

사건 관계 추출을 위한 사전 학습 임베딩 비교

양승무, 이미라, 정찬희, 정혜동*
 한국전자기술연구원
daniel0826@keti.re.kr

Comparing of pre-trained Embedding for Event Extraction

Seung-Moo Yang, Mira Lee, Chan-Hee Jeong, Hye-Dong Jung*
 Human IT Convergence Research Center
 Korea Electronics Technology Institute (KETI)

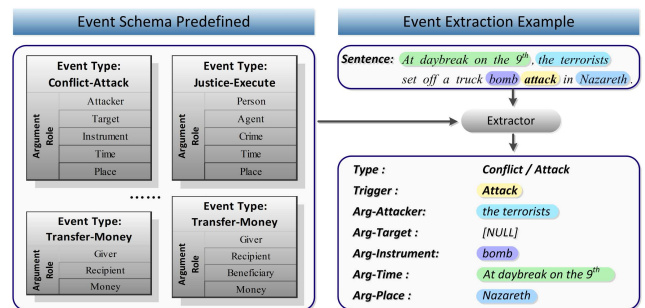
요 약

사건 관계 추출 태스크는 구조화되지 않은 텍스트 데이터에서 사건의 구조화된 표현을 얻는 것이다. 하나의 문장에서도 많은 정보를 얻을 수 있는 중요한 태스크임에도 불구하고, 다양한 사전 학습 모델을 적용한 연구는 아직 활발하게 연구되지 않고 있다. 따라서 본 연구에서 사전 학습된 모델의 임베딩 기법 중 BERT, RoBERTa, SpanBERT에 각각 base, large 아키텍처를 적용하여 실험하였다. 사건을 식별하기 위한 trigger와 해당 trigger의 세부 argument를 식별하기 위한 분류기를 상위레이어로 각각 설계하였고, 다양한 배치 크기를 적용하여 실험하였다. 성능평가는 trigger/argument 각각 F1 score를 적용하였고, 결과는 RoBERTa large 모델에서 좋은 성능을 보인 것을 확인하였다.

1. 서론

사건 관계 추출 태스크의 목적은 주어진 문장에서 사전에 정의된 사건과 사건에 속한 세부적인 사항을 감지하고 식별하는 것이다.[1] 그림 1처럼 사건이란 특정 시간과 장소에서 한 명 이상의 사람이 관련된 특정한 사건이며, 종종 상태의 변화로 묘사될 수 있다. 이러한 사건과 세부적인 사항을 식별하는 것은 뉴스 기사, 소셜 미디어 게시물 등의 수많은 텍스트 출처에서 실제 사건의 육하원칙을 포함한 질문에 대답하는 것을 돕기 위해서이다.

사건 관계 추출 태스크를 해결하기 위해서 합성곱 신경망을 이용한 방법[2], 순환 신경망을 이용한 방법[3]들이 있었다. 나아가 문장 단위의 의미를 파악하기 위한 문장 임베딩 방법도 등장하였다. 하지만 다양한 사전 학습 방법을 통해 얻은 임베딩들의 성능은 비교된 적이 없다. 따라서 본 연구에서는 여러 문장 단위의 임베딩을 사용한 실험을 통해서 비교하고 분석한다.



(그림 1) ACE 2005 데이터 세트 예시[1]

2. 실험 방법

실험에 사용한 사전 학습된 임베딩은 BERT[4], RoBERTa[5], SpanBERT[6]이다.

BERT는 트랜스포머[7]의 인코더 구조에 대규모의 텍스트 데이터를 이용하여 학습한다. BERT는 두 가지의 태스크를 차례로 학습한다. 첫 번째 Masked Language Modeling(MLM)으로 임의의 토큰을 [MASK] 토큰으로 마스킹하여 실제 단어를 맞추는 태스크이다. 두 번째는 Next Sentence Prediction(NSP)으로 입력으로 들어오는 두 문장을 한 쌍으로 전/후 관계가 맞는지 판단하는 이진 분류 태스크이다.

RoBERTa는 기존 BERT 모델의 아키텍처와 동일하지만 배치 크기, 데이터의 양, 학습 시간을 늘려 학습시킨 모델이다.

SpanBERT는 토큰을 개별적으로 랜덤하게 마스킹하는 BERT와 달리 연속적으로 마스킹하여 학습하는 방식으로, 질의응답 태스크 같이 문장 간의 관계를 파악하는 태스크에서 좋은 성능을 보인다.

<표 1> 데이터 개요

사건 타입(9개)	사건 세부 분류(34개)	사건 내 역할 정보 (28개)
1	Life Be-Born, Marry, Divorce, Injure, Die	Place, Crime, Prosecutor, Sentence, Org, Seller, Entity, Agent, Recipient, Target, Defendant, Plaintiff, Origin, Artifact, Giver, Position, Instrument, Money, Destination, Buyer, Beneficiary, Attacker, Adjudicator, Person, Victim, Price, Vehicle, Time
2	Movement Transport	
3	Contact Meet, Phone-write	
4	Conflict Attach, Demonstrate	
5	Business Merge-org, Declare-bankruptcy, Start-org, End-org	
6	Transaction Transfer-money, Transfer-ownership	
7	Personnel Elect, Start-position, End-position, Nominate	
8	Justice Arrest-jail, Execute, Pardon, Release-parole, Fine, Convict, Charge-indict, Trial-hearing, Acquite, Sentence, Sue, Extradite, appeal	

실험에는 관계 추출, 개체명 인식, 사건 관계 추출 태스크를 위한 벤치마크 사용되는 ACE 2005 데이터 세트를 사용하였다.[8] 사전에 정의된 사건의 타입은 <표 1>과 같다. 사건의 타입은 총 34개, 사건 내 역할 정보는 28개이다.

(그림 2)에서 확인할 수 있듯이 ACE 2005 데이터 세트에는 사건 관계 추출 태스크를 위한 trigger와 argument 정보가 포함되어 있다. 사전에 정의된 34개의 Trigger는 '공격 상황' 등 특정 사건을 가장 명확하게 표현하는 단어를 의미하며, 사전에 정의된 28개의 argument role은 사건에 해당하는 세부 속성을 의미한다. 이러한 정보들을 이용하여 사건 관계 추출 태스크를 해결한다.

<표 2> base 모델과 large 모델의 설정 값

모델 크기	num of layers	hidden dim	Multi-head attention
base	12	768	12
large	24	1024	16

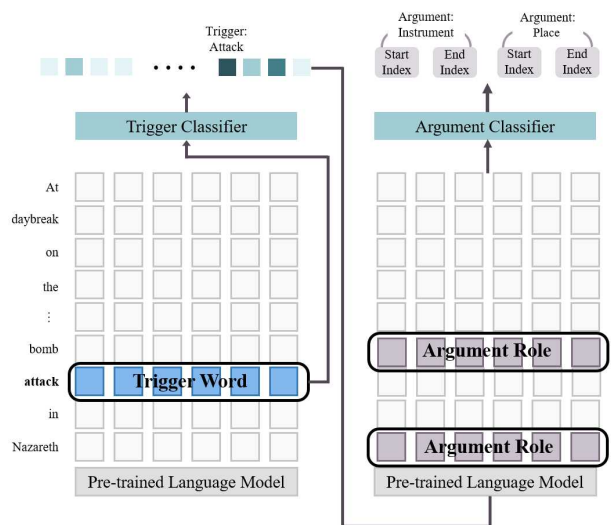
```

"golden-event-mentions": [
  {
    "trigger": {
      "text": "violence",
      "start": 19,
      "end": 20
    },
    "arguments": [
      {
        "role": "Place",
        "entity-type": "FAC:Path",
        "text": "street",
        "start": 18,
        "end": 19
      },
      {
        "role": "Place",
        "entity-type": "GPE:Nation",
        "text": "the country",
        "start": 21,
        "end": 23
      }
    ]
  },
  {
    "event_type": "Conflict:Attack"
  }
]
    
```

(그림 2) ACE2005 구조

사전 학습 모델과 토큰나이저는 Huggingface[9]의 라이브러리를 사용하였으며, base 모델과 large 모델을 각각 적용하였고 설정 값은 <표 2>와 같다.

전체 모델의 구조는 그림 3과 같으며, 사전 학습된 임베딩에 문장의 토큰들과 argument를 입력한다. 임베딩 레이어를 거친 뒤, 상위 레이어에 trigger를 식별하는 분류기와 argument를 식별하는 분류기를 추가해서 파인튜닝을 차례로 진행한다. 즉, trigger를 먼저 식별한 후 식별된 해당 trigger의 하위 argument role를 찾는 방식으로 모델이 구성된다. 따라서 이 두 개의 하위 태스크는 상호 의존적이며, 하나의 단어에서 trigger의 종류와 시작과 끝의 위치를 분류하는 전자의 태스크보다 문장 혹은 구에서 argument role의 시작 지점과 끝 지점을 맞춰야 하는 후자의 태스크의 난이도가 높다.



(그림 3) 사건관계추출 모델

실험 환경은 모든 모델에 동일하게 적용하였다. GPU는 Nvidia DGX의 A100 1개를 사용하였다. 배치 크기는 128, optimizer는 AdamW[10]를 사용하였고 총 100 epoch 학습시켰다. 모델의 성능 검증을 위한 평가지표는 F1 score를 사용하였다.

<표 3> 실험 결과

사전 학습된 모델	Results	
	Trigger_f1	Argument_f1
BERT base	0.67	0.42
BERT large	0.70	0.44
RoBERTa base	0.67	0.42
RoBERTa large	0.72	0.47
SpanBERT base	0.67	0.40
SpanBERT large	0.68	0.41

실험 결과는 <표 3>에서 확인할 수 있으며, RoBERTa large의 사전 학습된 임베딩을 사용한 결과가 가장 좋았다.

3. 결론

다른 자연어처리 태스크들과 마찬가지로 사전 학습에 사용된 리소스가 크고 아키텍처가 클수록 좋은 성능을 보인 것을 확인하였다.

추후 연구로는 전결합층 레이어만 사용한 기본적인 파인튜닝 뿐 아니라, 그래프 모델 등의 레이어를 적용하여 실험을 진행할 예정이다.

ACKNOWLEDGEMENTS

이 연구는 2021년도 산업통상자원부 및 산업기술 평가관리원(KEIT) 연구비 지원에 의한 연구임(20016308, 산업지능화 지원을 위한 인공지능 성능지표 개발 및 신뢰성 기반조성)

참고문헌

[1] Xiang, W., &Wang, B. (2019). A survey of event extraction from text. *IEEE Access*, 7, 173111-173137.

[2] Nguyen, T. H., &Grishman, R. (2015, July). Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume*

2: Short Papers) (pp. 365-371).

[3] Nguyen, T. H., Cho, K., &Grishman, R. (2016, June). Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 300-309).

[4] Devlin, J., Chang, M. W., Lee, K., &Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[5] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... &Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[6] Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., &Levy, O. (2020). Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8, 64-77.

[7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... &Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

[8] Doddington, G. R., Mitchell, A., Przybocki, M. A., Ramshaw, L. A., Strassel, S. M., &Weischedel, R. M. (2004, May). The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec (Vol. 2, No. 1, pp. 837-840)*.

[9] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... &Rush, A. M. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

[10] Loshchilov, I., &Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.