 열 붕괴를 감소됨을 나타낸다. 이는 모델이 학습 과정에서 베이스라인에 비해 안정적이고 문장의 취약성을 효과적으로 극복할 수 있음을 시사한다.

Keywords: Transformer, Preprocessing, Augment, Beam Search

1. 서론

Transformer는 최근 몇 년간 자연어 처리 모델로서 기계 번역, 텍스트 분류, 질의 응답, 음성 인식 등의 다양한 분야에서 응용되고 있다. 과거에 딥러닝(Deep Learning, DL) 알고리즘이 도입된 이후 기계 번역 분야는 일련의 입력 시퀀스를 또 다른 일련의 출력 시퀀스로 변환하는 seq2seq 기법으로 발전해왔다[1]. 특히 Encoder-Decoder 구조에서 Decoder가 각 시점마다 소스 토큰들에 가중치를 두어 필요한 정보를 직접 참조하는 형태의 Attention 알고리즘을 결합하여, 고정 길이 벡터 병목 현상을 완화할 수 있었다. 이러한 발전과정을 통해 2017년 Google에서 “Attention Is All You Need”라는 논문을 통해 Transformer를 제안했다[2]. 해당 알고리즘은 RNN의 재귀적 합성곱 구조를 배제하고 Attention을 통해서만 문맥 의존성을 포착하여 병렬 학습 구조를 제안한다.

Transfomer이 등장한 이후, 자연어 처리 분야는 사전학습(Pre-training)에만 치중되지 않고 미세조정(Fine-tuning)과의 결합을 통한 패러다임으로 변화하였다. 이렇게 병렬 알고리즘을 통한 대규모 학습이 가능해지자 Transformer 모델은 자연스럽게 언어 번역 분야에서 큰 성공을 거둘 수 있었다. 다국어와 부족한 자원들까지 포괄하는 거대한 모델의 등장으로 언어 범위와 표현력이 폭발적으로 늘어났고 FLORES-200 같은 다국어 기계 번역 평가 벤치마크에서도 Transformer 모델을 통해 인류 평가와의 상관도 또한 개선할 수 있었다[3].

하지만 여전히 Transformer에 대한 해결 과제가 남아 있다. 타 알고리즘에 비해 긴 문장에 강건한 것은 사실이지만 여전히 희귀어나 복합구문에서의 과번역, 편향 문제가 여전히 일어난다[4][5]. 또한 학습-추론 불일치에서 비롯된 노출 편향(Exposure bias)과 디코딩의 불안정함에 대한 논의는 계속 지속된다[6][7].

본 논문에서는 Transformer를 통한 번역기 시스템을 통해 희귀어나 복합구문에 대한 문제점들

을 완화하기 위한 다양한 해결 방안을 제안한다. 기본적인 Transformer 모델의 베이스라인을 구축한 뒤, Fine-tuning 과정을 거쳤다. 데이터 전처리(Preprocessing), 데이터 증강(Augment), 빔 서치(Beam Search)를 통해 해당 기법들이 얼마나 Transformer 모델을 구축하는데 기여하는지에 대한 성능 개선점을 제안한다. 본 논문의 유효성을 평가하기 위해서 손실(Loss)을 비교했으며 어텐션 가중치 히트맵(self-Attention Matrix)을 그려 비교 분석을 통해 실험 결과 각 기법들이 성능을 개선할 수 있었다.

2. Transformer 번역기 성능 개선

본 파트에서는 Transformer를 기반으로 만든 영문 번역기의 성능을 개선하고자 한 노력들을 기록한다. 본 연구의 출발점 입력은 서브워드 단위로 분절되어 임베딩이 된 후 Encoder-Decoder 블록으로 처리되며 디코더는 자기회귀 방식으로 토큰을 생성한다[2]. 초기 디코딩은 구현 단순성과 속도를 위해 Greedy Search를 기준선으로 삼는다. Greedy는 각 시점에서 최대 확률의 토큰을 바로 선택하여 빠르지만, 문장 전체의 관점에서는 최적화를 보장하지 못해 긴 문장이나 희귀패턴에서의 품질이 저하될 수 있다[2][4].

베이스라인으로는 초기 모델인 “Attention Is All You Need” 논문 구조 기반의 모델을 시작으로 Positional Encoding, Multi-Head Attention, Position-wise FFN, Residual + LayerNorm으로 구성된 표준 설정을 사용하고, 디코딩은 구현 단순성과 속도를 위해 Greedy Search를 기준선으로 삼았다[2]. 학습 코퍼스 데이터는 공개 병렬 말뭉치인 Korean Parallel Corpora의 Korean-English News v1을 채택하였으며, 뉴스 도메인의 주제·문체 다양성이 한-영 일반 도메인 번역의 강건성 평가에 적합하다는 점에서 베이스라인 데이터로 선정하였다[8]. 해당 베이스라인은 데이터 전처리, 데이터 증강, 빔 서치 등을 통해 개선 폭을 정량하며 비교하기 위한 기본적인 출발선의 기능을 한다.

2.1 Data Preprocessing

데이터 전처리를 통해 베이스라인 데이터에 대해 문장 길이 분포, 길이 비율, 중복률, 이스케이프/제어문자 비중 등을 탐색적으로 분석하여 학습 안정성과 정렬 품질에 악영향을 미치는 이상치를 제거한다. 전통적으로 널리 쓰이는 Moses의 clean-corpus-n.perl 절차를 참조해, 너무 짧거나 긴 문장쌍 및 길이 비율이 과도한 쌍을 필터링하며 공백이 있는 행들을 제거한다[9]. 이러한 길이, 비율 정제는 정렬 오차와 학습 불안정성을 줄여 초기 수렴을 개선하고, 과도한 길이 편향으로 인한 디코딩 취약성을 완화하는 데 유효하다[10][11].

한국어는 교착어로, 조사·어미 결합에 의해 표면형 변이가 많고 희귀 형태가 급증한다. 이에 Mecab(ko-dic)으로 먼저 형태소 단위로 안정적으로 분할한 뒤, 그 결과에 SentencePiece를 적용해 서브워드 사전을 학습하고 적용하는 2단계 전처리를 구성한다. 전처리 순서는 Mecab 형태소를 분리한 뒤 공백 단위 토크나이즈된 문장에 SentencePiece 적용하는 순서로 정리되며, 형태 경계를 우선 보존하면서도 미등록어나 신조어에 대해 서브워드로 일반화할 수 있다[12][13]. 또한 사전 토크나이즈가 없어도 동작하지만, 한국어에서는 형태 경계를 준거로 삼아 서브워드가 보다 일관되게 학습되는 이점이 있다.

2.2 Data Augment

Transformer의 병렬 데이터 학습의 한계를 보안하기 위해 역번역을 적용한다. 먼저 타깃에서 소스의 역방향 소규모 모델을 학습한 후, 대규모 타깃 단독 말뭉치에 대해 번역을 수행하여 합성 소스와 실제 타깃 쌍을 만든다. 이를 원 병렬 데이터와 혼합해 학습하면 도메인 적응력이 향상되는 것을 확인할 수 있다[14].

본 연구의 파이프라인으로는 먼저 원문에서 중복, 길이, 문자 등의 이상치들을 제거한 뒤 한국어는 Mecab/Okt로 형태소 단위로 분리하여 문자열을 생성하였다. 그리고 SentencePiece 토크나이저는 원본 코퍼스로 먼저 학습해 도메인 전반의 분절 단위 규칙을 안정화했으며 일부 문장쌍을 샘플링해 의미를 해치지 않는 선에서 변형하여 원본의 약 1.5배가 되도록 확장했다. 마지막으로 원본 데이터에 증강을 추가하여 세트를 셔플해 학습했으며 검증/테스트 데이터는 왜곡을 방지하기 위해 증강에서 배제했다. 이 구성은 형태 경계가 드러난 표면에서 다양성을 넓히면서 SentencePiece는 원본 분포로 학습해 과도한 분절 왜곡을 피하려는 목적을 가진다.

하지만 동시에 몇 가지 제약사항 또한 가지게 된다. SentencePiece를 원본으로 학습하면서 한국어는 형태소 분해 문자열을 병행해서 사용하므로 분절 규칙-표면 분포 간의 미세한 불일치가 생길 수 있다[9]. 또한 동의어 치환, 토큰 스왑 등의 얕은 변형도 문맥에 따라 미묘하게 바꾸거나 중복 표본을 늘려 표면 과적합을 유발할 수 있다. 그리고 데이터 증강을 할 경우에는 메모리가 크게 급증하여 Out of Memory 오류를 자주 유발하기에 실험단계에서의 데이터 메모리 관리가 필수적이다.

2.3 Beam Search

Greedy Search는 각 스텝에서 조건부 확률이 최대인 단어를 하나씩 선택하여 빠르지만, 문장 전체 확률 관점에서 특정 문장에서는 좋을 수는 있지만 전체 영역에서는 최적의 문장이 아닐 가능성이 높다. 이에 반해 Beam Search는 각 스텝마다 상위 k개의 후보 경로를 유지한 채로 전개하여 전역적으로 더 우수한 후보를 탐색한다. 이로써 문장 수준 우도와 평가 지표의 균형이 개선되는 것이 다수 시스템에서 관찰되며, 기계 번역의 표준 디코딩으로 자리잡았다 [4][15]. 또한 순수 로그우도 합계는 긴 문장에 불리한 편향을 유발할 가능성이 있어 길이 정규화가 널리 쓰이며, Attention 누적치를 활용해 소스 토큰의 미번역을 억제하는 커버리지 패널티가 함께 적용되기도 한다. GNMT는 실제 프로덕션 환경에서 길이 정규화와 커버리지 패널티를 결합해 품질을 높였음을 보고한다[4].

본 연구의 빔 서치 구현은 빔 크기 k를 탐색하고 길이 패널티의 계수를 스윕한 이후 커버리지 항의 유무 비교를 포함하여 베이스라인 Greedy 대비 번역 적합도·유창성 향상을 목표로 한다[16]. 실제 연구단계에선 빔 크기를 무작정 키우면 계산 비용이 증가하고, 오히려 길이 편향과 반복적, 상투적 출력이 심화됨을 확인할 수 있다. 최근 연구는 빔 서치가 MAP 목표만으로는 설명되지 않는 텍스트 특성을 구현하며, 그로 인해 특정 조건에서 빔 확장이 항상 품질 개선으로 이어지지 않는다는 점을 지적한다[17]. 따라서 실제 시스템에서는 빔 크기·길이에 패널티를 부여하거나 조기 종료(Early Stop) 기준을 개발셋에서 체계적으로 선택하고, 필요에 따라 커버리지, 다양성 제약이나 사전구속(lexical constraints)을 병행하는 보수적 튜닝이 권장된다[16][17].

3. 실험 결과

4장에서는 본 논문에서 번역기 성능을 개선하기 위해 제안하는 방법론과 베이스라인 간의 다양한 평가 지표와 성능 비교를 통해 해당 기술들을 사용했을 때의 이점과 한계점에 대해서 서술한다.

4.1 에포크 당 손실(Loss) 비교

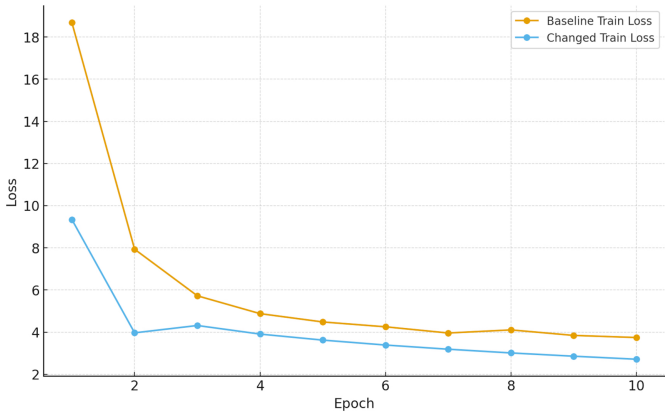
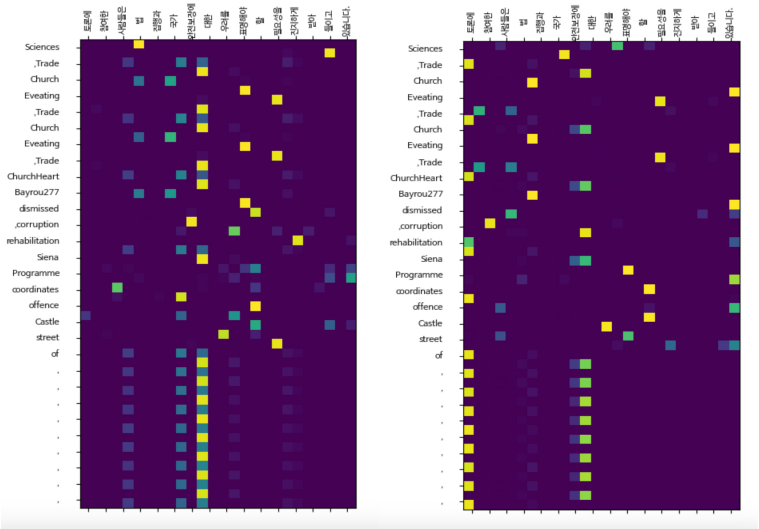


Fig. 1. Loss comparison graph per epoch

Fig. 1. 그래프를 통해 Loss는 베이스 모델에 비해서 학습 과정 동안 손실값이 크게 증폭되거나 급격한 변화 없이 안정적으로 감소하는 것을 확인할 수 있다. 이는 학습이 안정적으로 진행되고 있음을 의미하며, 과적합의 위험을 감소시키는 것을 나타낸다. 또한 개선한 모델에서는 에포크가 2일 때 먼저 수렴하는 것을 볼 수 있다. 이는 학습이 원활하게 잘 진행되고 있음을 나타낸다.

4.2 어텐션 가중치 히트맵(self-Attention Matrix)



(a) (b)

Fig. 2. self-Attention Matrix (a) Baseline, (b) Changed

Fig. 2. 히트맵을 통해 (a)의 베이스라인 모델에 비해 (b) 개선된 모델이 열 붕괴가 소폭 감소한 것을 확인할 수 있다. (a)를 확인하면 특정 열에 노란 세로 띠가 길게 나타나 다수 타깃 토큰이 같은 소스 토큰만 보는 현상을 보인다. 하지만 (b)를 확인하면 밝은 점들이 여러 열로 분산돼 있어 단일 열에 매달리는 정도가 (a)에 비해 낮다. 또한 문장 부호와 기능어에 대해서 (a)는 마지막 토큰, 쉼표 쪽에 과하게 밝은 반면 2번째는 핵심 내용어인 국가, 우려, 표명 등의 주변에도 피크가 살아 있는 모습을 확인할 수 있다. 이는 베이스라인 모델에 비해 개선된 모델이 히트맵 상에서 소폭 상승된 결과를 나타낼 수 있음을 시사한다. 다만 두 히트맵 모두 잡음이 있고, 완벽한 정렬의 형태는 아니므로 더욱 발전된 모델을 위한 활발한 연구가 필요할 것이다.

4. 결론

본 논문은 표준 Transformer 영문 번역기 베이스라인 위에서 전처리, 데이터 증강, 빔 서치 등을 결합한 개선 모델을 통해 모델의 안정성, 완전성, 유창성을 개선하고자 실용적 경로를 제시했다. 이 접근은 데이터와 모델, 탐색을 종합적으로 다루며 대규모 사전학습의 유무와 무관하게 도메인 맞춤 번역기를 고도화를 기대할 수 있다. 하지만 체계적인 실험이 필요하다. 베이스라인에서 시작해 Fine-Tuning, 데이터 정제, 아키텍처 변경 등을 거치며 점진적으로 성능을 개선할 수 있었지만, 모든 개선안을 다 집대성한 최종 모델이 베이스라인에 비해 성능이 크게 영향력있게 급증하지는 못했다. 결국 성공적인 딥러닝 기술은 데이터를 끊임없이 알고리즘을 분석하고 데이터를 정제하며 결과를 분석해야 한다. 본 논문은 이러한 점에서 해당 Transformer를 분석하는 과정을 통해 좋은 모델을 만드는 “방법론”에 대한 방향성을 제시할 수 있는 연구과제가 되기를 기대한다.

REFERENCE

[1] Sutskever, I., Vinyals, O., and Le, Q. V., "Sequence to Sequence Learning with Neural Networks," *Advances in Neural Information Processing Systems*, Vol. 2014, pp. 3104-3112, 2014.

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I., "Attention Is All You Need," *Advances in Neural Information Processing Systems*, Vol. 2017, pp. 5998-6008, 2017.

[3] Goyal, N., Gao, C., Chaudhary, V., et al., "The FLORES-200 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation," Transactions of the Association for Computational Linguistics, Vol. 10, pp. 522-538, 2022.

[4] Wu, Y., Schuster, M., Chen, Z., et al., "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," arXiv preprint, Vol. 2016, 2016.

[5] Tu, Z., Lu, Z., Liu, Y., Liu, X., and Li, H., "Modeling Coverage for Neural Machine Translation," Annual Meeting of the Association for Computational Linguistics, Vol. 2016, 2016.

[6] Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N., "Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks," Advances in Neural Information Processing Systems, Vol. 2015, 2015.

[7] Ranzato, M., Chopra, S., Auli, M., and Zaremba, W., "Sequence Level Training with Recu rrent Neural Networks," International Conference on Learning Representations, Vol. 2016, 2016.

[8] Park, J., "Korean Parallel Corpora: Korean-English News v1," *GitHub Repository*, Vol. 2 019, 2019.

[9] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., et al., "Mo ses: Open Source Toolkit for Statistical Machine Translation," *Proceedings of ACL (Demo an d Poster Sessions)*, Vol. 2007, pp. 177-180, 2007.

[10] Moses SMT, "Support Tools and clean-corpus-n.perl," *Project Documentation*, Vol. 201 6, 2016.

[11] NVIDIA NeMo Team, "Length/Ratio Filtering for Bitext," *NeMo Curator & MT Docs*, Vol. 2021, 2021.

[12] Kudo, T., and Richardson, J., "SentencePiece: A Simple and Language Independent Sub word Tokenizer and Detokenizer for Neural Text Processing," *Proceedings of EMNLP*, Vol. 2 018, 2018.

[13] Kudo, T., "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates," *Proceedings of ACL*, Vol. 2018, 2018.

[14] Edunov, S., Ott, M., Auli, M., and Grangier, D., "Understanding Back-Translation at Sc ale," *arXiv preprint*, arXiv:1808.09381, Vol. 2018, 2018.

[15] Meister, C., Cotterell, R., and Vieira, T., "If Beam Search is the Answer, What Was the Question?," *Proceedings of EMNLP*, Vol. 2020, pp. 2173-2185, 2020.

[16] Yang, Y., et al., "A Systematic Analysis of Decoding Objectives (incl. Length/Coverage Penalties)," *arXiv preprint*, arXiv:1808.09582, Vol. 2018, 2018.

[17] Meister, C., Cotterell, R., and Vieira, T., "If Beam Search is the Answer, What Was the Question?," *Proceedings of EMNLP*, Vol. 2020, pp. 2173-2185, 2020.