BERT를 이용한 지도학습 기반 문장 임베딩 모델

최기현†○, 김시형†, 김학수†, 김관우‡, 안재영‡, 최두진‡
강원대학교†, 삼성중공업‡
pluto32@kangwon.ac.kr, sureear@kangwon.ac.kr, nlpdrkim@kangwon.ac.kr,
kwanwoo.kim@samsung.com, encage.an@samsung.com, dj01.choi@samsung.com

Supervised Learning for Sentence Embedding Model using BERT

Gihyeon Choi†○, Sihyung Kim†, Harksoo Kim†, Kwanwoo Kim‡, Jaeyoung An‡, Doojin Choi‡
Kangwon National University Computer and Communication Engineering†
Samsung Heavy Industries‡

요 약

문장 임베딩은 문장의 의미를 잘 표현 할 수 있도록 해당 문장을 벡터화 하는 작업을 말한다. 문장 단위 입력을 사용하는 자연언어처리 작업에서 문장 임베딩은 매우 중요한 부분을 차지한다. 두 문장 사이의 의미관계를 추론하는 자연어 추론 작업을 통하여 학습한 문장 임베딩 모델이 기존의 비지도 학습 기반 문장 임베딩 모델 보다 높은 성능을 보이고 있다. 따라서 본 논문에서는 문장 임베딩 성능을 높이기 위하여 사전 학습된 BERT 모델을 이용한 문장 임베딩 기반 자연어 추론 모델을 제안한다. 문장 임베딩에 대한 성능 척도로 자연어 추론 성능을 사용하였으며 SNLI(Standford Natural Language Inference) 말뭉치를 사용하여 실험한 결과 제안 모델은 0.8603의 정확도를 보였다.

주제어: 문장 임베딩, BERT, 자연어 추론, 문장 인코더

1. 서론

문장 임베딩(Sentence Embedding)이란 문장이 가지고 있는 의미를 효과적으로 표현할 수 있도록 벡터화 하는 작업(Task)을 말한다. 문장 임베딩은 문서 요약이나 인공지능 챗봇(Chatbot)과 같은 문장 단위 입력을 사용하는 자연언어처리 작업의 성능을 향상시키기 위하여 필수적으로 연구되어야 한다.

최근 여러 자연언어처리 분야에서 state-of-the-art 성능을 달성한 BERT(Bidirectional Encoder Representations from Transformers)[1] 공개 이후, 이를 활용한 여러 자연언어처리 모델이 개발되고 있다[2-3]. 본 논문에서는 문장 임베딩의 성능 향상을 위하여 BERT 기반 문장 임베딩 모델을 제안한다.

2. 관련 연구

문장 임베딩 모델은 비지도 학습(Unsupervised Learning) 접근 방식을 적용한 모델[4,7]과 지도 학습(Supervised Learning) 접근 방식을 적용한 모델[8-9] 등 다양하게 연구되었다. [4]는 단어 임베딩 방법 중 하나인 Skip-gram 알고리즘[5]을 문장 단위로 확장하여 Seq2Seq 모델(Sequence to Sequence Model)[6]을 기반으로 주어진 문장 주변의 문장을 예측하는 방식으로 문장 임베딩을 수행하는 Skip-thought 모델을 제안하였다. [7]은 문장에서 중요도가 낮은 단어의 정보를 감소시키고 공통 구성 성분을 제거하는 방식으로 문장을 임베딩하는 SIF(Smoother Inverse Frequency) 모델을 제안하였다. [8]에서 제안한 InferSent 모델은 두 문장 사이의 의미적 관계를 분류하는 자연어 추론 작업을 활용하여 문장 임베딩을 수행하였다. [8]의 연구는 자연어 추론 작업을 이용한 지도 학습 접근 방식이 비지도 학습 접근 방식 보다 우수함을 증명하였다. [9]는 [8]의 인코더 구조를 확장하여 Hierarchical BiLSTM(Bidirectional Long Short Term Memory)[10] 인코더로 구성된 문장 임베딩 모델을 제안하였다. 본 논문은 [8]과 같이 자연어 추론 작업을 통해 학습하는 문장 임베딩 모델을 제안한다.

자연어 추론은 두 문장이 있을 때 해당 두 문장의 의미가 유사한지, 모순되는지, 서로 관련이 없는지를 분류하는 작업이다. 자연어 추론 작업에는 두 가지 주요 접근 방식이 있다. 첫 번째 접근 방식은 두 문장이 있을 때 문장 인코더로 각각 문장 벡터를 독립적으로 생성한다. 그리고 생성된 두 문장 벡터를 분류 디코더의 입력으로 사용하여 두 문장 사이의 의미적 관계를 분류하는 방식이다. 두 번째 접근 방식은 두 문장을 개별적으로 인코딩하지 않고 동시에 인코더의 입력으로 사용한다. 인코더에서 문장 사이의 관계 정보를 추상화하여 이를 바탕으로 분류 디코더가 두 문장의 의미적 관계를 분류하는 방식이다.

[8]의 연구에 따르면 첫 번째 접근 방식을 적용하여 학습 된 인코더 모델이 기존의 비지도 학습 기반 문장 임베딩 모델보다 높은 성능을 보였다. 따라서 본 논문에서는 [8]의 연구와 같이 자연어 추론 작업을 통하여 문장 임베딩 모델을 학습한다. 문장 임베딩 성능을 향상시키기 위하여 BERT 기반 문장 인코더로 구성된 모델을 제안한다.

BERT는 공개 당시, 11개의 자연언어처리 작업에서 가장 높은 성능을 보인 언어 모델이다. BERT는 대용량의 일반 데이터로 학습한 언어 정보를 토대로 특정 작업에 대한 정답이 부착된 데이터를 추가 학습하여 성능을 높

인 모델이다. 여러 개의 트랜스포머 계층(Transformer Layer)[11]로 구성되어 있으며 모델의 크기에 따라서 계층의 개수가 나뉜다(BERT_base:12개, BERT_large:24개).

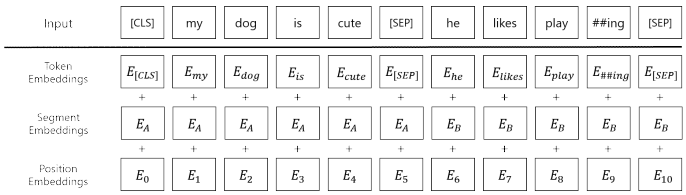

그림 1 BERT 입력 예시

모델의 입력은 그림 1과 같으며 토큰 임베딩(Token Embedding), 세그먼트 임베딩(Segment Embedding), 포지션 임베딩(Position Embedding)의 합을 사용한다. 토큰 임베딩은 각 단어의 의미를 나타낸다. 세그먼트 임베딩은 여러 문장을 입력으로 사용할 때 각 문장을 구분해주기 위한 임베딩이며 포지션 임베딩은 각 token의 위치정보를 나타내기 위한 임베딩이다. 모든 문장의 입력에 [CLS] 토큰을 추가하는데 이 토큰에 트랜스포머 계층을 통해 추상화된 입력 문장의 의미가 집중된다. 최종적으로 BERT는 마지막 계층의 [CLS] 토큰을 사용하여 분류 문제를 해결한다.

3. 제안 모델

3.1 문장 임베딩 기반 자연어 추론 모델 구조도

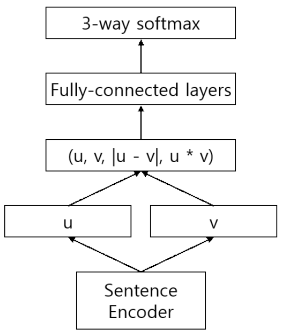

그림 2 자연어 추론 모델 구조도

자연어 추론 작업을 수행하기 위한 문장 임베딩 기반 모델의 구조도는 그림 2와 같다. 동일한 인코더를 사용하며 개별적으로 문장 1과 문장 2를 인코딩하여 문장 벡터를 구한다. 그림에서 u와 v는 각각 문장 인코더를 통하여 인코딩된 문장 벡터를 의미한다. 두 문장의 관계 정보를 추출하기 위하여 u와 v의 요소별 차의 절대값(|u-v|), 요소별 곱(u*v)과 각 문장 벡터(u, v)를 연결(Concatenate)한 벡터를 3개의 완전 연결 계층(Fully-connected Layer)과 소프트맥스 계층(Softmax Layer)로 구성된 분류 네트워크의 입력으로 사용한다. 분류 네트워크를 통하여 두 문장의 의미가 유사한지, 서로 모순되는지, 관계가 없는지를 분류한다. 위의 구조는 [8]에서 제안한 문장 임베딩 기반 자연어 추론 모델의 구조이며 본 논문에서는 문장을 인코딩 하기 위한 인코더로 BERT 기반 문장 인코더를 사용한다.

3.2 BERT 기반 문장 인코더

BERT_base 모델을 이용하여 실험하였으며 BERT 모델로부터 입력 문장에 대한 문장 벡터로서의 역할을 가장 잘 수행할 수 있는 값을 추출하고 활용하기 위하여 여러 구조의 BERT 기반 문장 인코더를 사용하여 실험하였다.

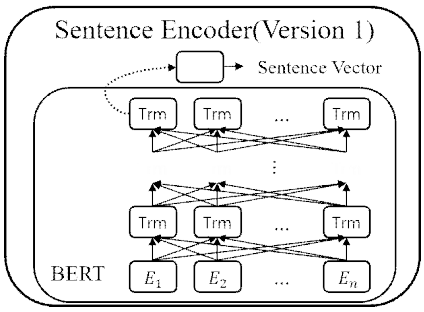

그림 3 BERT 기반 문장 인코더 Version 1

첫 번째 문장 인코더의 구조는 그림 3과 같다. 기존의 BERT 출력과 동일하게 마지막 트랜스포머 계층의 CLS 벡터를 추출하여 이를 문장 벡터로 사용한다.

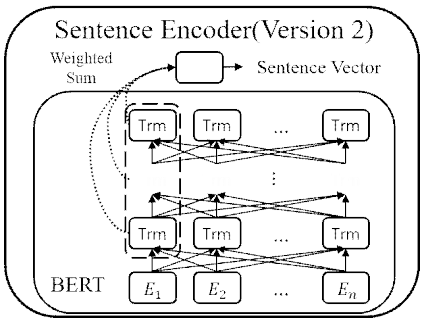

그림 4 BERT 기반 문장 인코더 Version 2

두 번째 문장 인코더의 구조는 그림 4와 같다. 각 트랜스포머 계층에서 추출한 CLS 벡터의 가중합(Weighted Sum)을 문장 벡터로 사용하였다.

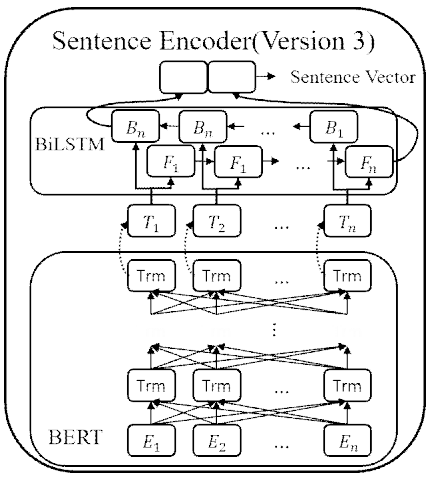

그림 5 BERT 기반 문장 인코더 Version 3

세 번째 문장 인코더의 구조는 그림 5와 같다. 마지막 트랜스포머 계층의 CLS 토큰의 대한 출력만을 사용하는 것이 아니라 전체 입력 토큰에 대한 출력을 사용한다. T_1부터 T_n까지의 출력 값들을 양방향 LSTM 계층의 입력으로 하여 출력으로 나온 양방향 LSTM의 마지막 은닉 상태(Hidden state) 값을 연결한 벡터를 문장 벡터로 사용한다.

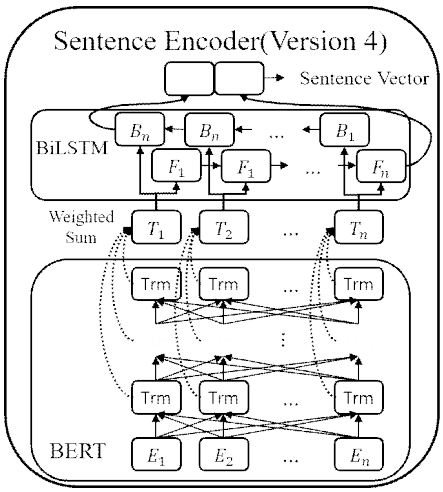

그림 6 BERT 기반 문장 인코더 Version 4

네 번째 문장 인코더의 구조는 그림 6과 같다. 각 트랜스포머 계층에서 추출한 전체 입력 토큰 값들에서 토큰 별 가중합을 통하여 T_1부터 T_n를 구한다. Version3 모델과 같이 T_1부터 T_n까지의 출력 값들을 양방향 LSTM 계층의 입력으로 하여 출력으로 마지막 은닉 상태값을 연결한 벡터를 문장 벡터로 사용한다.

4. 실험

4.1 데이터 셋

표 1 자연어 추론에 사용된 말뭉치

Train	Valid	Test
549,367	9,842	9,824

본 논문에서는 자연어 추론 데이터를 이용한 문장 임베딩 실험을 위해 SNLI(Stanford Natural Language Inference)[12] 말뭉치를 사용하였다. SNLI 말뭉치의 통계는 표 1과 같다. SNLI 말뭉치는 두 문장과 해당 두 문장의 의미 관계로 구성되어 있다. 두 문장의 의미 관계는 두 문장의 의미가 유사할 경우 “entailment”, 모순될 경우 “contradiction”, 서로 관련이 없을 경우 “neutral”로 표현한다. 문장 임베딩에 대한 정량적 성능 측정이 어렵기 때문에 본 논문에서는 두 문장 간의 의미 관계를 분류하는 성능에 대한 정확도(Accuracy)를 문장 임베딩의 성능으로 보았다.

4.2 실험 결과

표 2 BERT 기반 문장 인코더 별 성능

모델	Accuracy
Version 1	0.8436
Version 2	0.8548
Version 3	0.8556
Version 4	0.8603

표 2는 BERT 기반 문장 인코더 별 성능 비교를 보여준다. Version2 모델이 Version1 모델보다 1.12%p 높은 것을 확인 할 수 있다. 이는 BERT 모델의 CLS 벡터만을 사용한 것 보다 각 계층의 CLS 벡터를 모두 활용한 문장 벡터가 입력 문장을 좀 더 잘 설명 할 수 있는 값임을 보여준다. Version3 모델과 Version4 모델은 CLS 토큰에 대한 출력만을 사용하지 않고 전체 입력 토큰에 대한 출력을 모두 활용하는 모델이다. Version3 모델은 Version1 모델과 비교하여 1.2%p 증가하였고 Version4 모델은 Version2 모델과 비교하여 0.55%p 증가하였다. 이는 전체 입력 토큰에 대한 출력을 모두 사용하는 것이 CLS 벡터만을 사용하는 것 보다 더 나은 문장 벡터를 생성할 수 있음을 보여준다.

5. 결론

본 논문은 BERT 기반 문장 인코더를 사용하여 문장 벡터를 생성하는 문장 임베딩 기반 자연어 추론 작업을 통해 문장을 벡터화 하는 모델을 제안하였다. 실험결과 BERT 모델에서 단일 계층의 출력만을 사용하는 것보다 모든 계층의 출력을 활용하는 것이 더 나은 성능을 보였고, CLS 토큰에 대한 출력만 사용하는 것보다 전체 입력 토큰에 대한 출력 모두를 활용하는 것이 더 나은 성능을 보였다. 이는 기존 BERT 모델의 CLS 토큰에 대한 마지막 트랜스포머 계층의 출력 값만 활용하는 것보다 입력 토큰에 대한 모든 출력과 각 트랜스포머 계층별 출력을 적절히 활용하는 것이 더 나은 문장 벡터를 생성함을 보여준다. 향후 연구로 모델이 생성한 문장 벡터를 여러 자연어처리 작업에 적용하여 문장 벡터에 대한 범용성을 정량화 할 수 있는 SentEval[13] 도구를 사용하여 기존의 문장 임베딩 연구들과 정량적 성능 비교를 수행할 예정이다.

감사의글

본 연구는 삼성중공업 산학연구용역 과제의 지원을 받아 수행되었음

참고문헌

[1] J. DEVLIN, M. Chang, K. Lee and K. Toutanova. “Bert: Pre-training of deep bidirectional transformers for language understanding”, *arXiv preprint arXiv:1810.04805*, 2018.

[2] 황현선, 이창기, “BERT 기반 한국어 개방형 정보 추출”, *한국정보과학회 학술발표논문집*, pp.521-523, 2019.

[3] 박천음, 김기훈, 이창기, 임준호, 류지희, 김현기, “ BERT 기반 Deep Biaffine 을 이용한 한국어 상호 참조해결”, *한국정보과학회 학술발표논문집*, pp.488-490, 2019.

[4] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, “Skip-thought vectors”, *Advances in Neural Information Processing Systems (NIPS)*, pp.3294-3302. 2015.

[5] T. Mikolov, K. Chen, G. Corrado and J. Dean, “Efficient estimation of word representations in vector space”, *arXiv preprint arXiv:1301.3781*, 2013.

[6] I. Sutskever, O. Vinyals and Q. V. Le, “Sequence to sequence learning with neural networks”. *Advances in Neural Information Processing Systems (NIPS),* pp.3104-3112. 2014.

[7] S. Arora, Y. Liang and T. Ma, “A simple but tough-to-beat baseline for sentence embeddings”, *Advances in International Conference on Learning Representations*, 2017.

[8] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, “Supervised learning of universal sentence representations from natural language inference data”, *arXiv preprint arXiv:1705.02364*, 2017.

[9] A. Talman, A. Yli-Jyra and J. Tiedemann, “Sentence embeddings in NLI with iterative refinement encoders”, *arXiv preprint arXiv:1808.08762*, 2018.

[10] S. Hochreiter, J. Schmidhuber, “Long short-term memory”, *Neural computation*, 9(8), pp.1735-1780, 1997.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, “Attention is all you need”, *Advances in Neural Information Processing Systems (NIPS)*, pp.5998-6008, 2017.

[12] S. R. Bowman, G. Angeli, C. Pott and C. D. Manning, “A large annotated corpus for learning natural language inference”, *arXiv preprint arXiv:1508.05326*, 2015.

[13] A. Conneau, D. Kiela, “Senteval: An evaluation toolkit for universal sentence representations”, *arXiv preprint arXiv:1803.05449*, 2018.