ISSN 2288-4866 (Print)
ISSN 2288-4882 (Online)
http://www.jiisonline.org
J Intell Inform Syst 2020 December: 26(4): 1~25 http://dx.doi.org/10.13088/jiis.2020.26.4.001

BERT를 활용한 속성기반 감성분석: 속성카테고리 감성분류 모델 개발*

박현정
고려대학교 Human-inspired 복합지능연구센터
(hyunjpark@korea.ac.kr)

신경식
이화여자대학교 경영대학
(ksshin@ewha.ac.kr)

대규모 텍스트에서 관심 대상이 가지고 있는 속성들에 대한 감성을 세부적으로 분석하는 속성기반 감성분석(Aspect-Based Sentiment Analysis)은 상당한 비즈니스 가치를 제공한다. 특히, 텍스트에 속성어가 존재하는 명시적 속성뿐만 아니라 속성어가 없는 암시적 속성까지 분석 대상으로 하는 속성카테고리 감성분류(ACSC, Aspect Category Sentiment Classification)는 속성기반 감성분석에서 중요한 의미를 지니고 있다. 본 연구는 속성카테고리 감성분류에 BERT 사전훈련 언어 모델을 적용할 때 기존 연구에서 다루지 않은 다음과 같은 주요 이슈들에 대한 답을 찾고, 이를 통해 우수한 ACSC 모델 구조를 도출하고자 한다. 첫째, [CLS] 토큰의 출력 벡터만 분류벡터로 사용하기보다는 속성카테고리에 대한 토큰들의 출력 벡터를 분류벡터에 반영하면 더 나은 성능을 달성할 수 있지 않을까? 둘째, 입력 데이터의 문장-쌍(sentence-pair) 구성에서 QA(Question Answering)와 NLI(Natural Language Inference) 타입 간 성능 차이가 존재할까? 셋째, 입력 데이터의 QA 또는 NLI 타입 문장-쌍 구성에서 속성카테고리를 포함한 문장의 순서에 따른 성능 차이가 존재할까?

이러한 연구 목적을 달성하기 위해 입력 및 출력 옵션들의 조합에 따라 12가지 ACSC 모델들을 구현하고 4종 영어 벤치마크 데이터셋에 대한 실험을 통해 기존 모델 이상의 성능을 제공하는 ACSC 모델들을 도출하였다. 그리고 [CLS] 토큰에 대한 출력 벡터를 분류벡터로 사용하기 보다는 속성카테고리 토큰의 출력 벡터를 사용하거나 두 가지를 함께 사용하는 것이 더욱 효과적이고, NLI 보다는 QA 타입의 입력이 대체적으로 더 나은 성능을 제공하며, QA 타입 안에서 속성이 포함된 문장의 순서는 성능과 무관한 점 등의 유용한 시사점들을 발견하였다. 본 연구에서 사용한 ACSC 모델 디자인을 위한 방법론은 다른 연구에도 비슷하게 응용될 수 있을 것으로 기대된다.

주제어 : 속성기반 감성분석, ABSA, 속성카테고리 감성분류, BERT, NLP

논문접수일 : 2020년 9월 23일 논문수정일 : 2020년 11월 2일 게재확정일 : 2020년 11월 23일
원고유형 : 일반논문 교신저자 : 신경식

1. 서론

감성분석(Sentiment Analysis)은 소비자나 대중이 작성한 글로부터 이들이 임의의 대상에 대해 느끼는 감성을 분석하는 자연어처리(NLP, Natural Language Processing) 작업중 하나이다 (Do et al., 2019; Liu, 2012; Schouten and Frasincar, 2016). 더 나아가, 속성기반 감성분석(ABSA, Aspect-Based Sentiment Analysis)은 대상이 가지고 있는 각 속성(aspect)에 대한 감성을

* 이 논문 또는 저서는 2017년도 대한민국 교육부와 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2017S1A5B5A02024287).

세부적으로 분석하는 것으로, 비즈니스 측면에서 더욱 실질적인 가치를 제공하기 때문에 학계뿐만 아니라 산업계의 주목을 받고 있다 (Araque et al., 2019; Dragoni et al., 2018; Park et al., 2020; Peng et al., 2018; Quan and Ren, 2014; Song et al., 2019; Xiaomei et al., 2018; Zhao et al., 2018). 예를 들어, "그 레스토랑은 비싸지만 음식은 정말 환상적이다." 라는 리뷰가 있을 때, 일반적인 감성분석은 '레스토랑'이라는 대상에 대한 전반적인 감성이 '긍정'이라고 평가하지만, ABSA는 레스토랑의 '가격'은 '부정', '음식'은 '긍정'과 같이 '가격'과 '음식'이라는 레스토랑의 속성에 대한 감성을 보다 정교하게 분석한다. 따라서, ABSA는 일반적인 감성분석에 비해 더욱 구체적이고 효과적인 마케팅 전략을 수립할 수 있게 해준다.

이러한 ABSA를 수행하기 위해서는 먼저 텍스트 안에 포함된 속성어(aspect terms)나 속성카테고리(aspect category)가 무엇인지를 규명해야 하고, 이에 대한 감성을 판단해야 한다. 그래서, ABSA에는 속성어 추출(aspect term extraction), 속성카테고리 감지(aspect category detection), 속성어 감성분류(ATSC, Aspect Term Sentiment Classification), 속성카테고리 감성분류(ACSC, Aspect Category Sentiment Classification) 등의 네 가지 주요 영역이 존재한다 (Park et al., 2020; Zhu et al., 2019). 대개 속성어를 추출하고 주어진 속성어에 대한 감성을 분석하는 ATSC를 수행하거나, 속성카테고리를 추출하고 주어진 속성카테고리에 대한 감성을 분석하는 ACSC를 수행하는 방식으로 활용된다.

여기에서 속성카테고리(aspect category)는 하나 이상의 속성어(aspect term)로 표현되거나, 다른 단어에 의해 간접적으로 추론된다. 앞 예시 문장에서, '가격'과 '음식'은 모두 속성카테고리이며, '음식'이라는 속성카테고리는 리뷰에 포함된 '음식'이라는 속성어에 의해 표현된다. 리뷰 문장에 '파스타', '스테이크', 또는 '그릴드 치킨 스페셜' 등이 포함되어 있다면 이것들은 모두 '음식'이라는 속성카테고리에 대한 속성어가 될 수 있다. 이와 같이, 구체적인 속성어에 의해 언급되는 속성카테고리를 명시적(explicit) 속성이라고 한다. 반면에, '가격'이라는 속성카테고리는 구체적인 속성어는 없지만 '비싸지만'과 같은 감성어로 간접적으로 추측할 수 있는데 이러한 속성카테고리를 암시적(implicit) 속성이라고 한다. 여기까지는 '속성어'에 대한 혼동을 피하기 위해 '속성카테고리'라는 단어를 사용했는데, 이후부터는 '속성카테고리'와 '속성'을 같은 개념으로 간주하고 편의상 '속성'이라는 단어를 더 많이 사용하기로 한다. 그리고 한 가지 주목해야 할 점은 ATSC는 속성어에 대한 감성을 분석하므로 명시적 속성만을 다루며, ACSC는 명시적 속성 뿐만 아니라 암시적 속성도 분석 대상으로 한다는 것이다.

기존의 딥러닝을 활용한 연구들을 살펴보면 ATSC가 훨씬 많고 ACSC에 대한 연구는 상대적으로 부족한 상황이다 (Hai et al., 2011; Zhu et al., 2019). 이러한 현상은 여러 NLP 영역에서 뚜렷한 성능 향상을 가능하게 한 BERT(Bidirectional Encoder Representations from Transformers) 사전훈련(pre-training) 언어 모델 등장 이전과 이후 모든 기간에 공통적이다. BERT 이전의 ATSC 연구들은 주로 LSTM(Long Short-Term Memory)이나 GRU(Gated Recurrent Unit)를 활용하여 속성어와 다른 컨텍스트 단어 간의 상관 정도를 고려한 콘텐트 어텐션(content attention)과 속성어의 위치를 반영한 위치 어텐션(location attention)

을 효과적으로 구현하는 모델 구조에 집중했었다 (Chen et al., 2017; Liu et al., 2018; Ma et al., 2017; Tang et al., 2016; Wang et al., 2016). 콘텐트 및 위치 어텐션은 임의의 속성에 대한 감성어를 정확히 매치하기 위해 문장을 구성하고 있는 단어들의 중요도를 해당 속성을 중심으로 계산하여 딥러닝 모델에 반영하기 위한 메커니즘이다. 이에 반해, BERT 이후의 ATSC 연구들은 대부분 BERT를 효과적으로 활용하는 방안에 대해 다루고 있다. 여기에는 크게 BERT 출력물을 통합하여 감성분류로 연결하는 최종단의 구조에 대한 연구와 추가적인 데이터셋에 대한 사후훈련(post-training)으로 BERT 가중치를 해당 도메인에 더욱 적합하게 개선하는 방안에 대한 연구가 있다 (Gao et al, 2019; Li et al., 2019; Rietzler et al., 2019; Song et al., 2020; Xu et al., 2019; Zeng et al., 2019). 한편, BERT 이전의 ACSC 연구는 ACSC 자체가 속성어가 없는 경우도 포함하는 상황이므로 주로 위치 어텐션은 제외하고 콘텐트 어텐션을 효과적으로 구현하는 모델 구조를 모색하였다 (Khalil and El-Beltagy, 2016; Ruder et al., 2016; Wang et al., 2016). 이와 다르게, BERT 이후의 ACSC 연구에는 BERT 사후훈련에 관한 연구와 BERT를 적용하기 위해 문장-쌍 분류(sentence-pair classification) 문제로 입력 데이터셋을 구성하는 방안에 대한 연구가 있다 (Hoang et al., 2019; Sun et al., 2019). 전반적으로 ATSC와 ACSC에 대한 연구는 비슷한 방향으로 발전하고 있지만 ACSC에 대한 연구가 상대적으로 훨씬 적다는 것을 살펴볼 수 있다.

그런데 암시적 속성을 언급하는 문장들은 트위터나 상품 리뷰 데이터에 꽤 자주 나타난다 (Davidov et al., 2010; Dosoula et al., 2016; Zhu et al., 2011). ABSA 관련 벤치마크 데이터셋인 SemEval 2016 레스토랑 데이터셋에서는 암시적 속성 문장들이 약 25%를 차지한다.트윗에 포함된 암시적 속성들은 범죄를 감지하고 예방하기 위한 의미 있는 역할을 담당한다 (Hannach and Benkhalifa, 2018). 암시적 속성에 대한 문장들은 중요한 의견(opinion)을 전달하고 감성분석 시스템의 성능을 향상시켜준다 (Dosoula et al., 2016; Hannach and Benkhalifa, 2018; Tubishat et al., 2018; Zhu et al., 2011). 따라서 암시적 속성에 내재된 소중한 정보 가치를 잃지 않기 위해서는 ACSC에 대한 좀 더 활발한 연구가 필요하다.

그리고 다른 NLP 영역에서와 마찬가지로 기존 감성분류 연구에서도 BERT의 [CLS] 토큰에 대한 최종 벡터를 감성분류벡터로 사용하고 있는데 (Hoang et al., 2019; Rietzler et al., 2019; Song et al., 2020; Sun et al., 2019; Xu et al., 2019), 속성어나 속성을 나타내는 토큰에 대한 최종 벡터도 감성분류벡터에 반영해 볼 필요가 있다. 왜냐하면, ATSC와 ACSC에서는 주어진 속성어나 속성에 따라 감성이 달라질 수 있으므로 속성어나 속성을 나타내는 토큰에 대한 최종 벡터에 의미 있는 정보가 포함될 수 있기 때문이다. 또, BERT를 적용하기 위해 여러 가능한 문장-쌍 형태 간 성능 차이에 대한 고려 없이 대부분 입력 데이터를 임의의 문장-쌍(sentence-pair) 형태로 변환해 주었는데, 이에 대한 체계적인 검토가 필요하다.

따라서, 본 연구는 암시적 속성도 분석 대상으로 포함하는 ACSC에 대해BERT의 출력단 구조와 입력 데이터 형식의 다양한 조합에 따라 12가지 ACSC 모델을 구현하고 4종 영어 벤치마크 데이터셋에 대한 체계적인 실험을 수행한다. 이를통해, 우수한 ACSC 모델을 도출하고

BERT의 활용 가치를 높일 수 있는 실제적인 방안을 도출한다. 구체적인 연구 목적은 관련 지식을 좀 더 서술한 후 3.3절에서 제시된다. 본 논문의 공헌점을 간단하게 정리하면 다음과 같다. 첫째, 연구가 부족한 ACSC 영역에서 ACSC의 특수성을 고려하여 BERT를 효과적으로 적용할 수 있는 모델 구조를 제안하였다. 둘째, 기존 연구에서 주목하지 않았던 BERT의 [CLS] 토큰 출력 벡터 사용, QA(Question Answering)나 NLI(Natural Language Inference) 형태의 문장-쌍(sentence-pair) 입력 및 문장 구성 순서 등에 대한 구체적이고 유용한 시사점을 도출하였다. 셋째, 본 연구에서 ACSC 모델 디자인 이슈들에 대해 체계적으로 접근하기 위해 사용한 새로운 방법은 ATSC 연구와 BERT 이외의 다른 모델 적용을 위해서도 응용할 수 있는 시사점을 제공한다.

2. 관련 연구

2.1. BERT

최근 ELMo (Peters et al., 2018), OpenAI GPT (Radford et al., 2018), 그리고BERT (Devlin et al., 2018)와 같은 사전훈련 언어 모델들은 피쳐엔지니어링(feature engineering)의 노력 비용을 절감해줌으로써 그 효과성을 입증했다. 특히, 구글에서 2018년 말에 공개한 BERT(Bidirectional Encoder Representations from Transformers)는 주어진 질문에 대한 해답 부분을 찾는 QA와 두 문장의 관계를 예측하는 NLI에서 우수한 성적을 달성하였다. BERT는 컨텍스트를 반영한 언어 표현 (Peters et al., 2018), 양방향(bidirectional) 트랜스포머(transformer) 구조 (Vaswani et al., 2017), 하위(downstream) 작업을 위한 연속적인 엔드-투-엔드(end-to-end) 파인튜닝(finetuning)으로 연결되는 언어 모델의 사전훈련(pre-training) (Radford and Salimans, 2018; Howard and Ruder, 2018) 등 이전의 혁신적인 연구들을 토대로 개발되었다. 대규모 위키피디아(Wikipedia) 아티클(article) 코퍼스(corpus)에 대해 사전훈련된 BERT 모델을 다른 작업을 위해 파인튜닝하여 적용하면 된다. BERT 구조는 양방향(bidirectional)으로 작동하기 때문에 여러 NLP 작업들의 성능을 개선하는 강력한 시퀀스(sequence) 표현을 제공한다 (Devlin et al., 2018).

BERT의 핵심적인 아이디어는 매스크 언어 모델(MLM, Masked Language Model)과 다음-문장예측(next-sentence prediction)이다. 매스크 언어 모델은 모델이 랜덤하게 매스크 처리된 토큰들을 컨텍스트를 고려하여 예측하는 방법을 학습하는 것이다. 그리고 다음-문장 예측(next-sentence prediction)으로 문장 B가 이전 문장 A 뒤에 자연스럽게 연결되는지 여부를 예측하도록 학습하는 것이다. 이러한 학습 목표는 기존의 다음-단어 예측(next-word prediction)에 비해 양방향 예측이 가능하고 좀 더 긴(long-term) 의존관계를 잘 포착할 수 있도록 해주는 장점을 가지고 있다.

특정 작업에 BERT를 적용하기 위해서는 입력 데이터를 단일 문장이나 문장-쌍(sentence-pair)에 대한 토큰(token)의 시퀀스(sequence)로 나타낸다.하나의 토큰에 대한 입력 벡터는 해당 토큰 임베딩과 시그먼트(segment) 임베딩 그리고 위치(position) 임베딩의 합으로 계산된다. 토큰 임베딩 시퀀스는 분류(classification) 토큰으로 맨 앞에 [CLS] 토큰을, 문장 분리자(separator)로 문

장과 문장 사이에 [SEP] 토큰을 배치하여 구성한다. 예를 들어, 단일 문장의 경우에는 "[CLS] 문장 [SEP]" , 문장-쌍의 경우에는 "[CLS] 문장 A [SEP] 문장 B [SEP]" 형식으로 구성한다. 그리고 이렇게 입력된 단일 문장 또는 문장 쌍에 대한 분류(classification) 작업은 대개 [CLS] 토큰에 해당하는 출력 벡터를 전결합(fully-connected) 계층으로, 이후 소프트맥스(softmax) 계층으로 연결함으로써 수행된다.

BERT에는 다음과 같은 두 가지 패러미터 세팅이 존재한다.

- BERT-Base: 12개의 트랜스포머 블록, 768차원의 히든 레이어, 12개의 셀프 어텐션 헤드를 가지고 있고, 사전훈련 모델의 패러미터 총 수는 110M.
- BERT-Large: 24개의 트랜스포머 블록, 1024차원의 히든 레이어, 16개의 셀프 어텐션 헤드를 가지고 있고, 사전훈련 모델의 패러미터 총 수는 340M.

BERT-Large는 BERT-Base보다 메모리 요구량이 훨씬 많기 때문에 대부분의 연구에서는 BERT-Base 모델의 가중치를 해당 작업을 위한 파인튜닝의 초기값으로 사용한다.

2.2. BERT를 이용한 ATSC 연구

앞에서 언급한 바와 같이, ACSC 연구는 ATSC(Aspect Term Sentiment Classification)와 비슷한 방향으로 진화해왔기 때문에 ATSC에 대한 연구도 참고할 필요가 있다. ATSC 연구는 대략적으로 BERT를 더욱 효과적으로 활용하기 위해 BERT 출력물을 처리하는 최종단의 구조를 디자인하거나 (Gao et al, 2019; Li et al., 2019; Song et al., 2020; Zeng et al., 2019), 해당 도메인에 사전훈련된 BERT 모델을 적용하기 전에 추가적인 사후훈련(post-training)을 통해 BERT 가중치를 더욱 개선하기 위한 방안을 모색해왔다 (Rietzler et al., 2019; Xu et al., 2019). 먼저, 최종단 디자인 관련 연구에 대해 좀 더 자세히 살펴보면, BERT 임베딩 중간 및 최종 계층의 [CLS] 토큰들을 LSTM이나 SAN(Self-Attention Networks)으로 통합한 후 감성분류에 사용한 연구가 있다 (Song et al., 2020). 그리고 BERT 최종 출력물 중 속성어에 해당하는 벡터들에 대해 각 차원의 최대값을 취하는 맥스풀링(max-pooling)을 한 후 이것을 최종 [CLS] 토큰 벡터와 연결하거나 차원별로 곱한 벡터를 감성분류에 사용하는 구조를 제안한 연구도 있다 (Gao et al, 2019). 또, 각 토큰(token)에 대한 BERT 최종 출력물들을 선형계층(Linear Layer), GRU(Gated Recurrent Unit), SAN(Self-Attention Networks), 또는 CRF(Conditional Random Fields) 등을 통해 변환하여 시퀀스 레이블링(Sequence Labeling)하는 문제로 접근함으로써, 속성어(aspect term)와 이에 대한 감성(sentiment)을 동시에 추출한 연구도 있다 (Li et al., 2019). 한편, "[CLS] 리뷰 문장 [SEP]" 으로 이루어진 로컬 컨텍스트 모듈과 "[CLS] 리뷰 문장 [SEP] 속성어 [SEP]" 으로 이루어진 글로벌 컨텍스트 모듈에 대해 BERT 임베딩을 적용하고 각 출력물에 멀티헤드 셀프어텐션을 적용한 후 연결하여 감성분류에 사용한 연구도 있다 (Zeng et al., 2019). BERT를 이용한 ATSC에 관한 다른 연구들에서는 위치 어텐션을 고려하지 않았는데, 이 연구의 로컬 컨텍스트 모듈에서는 리뷰 문장에 포함된 속성어를 중심으로 의미적 상대 거리(semantic-relative distance) 이내에 있는 토큰들의 경우 해당 속성어에 대한 로컬 컨텍스트로 간주하여 비중을 높여주는 메커니즘

을 적용하였다.

다음으로, BERT 사후훈련(post-training) 관련 연구에는 BERT-base를 같은 도메인의 풍부한 데이터로 파인튜닝을 하여 사용하면 성능이 향상된다는 사실을 검증한 연구가 있다 (Rietzler et al., 2019). 이 때 아마존(Amazon) 랩탑(Laptop) 리뷰 데이터로 BERT-base를 파인튜닝한 모델을 레스토랑(Restaurant) 리뷰 데이터로 훈련시키고 랩탑 데이터에 대해 테스트하는 것과 같은 크로스-도메인 방식으로 적용하는 경우에도 단순한 BERT-base나 XLNet-base 모델보다 성능이 좋았다고 한다. 한편, BERT 사전훈련(pre-training) 가중치를 기본적인 언어 이해를 위한 초기값으로 사용하고, 도메인(domain) 지식과 태스크(task) 지식을 이용하여 BERT를 개선하는 조인트(joint) 사후훈련(post-training) 기법을 제안한 연구가 있다 (Xu et al., 2019). 도메인 지식을 반영하기 위해서는 아마존 랩탑 리뷰나 옐프(Yelp) 레스토랑 리뷰 데이터에 대해 MLM (Masked Language Model) 외에 NSP(Next Sentence Prediction)와 비슷하게 두 문장이 같은 리뷰에 속하는지 여부를 예측했다. 이와 동시에, 태스크 지식을 투입하기 위해 MRC(Machine Reading Comprehension)를 위한 대규모 데이터셋인 SQuAD 1.1. (Rajpurkar et al., 2016)에 대해 임의의 질문(Question)에 대한 대답(Answer)에 해당하는 시작 및 끝 토큰 스팬(span)을 예측하는 조인트(joint) 손실(loss) 함수를 사용하여 훈련시켰다. 이들은 BERT가 여러 NLP 작업에 활용될 수 있도록 컨텍스트를 고려한 언어 표현을 배우도록 설계되어 있지만, 위키피디아(Wikipedia) 기사(article)들로 훈련되었기 때문에 글쓴이의 의견을 담고 있는 오피니온(opinion) 텍스트 도메인에 적용하거나 리뷰에 대한 이해도를 묻는 RRC(Review Reading Comprehension) 태스크에 적용하려면 한계점이 수반된다는 점에 주목했던 것이다.

2.3. BERT를 이용한 ACSC 연구

ACSC(Aspect Category Sentiment Classification) 분야에도 BERT 사후훈련에 관한 연구가 있었고 (Hoang et al., 2019), BERT를 활용하기 위해 문장-쌍 분류(sentence-pair classification) 문제로 입력 데이터셋을 구성하는 방안에 대한 연구가 있었다 (Sun et al., 2019). ATSC에 비해 관련 연구가 부족한 상황이다. 먼저, BERT 사후훈련 연구로는 사전훈련된 BERT를 기반으로 하는 속성분류, 감성분류, 또는 속성 및 감성 통합 분류를 위한 모델을 랩탑, 레스토랑, 또는 랩탑과 레스토랑 통합 데이터로 파인튜닝하여, 통합 모델을 통합 데이터로 훈련 시킬 때 감성분류 성능이 높다는 결과가 제시된 바 있다 (Hoang et al., 2019). 여기에서 데이터는 랩탑 및 레스토랑 벤치마크 데이터셋에 포함된 하나의 리뷰 문장에 대해 해당 문장에서 언급되고 있는 속성 외에 다른 도메인 속성들에 대해서도 문장-쌍 형태로 추가적으로 생성하여확장한 것이다. 다음으로, 입력 데이터 구성 관련 연구로는 속성(aspect)으로부터 보조적인(auxiliary) 문장을 생성하여 ABSA를 QA(Question Answering)나 자연어추론(NLI, Natural Language Inference)과 같은 BERT를 활용한 문장-쌍(sentence-pair) 분류 태스크로 변환하여 효과성을 비교 및 검증한 연구가 있다 (Sun et al., 2019).

3. 연구 목적 및 방법

BERT 모델을 ACSC 연구에 효과적으로 적용하고자 하는 본 연구의 구체적인 목적과 방법을 서술하기 전에 먼저 관련 내용에 대해 좀 더 설명하겠다.

3.1. BERT 입력 데이터 형식

일차적으로, BERT를 단일 문장(single sentence) 입력 포맷으로 적용할 수도 있겠지만 이렇게 하면 ACSC에서는 속성에 대한 정보를 딥러닝 모델에 반영할 수 없는 치명적인 약점이 발생한다. 실제로 단일 문장 포맷보다 문장-쌍 포맷이 더 나은 성능을 제공한다는 연구 결과도 있으므로 (Sun et al., 2019), 본 연구에서는 단일 문장 포맷은 고려 대상에서 제외하기로 한다.

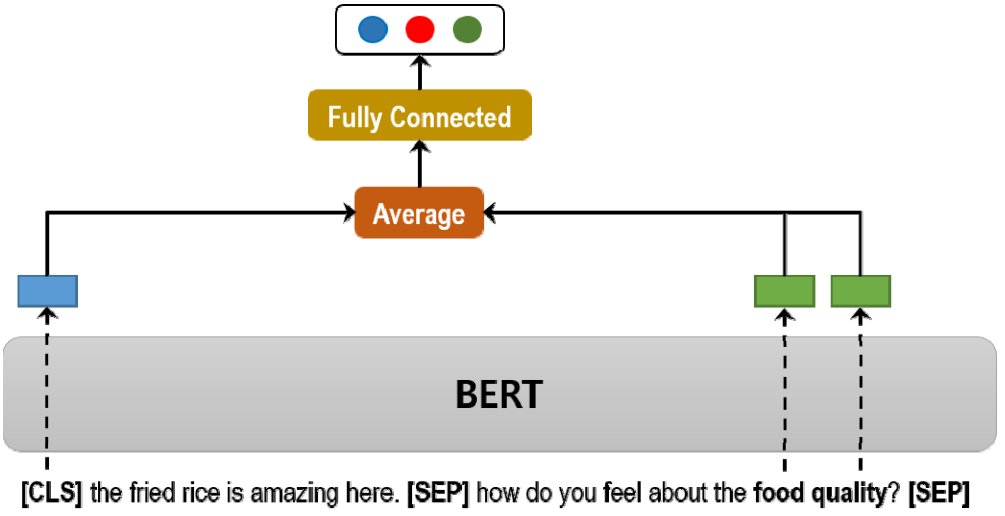
다음으로, ATSC나 ACSC를 BERT를 이용한 문장-쌍(sentence-pair) 분류 문제로 변환할 때, NLI나 QA 타입(type)을 활용할 수 있다. "The fried rice is amazing here."라는 리뷰가 있을 때, ATSC의 NLI에서는 "[CLS] the fried rice is amazing here. [SEP] fried rice [SEP]" 와 같이 속성어 "fried rice"를 하나의 문장처럼 넣어준다. 반면에 본 논문에서 다루고 있는 ACSC의 NLI에서는 속성어 "fried rice" 대신에 이에 대한 속성카테고리인 "food quality"란 단어를 사용하여 "[CLS] the fried rice is amazing here. [SEP] food quality [SEP]"와 같은 형식으로 입력 데이터를 구성한다. ACSC의 QA에서는 "[CLS] the fried rice is amazing here. [SEP] how do you feel about the food quality? [SEP]"와 같이 단순한 단어의 나열 대신에 의문문을 만들어 넣어준다.

그런데 여기에서 속성카테고리를 두 번째 문장에 넣을 수도 있겠지만 첫 번째 문장에 오게 할 수도 있을 것이다. 기존 연구에서는 문장 순서에 대한 검증이나 언급 없이 임의적으로 첫 번째 (Sun et al., 2019; Xu et al., 2019) 또는 두 번째 (ang et al., 2019; Rietzler et al., 2019) 문장으로 처리하고 있다. 본 연구에서는 문장 순서에 따른 성능 차이가 있는지를 살펴보기 위해 NLI 또는 QA 입력 타입 각각에 대해 속성카테고리 단어들을 첫 번째 또는 두 번째 문장에 위치시키는 옵션들을 모두 고려한다. <Figure 1>은 QA 입력 포맷인 경우 속성카테고리를 두 번째 문장에 넣는 경우를 보여주고 있다.

이 외에 문장-쌍 분류 문제로 변환할 때 ‘yes’/‘no’ 중 하나로 답하도록 의문문 샘플을 확장하여 생성하는 방안도 보고된 바 있다 (Sun et al., 2019). 예를 들어, "what do you think of the safety?" 라는 질문 대신에 "is the polarity of the aspect safety positive?" 와 같이 ‘yes’/‘no’ 중 하나로 답할 수 있는 의문문을 여러 개 만들어준다. 예를 들어, 앞 문장의 ‘positive’를 ‘negative’, ‘neutral’, ‘none’ 등으로 대체하면서 새로운 문장을 생성하는 것이다. ‘none’이 포함된 의문문은 해당 리뷰 문장에 관심 속성이 없을 때 ‘yes’가 된다. 그런데 본 연구에서는 데이터를 확장하기보다는 모델의 구조적인 차이에 따른 성능 차이에 더 집중하기 위해 이러한 옵션은 생략하기로 한다.

3.2. BERT 출력단 구조

BERT를 활용한 기존 연구에서는 대부분 [CLS] 토큰에 대한 최종 벡터를 분류벡터로 사용하였다(Hoang et al., 2019; Rietzler et al., 2019; Song et al., 2020; Sun et al., 2019; Xu et al.,



〈Figure 1〉A Model Architecture of QA-Second-Type Input and [CLS]-Aspect-Category Output

2019). 그런데 ACSC는 주어진 속성카테고리에 대한 감성을 도출하는 것이므로 속성카테고리를 나타내는 토큰들을 중심으로 다른 토큰들에 대한 의미적 관계를 반영한 속성카테고리에 대한 토큰들의 평균 벡터에는 더욱 가치있는 정보가 포함될 수 있다. 따라서, [CLS] 토큰만 고려하는 경우 외에, 속성카테고리에 대한 토큰들의 평균 벡터를 고려하는 경우, 그리고 [CLS] 토큰과 속성카테고리에 대한 토큰들의 평균 벡터를 고려하는 경우를 비교해볼 수 있다. 이 외에 모든 토큰들에 대한 최종 출력 벡터들을 LSTM으로 통합하든지, 속성카테고리에 대한 토큰들의 평균 벡터를 각 토큰의 출력 벡터에 연결하여 콘텐트 어텐션을 적용한 분류벡터를 도출하는 등의 좀 더 복잡한 구조도 가능할 것이다. 그러나 예비분석 결과 성능 향상 정도가 뚜렷하지 않아 본 논문에서는 고려 대상에서 제외하기로 한다. <Figure 1>은 QA 입력 타입인 경우 [CLS] 토큰 벡터와 속성카테고리 토큰 벡터를 모두 반영하여 분류벡터로 사용하는 경우를 보여주고 있다.

3.3. 연구 목적

본 연구의 궁극적인 목적은 BERT 모델을 ACSC에 효과적으로 적용하기 위한 방안을 도출하는 것이다. 좀 더 구체적으로는 다음 연구 질문들에 대한 해답을 찾는 것이라고 할 수 있다.

- RQ1: BERT 입력 및 출력 계층 옵션 조합 중 우수한 ACSC 모델 구조는 무엇일까?
- RQ2-1: [CLS] 토큰의 출력 벡터만 분류벡터로 사용하기보다는 속성카테고리에 대한 토큰들의 출력 벡터를 반영하면 더 나은 성능에 도달할 수 있지 않을까?
- RQ2-2: 입력 데이터의 문장-쌍 구성에서

〈Table 1〉 12 Implemented ACSC Models

INPUT Format	OUTPUT Structure		
	[CLS] Token	Mean of Aspect Category Tokens	Mean of [CLS] & Aspect Catogory Tokens
NLI	Bert_Asp_First_CLS	Bert_Asp_First_MEAN	Bert_Asp_First_CLS_MEAN
	Bert_Asp_Second_CLS	Bert_Asp_Second_MEAN	Bert_Asp_Second_CLS_MEAN
QA	Bert_Quest_First_CLS	Bert_Quest_First_MEAN	Bert_Quest_First_CLS_MEAN
	Bert_Quest_Second_CLS	Bert_Quest_Second_MEAN	Bert_Quest_Second_CLS_MEAN

QA와 NLI 타입 간 성능 차이가 존재할까?

- RQ2-3: 입력 데이터의 QA 또는 NLI 타입 문장-쌍 구성에서 속성카테고리를 포함한 문장의 순서에 따른 성능 차이가 존재할까?

3.4. 연구 방법

앞의 연구 목적을 달성하기 위해 BERT 모델의 입력 데이터 형식과 출력단 구조 옵션에 따라 <Table 1>과 같이 12가지 ACSC 모델들을 도출할 수 있다. 예를 들어, "Bert_Quest_Second_MEAN" 모델은 QA(Quest) 타입이면서 속성카테고리 단어들이 두 번째(Second) 문장에 들어가는 형식의 입력 데이터를 받아 출력단에서는속성카테고리에 대한 토큰들의 평균(MEAN)을 분류벡터로 사용하는 구조를 갖는다.

그리고 <Table 1>의 12가지 모델들을 4개의 영어 벤치마크 데이터셋(4.1절 참조)에 대해 훈련 및 테스트해본다. 이후 다른 조건은 동일하지만 한 가지 옵션의 차이로 인한 모델 구조 차이가 모델의 성능 차이로 연결되는지를 검증한다. 또 12가지 모델 중 상대적으로 성능이 좋은 상위 모델들을 기존 연구에서 제시한 모델들과 비교해본다.

4. 실험

4.1. 데이터셋

본 연구를 위해 세계적으로 사용되고 있는 4종 영어 벤치마크 데이터셋인 SemEval 2014의 Restaurant-2014 (Pontiki et al., 2014), SemEval 2015의 Laptop-2015 (Pontiki et al., 2015), 그리고 SemEval 2016의 Laptop-2016과 Restaurant-2016 (Pontiki et al., 2016)을 사용한다. 원래 데이터에는 ‘Positive’, ‘Negative’, ‘Neutral’, 그리고 ‘Conflict’ 등 4개의 감성 레이블 클래스가존재한다. 그런데 임의의 속성에 대해 긍정과 부정의 상반된 감성을 표현하는 ‘Conflict’에 해당하는 샘플들은 수도 적고 훈련에 방해가 될 수 있기 때문에 다른 연구와 같이 실험에서 제외하였다 (Chen et al., 2017; Liu et al., 2018; Tang et al., 2016; Zhu et al., 2019). 데이터셋의 감성 레이블에 대한 통계 정보는 <Table 2>에서 살펴볼 수 있다.

<Table 3>은 각 데이터셋에 포함된 속성 문장 수를 전체 고유한 속성의 수로 나누어 계산되는 하나의 속성당 속성 문장 수를 보여주고 있다. 속성 문장은 하나의 리뷰 문장에 여러 개의 속성이 포함될 경우 각 속성에 대해 해당 리뷰 문장

〈Table 2〉 Statistics about the Sentiment Label of the Datasets

Dataset		Pos.		Neg.		Neu.		Total	
		Num.	Per.	Num.	Per.	Num.	Per.	Num.	Per.
Restaurant 2014	Train	2,179	62 %	839	24 %	500	14 %	3,518	100 %
	Test	657	67 %	222	23 %	94	10 %	973	100 %
Restaurant 2016	Train	1,657	66 %	749	30 %	101	4 %	2,507	100 %
	Test	611	71 %	204	24 %	44	5 %	859	100 %
Laptop 2015	Train	1103	56 %	106	5 %	765	39 %	1,974	100 %
	Test	541	57%	79	8 %	329	35 %	949	100 %
Laptop 2016	Train	1,637	56 %	1,084	37 %	188	7 %	2,909	100 %
	Test	481	60 %	274	34 %	46	6 %	801	100 %

<Table 3> Statistics about the Number of Sentences per Aspect of the Datasets.

Dataset		Pos.		Neg.		Neu.		Total		Aspect
		Sent.	Sent. / Aspect	Sent.	Sent. / Aspect	Sent.	Sent. / Aspect	Sent.	Sent. / Aspect	
Restaurant 2014	Train	2,179	435.8	839	167.8	500	100.0	3,518	703.6	5
	Test	657	131.4	222	44.4	94	18.8	973	194.6	5
Restaurant 2016	Train	1,657	138.1	749	62.4	101	8.4	2,507	208.9	12
	Test	611	50.9	204	17.0	44	3.7	859	71.6	12
Laptop 2015	Train	1,103	13.6	106	1.3	765	9.4	1974	24.4	81
	Test	541	9.3	79	1.4	329	5.7	949	16.4	58
Laptop 2016	Train	1,637	18.6	1,084	12.3	188	2.1	2,909	33.1	88
	Test	481	5.5	274	3.1	46	0.5	801	9.1	88

을 카피해서 각 속성을 중심으로 감성 레이블을 태깅한 것이다. Restaurant-2014 데이터셋은 food, service, price, ambiance, anecdotes/miscellaneous 등 전체 5가지 속성을, Restaurant-2016은 RESTAURANT#GENERAL, FOOD#QUALITY, DRINKS#PRICES 등 전체 12가지의 속성을 가지고 있다. 그리고 Laptop-2015는 LAPTOP#OPERATION_PERFORMANCE, display#usability, graphics#design_features 등 전체 81개의 속성을 가지고 있는데 테스트셋에서는 이중 58개 속성에 대해서만 검증하고 있다. 그리고 Laptop-2016은 LAPTOP#USABILITY, LAPTOP#PRICE, DISPLAY#QUALITY 등 전체 88가지의 속성에 대한 감성을 다루고 있다. Restaurant-2014의 경우 하나의 속성당 속성 문장 수가 상대적으로 훨씬 많고 Restaurant-2016도 충분히 많은데, 랩탑 데이터셋들은 매우 적으므로 같은 ACSC 모델에 대한 평가 척도 점수는 레스토랑 데이터셋이 더 좋을 가능성이 높다고 할 수 있다.

4.2. 하이퍼패러미터(hyperparameters) 세팅 및 프로그램 실행

본 연구에서는 대소문자 구별이 없는 'BERT-base-uncased' 모델을 사용하였다. 각 도메인 데이터에 대해 BERT-base 모델을 파인튜닝할 때, 에픽(epoch) 수는 10, 배치(batch) 사이즈는 16으로 설정하였다. 드롭아웃(dropout) 확률은 튜닝 과정을 통해 0.2, 0.1, 0 중 0으로 세팅하였다. 그리고 모든 학습 가능한 패러미터들은 Xavier 유니폼(Uniform) 이니셜라이저(initializer)로 초기화하였고, Adam 옵티마이저(optimizer)를 사용하였으며 학습률(learning rate)은 2e-5, L2정규화(regularization) 계수는 0.01로 설정하였다. 모델 구현과 실행은 Windows10의 NVIDIA CUDA 10.0 환경에서 파이썬 3.7 및 파이토치(PyTorch) 1.4를 사용했다.

4.3. 평가 척도

서로 다른 모델의 성능을 비교하기 위해 본 논문에서는 두 가지 평가 척도를 사용한다. 첫 번째는 전체 테스트 샘플 수에 대한 정확하게 분류된 샘플 수의 비율인 정확도(accuracy)이다. 두 번째는 각 감성 레이블 클래스에 대한 F1 점수들의 평균 값인 매크로(macro-average) F1이다. 각 감성 레이블 클래스에 대한 F1 점수는 해당 클래스의 프리시젼(precision)과 리콜(recall)의 조화평균(harmonic mean)으로 계산된다. 그리고 본 연구에서는 랜덤 요인에 의한 편차의 영향을 최소화하기 위해 서로 다른 랜덤 시드(seed)를 사용한 10회 실험에 대한 정확도와 매크로 F1 점수의 평균을 사용하기로 한다.

4.4. 비교 대상 모델

4.4.1. BERT 이전 모델

- ATAE-LSTM: 기존 LSTM에 속성에 대한 콘텐트 어텐션을 추가하여 분류 성능을 제고한 모델이다. 콘텐트 어텐션은 LSTM 출력 벡터인 각 단어에 대한 히든(hidden)벡터와 속성벡터 간의 의미적 연관성의 정도에 따라 결정되며, 속성벡터는 워드벡터와는 다른 속성 임베딩(embedding) 스페이스로부터 생성된다. 워드와 속성 임베딩 벡터 모두 훈련 과정을 통해 최적화된다 (Wang et al., 2016).
- CNN Ensemble: 워드벡터를 옐프(Yelp) 레스토랑 리뷰와 아마존 전자제품 리뷰로 동적(dynamic) CNN을 통해 튜닝하고, 이를 사용하여 도메인 및 속성 정보를 반영한 세 개의 정적(static) CNN에 의한 투표(Voting)로 이루어지는 앙상블(ensemble) 기법을 통해 해당 속성에 대한 감성을 결정한다 (Khalil and El-Beltagy, 2016).
- Aspect-concatenating CNN: 속성벡터(aspect vector)는 속성을 이루고 있는 토큰 벡터들의 평균으로 도출한다. 예를 들어, "LAPTOP#PRICE"라는 속성에 대한 속성벡터는 'laptop'과 'price'에 대한 워드벡터들의 평균을 구하면 된다. 이렇게 구한 속성벡터를 리뷰를 구성하고 있는 토큰들의 워드벡터에 연결하여 컨볼루션(convolution), 맥스풀링(maxpooling), 그리고 소프트맥스(softmax) 계층을 통과시켜 해당 속성에 대한 감성을 판단한다 (Ruder et al., 2016).
- AAL-SS: Bi-LSTM에 의해 생성된 각 토큰에 대한 히든(hidden)벡터들은 해당 속성에 대한 감성 분류벡터를 도출하기 위해 속성벡터와

함께 세 계층으로 이루어진 감성 메모리 네트워크로 입력된다. 히든벡터들은 AAL(Aspect Aware Learning) 컴포넌트로도 입력되어 해당 속성과 각 단어들의 상관 정보가 손실함수에 반영된다 (Zhu et al., 2019).

4.4.2. BERT 이후 모델

- BERT-pair-QA-B:BERT 모델을 ABSA에 적용하기 위해 QA 타입 문장-쌍 분류 태스크로 변환한 모델 중의 하나로, 3.1절에서 설명한 바와 같이 ‘yes’/‘no’ 중 하나로 답할 수 있는 바이너리(Binary) 형태의 의문문을 여러 개 만들어 데이터를 확장하여 돌린다 (Sun et al., 2019). 속성은 첫 번째 문장에 포함되고, 분류 벡터로는 [CLS] 토큰에 대한 최종 출력 벡터를 사용한다.
- COM_BOTH: 속성분류와 감성분류를 위한 통합 모델을 랩탑과 레스토랑 전체 데이터를 훈련 데이터로 사용하여 파인튜닝한 모델이다 (Hoang et al., 2019). 이 모델을 해당 도메인의 테스트 데이터에 대해 검증한다. 통합 모델은 ‘positive’, ‘negative’, ‘neutral’, ‘conflict’ 등의 감성 레이블들을 ‘related’ 레이블로 매핑함으로써 속성 분류 모델로 사용되거나, ‘unrelated’ 레이블을 무시함으로써 감성분류 모델로 사용될 수 있다. 또는 이와 비슷하게 속성과 감성을 동시에 분류하도록 사용될 수도 있다.

5. 결과 분석

5.1. 상위 ACSC 모델 성능 비교

본 연구에서 구현한 <Table 1>의 12가지 모델 중 각 데이터셋별로 정확도나 매크로 F1 점수가 한 번이라도 최고 점수를 달성한 4종 모델을 <Table 4>의 하단부에 기록했다. 12가지 모델 모두에 대한 결과는 정확도 내림차순으로 정렬된 부록의 <Table A-1>과 <Table A-2>를 참조하기 바란다. 본 연구에서 구현하지 않은 모델들의 평가 점수는 해당 논문을 참조하였는데, 이들은 평균이 아닌 일회 최고 점수이기 때문에 샘플표준오차(sample standard error)가 없다. 4종 구현 모델인 Bert_Asp_Second_CLS_MEAN, Bert_Quest_Second_CLS_MEAN, Bert_Quest_Second_MEAN, 그리고 Bert_Quest_First_CLS_MEAN의점수는 10회 평균 점수임에도 기존 모델들보다 모두 더 높은 점수를 달성하였다. COM_BOTH 모델은 ‘conflict’ 레이블 클래스를 제거하지 않고 레스토랑과 랩탑 데이터를 통합하여 훈련시켰고, BERT-pair-QA-B 모델도 원래 데이터셋을 확장하여 돌린 점 등 BERT 이후의 모델에 대해서는 객관적인 비교가 어려운 부분이 있지만, 본 연구의 상위 4종 모델은 데이터셋 확장 없이도 우수한 점수를 획득했다는 측면에서 충분한 의미가 있다고 하겠다.

4종 우수 모델 중에 [CLS]만 사용하는 모델은 없는 점으로 미루어볼 때, 기존 연구들과 같이 단순히 [CLS] 토큰 벡터만 사용하는 것 보다 속성카테고리 토큰들에 대한 벡터를 함께 반영하는 것이 ACSC에서 유효하게 작용한다는 점을 예상할 수 있다. 실제로 각 데이터셋에서 최고 점수를 보인 모델과 이 옵션만 다른 모델에 대해 T 테스트를 해본 결과 거의 모든 경우에 유의수준 5% 또는 10%에서 유의한 차이가 있었다. 예를 들어, <Table A-1>에서 Restaurant-2014에 대한 정확도 기준에 의하면

〈Table 4〉 Main Results

Model	Restaurant_2014		Restaurant_2016		Laptop_2015		Laptop_2016	
	Acc.	Mac_F1	Acc.	Mac_F1	Acc.	Mac_F1	Acc.	Mac_F1
ATAE-LSTM	84.0	-	-	-	-	-	-	-
CNN Ensemble	-	-	85.45 U	-	-	-	77.40 U	-
Aspect-concatenating CNN	-	-	82.07 U 80.21 C	-	-	-	78.40 U 74.28 C	-
AAL-SS	85.61	75.54	84.35	67.14	-	-		-
BERT-pair-QA-B	89.9	-	-	-	-	-	-	-
COM_BOTH	89.8	-	-	-	-	-	82.8	-
Bert_Quest_First_CLS_MEAN	90.34 (0.49)	84.06 (1.07)	**89.09** (0.37)	73.55 (3.86)	84.07 (0.40)	70.30 (2.24)	83.55 (0.61)	66.65 (3.46)
Bert_Quest_Second_MEAN	90.59 (0.36)	83.92 (0.80)	88.56 (0.64)	**73.86** (2.16)	84.35 (0.63)	**71.58** (1.94)	83.40 (0.46)	66.38 (3.30)
Bert_Quest_Second_CLS_MEAN	**90.62** (0.44)	84.10 (0.84)	88.77 (0.54)	72.48 (2.11)	84.04 (0.65)	69.63 (3.50)	83.36 (0.46)	**67.39** (2.50)
Bert_Asp_Second_CLS_MEAN	90.58 (0.54)	**84.32** (1.14)	88.61 (0.41)	73.50 (2.30)	**84.46** (0.48)	69.06 (1.71)	**83.61** (0.76)	65.52 (3.97)

* The figures in parenthesis represent the sample standard errors and their paired value the sample mean with the best scores in bold.
* U stands for 'Unconstrained', using additional resources, such as lexicons or additional training data, and C 'Constrained', using only the provided training/development data in SemEval 2016.
* The values without parenthesis are referenced from their respective papers.
* "-" means not reported.

Bert_Quest_Second_CLS_MEAN 모델이 최고인데 이 모델과 Bert_Quest_Second_CLS 모델 성능을 T 테스트로 검증해보면, 유의수준 5%에서 유의한 차이를 보인다.

5.2. BERT 입력 및 출력 옵션별 성능 비교

BERT 모델 입력 및 출력 옵션간 성능 차이 비교는 이항분포(binomial distribution)의 정규분포 근사를 통해 검증해볼 수 있다. 예를 들어, RQ2-1의 [CLS] 토큰 벡터만 사용하는 방법에 대한 속성카테고리 평균 벡터 사용의 성능 우위 검증을 위해서는 다른 옵션은 동일한 모델들간에 쌍-비교 독립시행을 한다고 가정할 수 있다. 즉, <Table 3>에서 [CLS] 토큰 벡터만 사용하는 Bert_Asp_First_CLS 모델과 이와 다른 옵션은 같지만 [CLS] 토큰 벡터와 속성카테고리 평균 벡터를 함께 사용하는 Bert_Asp_First_CLS_MEAN 모델의 성능을 비교한다. 그리고 Bert_Asp_First_CLS 모델과 Bert_Asp_First_MEAN 모델도 비교한다. 이와 같이, Bert_Quest_Second_CLS 모델과 Bert_Quest_Second_CLS_MEAN 모델, 그리고 Bert_Quest_Second_CLS 모델과

〈Table 5〉 Results of 12 Models on Restaurant_2014 Dataset

Model	Mean_Acc.(%)	Mean_Mac_F1(%)
Bert_Asp_First_CLS	89.84	83.33
Bert_Asp_First_CLS_MEAN	90.45	83.87
Bert_Asp_First_MEAN	89.87	82.93
Bert_Asp_Second_CLS	90.05	83.20
Bert_Asp_Second_CLS_MEAN	90.58	84.32
Bert_Asp_Second_MEAN	90.39	83.93
Bert_Quest_First_CLS	89.95	83.21
Bert_Quest_First_CLS_MEAN	90.34	84.06
Bert_Quest_First_MEAN	90.06	83.27
Bert_Quest_Second_CLS	89.68	82.63
Bert_Quest_Second_CLS_MEAN	90.62	84.10
Bert_Quest_Second_MEAN	90.59	83.92

〈Table 6〉 Results of Pair-wise Comparison Regarding RQ1 on Restaurant Datasets

Dataset	Metric	Num_Comparison	Num_Greater_Second
Restaurant_2014	Mean_Acc	8	8
Restaurant_2016	Mean_Acc	8	5
Restaurant_2014	Mean_Mac_F1	8	7
Restaurant_2016	Mean_Mac_F1	8	6
Total Counts		32	26

Bert_Quest_Second_MEAN 모델 등 비슷한 방식으로 가능한 모든 쌍에 대해 비교한다. 성능은 정확도와 매크로 F1을 기준으로, 데이터는 4종 데이터셋에 대해 시행하여 전체 쌍 비교 횟수와 두 번째에 오는 모델 성능이 더 우수한 경우를 카운트한다. 랩탑과 레스토랑 데이터셋을 구분하여 검정하는 경우에는 예를 들어, <Table 6>과 같이 레스토랑 데이터셋에 대해 전체 쌍 비교 횟수와 두 번째 모델의 성능이 더 우수한 경우를 카운트하면 된다. 귀무가설(H0)은 "첫 번째 모델 성능은 두 번째 모델 성능 이하이다" 이고, 정규분포로 근사하여 유의수준 1%, 5%, 10% 단측검정을 수행한다. <Table 5>와 <Table 6>은 실제 결과이며 다른 결과들은 지면 관계상 생략한다.

RQ2-1에 대한 가설검정 결과는 랩탑과 레스토랑 데이터셋 모두에 대해 유의수준 1%에서 귀무가설이 기각되었다. 즉, [CLS] 토큰 벡터만 사용하는 것보다 속성카테고리 평균 벡터도 함께 사용할 때 분류기 성능이 유의하게 높아진다는 결론을 얻었다. <Table 4>의 4종 상위 모델에도 [CLS]만 사용한 모델은 존재하지 않는다. 이러한 결론은 다른 NLP 분야와 비슷하게 기존 ACSC 및 ATSC 연구에서도 대부분 최종 [CLS] 토큰 벡터만 분류벡터로 사용하는 경향을 수정할 필요가 있음을 보여준다.

다음으로, 입력 문장-쌍 구성에서 QA와 NLI 타입 간 성능 차이에 대한 RQ2-2 가설검정 결과는 랩탑과 레스토랑을 합해서 평균 정확도 기준으로 볼 때는 유의수준 10%에서, 평균 매크로 F1 기준으로 볼 때는 유의수준 5%에서 QA 타입이 나은 성능을 제공하는 것으로 판정되었다. 데이터셋별로 보면, 랩탑에 대해 유의수준 1%에서 QA 타입이 낫고, 레스토랑에 대해서는 QA와 NLI간 성능 차이가 유의하지 않았다. <Table 4>의 4종 상위 모델 중 3개가 QA 타입을 갖는다.

마지막으로, QA 또는 NLI 문장-쌍 구성에서 문장의 순서로 인한 성능 차이에 대한 RQ2-3 가설검정 결과, QA의 경우에는 랩탑과 레스토랑 모두에서 순서에 따른 성능 차이는 유의하지 않았다. 그리고 NLI의 경우에는 랩탑과 레스토랑 데이터셋을 합해서 볼 때 유의수준 1%에서 속성카테고리 문장을 두 번째에 위치시키는 형태가 더 나은 성능을 제공하는 것으로 나타났다. 또, 랩탑 데이터에 대해서는 유의수준 5%에서 속성카테고리 문장을 두 번째에 위치시키는 것이 나은 것으로 판단되었다. <Table 4>의 4종 상위 모델 중 QA 타입에는 속성카테고리가 첫 번째와 두 번째 문장에 오는 경우가 모두 존재하고, NLI 타입에는 두 번째에 오는 모델만 하나 찾아볼 수 있다.

6. 결론 및 시사점

본 연구에서는 BERT 언어 모델을 ACSC에 적용할 때 선택할 수 있는 입력 데이터 형식과 출력 계층 구조의 여러 옵션 조합에 따라 12가지 ACSC 모델들을 구현하고 4종 영어 벤치마크 데이터셋에 대해 실험한 후, 4종 상위 모델을 도출하였다. 그리고 각 옵션별 성능 비교를 통해 ACSC 모델 디자인에 대한 유용한 시사점을 제시하였다. 본 연구에서 제안한 4종 상위 모델은 데이터셋을 확장하지 않고도 기존 모델 이상의 성능을 제공한다. 특히, ACSC에서는 다른 NLP 영역과 다르게 [CLS] 토큰에 대한 출력 벡터만 분류벡터로 사용하는 것보다 속성을 나타내는 토큰들에 대한 출력 벡터를 반영하는 것이 중요한 성공 요인이라는 점을 검증하였다. 그리고 대체적으로 NLI 보다는 QA 타입의 문장-쌍 입력이 나은 성능을 제공하고 QA 타입 안에서 속성이 포함된 문장의 순서는 성능과 무관한 것으로 나타났다. 또, NLI 타입의 문장-쌍 입력을 사용할 경우에는 속성이 포함된 문장을 두 번째에 위치시키는 방안이 더 나을 가능성이 높아 보인다.

이러한 결과들은 다음과 같은 본 연구의 한계점을 고려하여 적용되어야 할 것이다. 먼저, 본 연구에서 설정한 하이퍼패러미터 세팅이 최적 조합이 아닐 가능성이 존재한다. 본 연구에서 에픽 수나 드롭아웃 및 배치 사이즈는 몇 가지 옵션 중에서 나은 것을 선택했고 이니셜라이저와 옵티마이저 및 학습률과 정규화 계수는 기존 ATSC 연구에서 많이 사용하는 값으로 설정했다. 결국 전역 탐색을 한 것이 아니기 때문에 더 좋은 하이퍼패러미터 조합이 존재할 가능성을 배제할 수 없다. 다음으로, QA 입력 타입이 NLI 입력 타입 보다 나은 것과 같은 결과가 데이터셋에 따라 달라지는 이유를 데이터셋의 특성과 연관지어 깊이 있게 분석하지 못하였다. 훈련 데이터셋의 속성당 속성 문장 수가 랩탑보다 레스토랑 데이터셋에서 훨씬 많기 때문에 같은 모델이라도 레스토랑 데이터셋에 대한 결과가 뚜렷하

게 우세하다고 할 수 있다. 또, 같은 레스토랑 데이터셋이라도 Restaurant-2014가 Restaurant-2016보다 속성당 속성 문장 수가 현저히 많기 때문에 Restaurant-2014의 결과가 더욱 좋다고 할 수 있겠다. 그런데 랩탑에서는 QA 입력 타입이 확연히 나은 결과를 제공하지만 레스토랑 데이터셋에서는 유의수준 10%에서도 성능 차이가 유의하지 않은 현상에 대해서는 확실한 원인을 규명하지 못했다. 물론 레스토랑 데이터셋에 대해서도 최고 성능의 모델은 QA 입력 타입인 것으로 미루어 이러한 경향성이 있는 건 분명해보인다. 이것은 QA 타입이 NLI 타입보다 더욱 자연스러운 문장의 연결 형태라는 점에서 BERT 모델의 메커니즘과 더욱 유사하기 때문일 수 있다. 마찬가지로, QA 타입에서 속성이 포함된 문장이 앞 또는 뒤에 오는 경우 모두 두 문장의 연결에 무리가 없기 때문에 QA 타입에서는 성능과 순서가 무관할 수 있을 것이다. 또, NLI 타입에서는 속성이 뒤에 오는 편이 의미적으로 좀 더 나은 연결이기 때문에 더욱 좋은 결과가 나왔을 것으로 추측된다.

한편, 본 연구를 통해 도출할 수 있는 이론적 시사점은 다음과 같다. 첫째, BERT와 같은 사전훈련 언어 모델에 대한 지속적인 연구가 이루어져야 할 것이다. QA와 NLI에서 이미 BERT를 적용한 모델들의 우수한 성능이 입증된 바 있지만, ACSC 분야에도 BERT를 적용할 때 기존의 LSTM 및 CNN 위주의 모델들보다 성능이 현저하게 향상되는 것을 본 연구에서 확인하였다. 다국어 버전으로 개발된 BERT를 조사와 어미가 발달한 교착어(agglutinative language) 특성을 반영하여 한국어에 특화한 KoBERT, KoGPT2, KorBERT, KR-BERT 등이 발표되고 있는 점은 상당히 고무적이다. 한국어는 동음이의어 비율이 높기 때문에 기존의 Word2Vec이나 Glove와 같이 하나의 단어에 하나의 단어 벡터가 매핑되는 모델보다는 BERT와 같이 하나의 단어라도 컨텍스트에 따라 다른 벡터 표현을 제공하는 모델로부터 누릴 수 있는 이점이 클 것으로 생각된다 (Park et al., 2018). 그런데 해외에서 BERT 및 후속모델들이 발표되면 국내에서는 이를 따라가면서 한국어에 응용하는 흐름으로 연구가 진행되고 있는데, 이를 바꾸어 주체적으로 선도해나갈 수 있다면 더욱 바람직할 것이다. 둘째, BERT와 같은 사전훈련 언어 모델을 적용할 때에는 해당 영역별로 차별화된 접근이 필요하다. ACSC에서는 다른 NLP 영역과 달리 속성을 나타내는 토큰들에 대한 출력 벡터를 분류 벡터에 반영하는 것이 유효한 것처럼, 영역별로 고유한 특성을 반영하여 모델 구조를 디자인하면 성능 향상을 기대할 수 있을 것이다. ACSC에서 QA나 NLI처럼 속성을 포함하는 문장을 생성하여 입력해주는 것도 특이한 점이라 하겠다. 셋째, 본 연구에서 사용한 모델 디자인 방법론을 ATSC를 포함한 다른 NLP 영역으로 확장하고, 본 연구에서 제시한 모델들을 한국어 데이터셋에 대해서도 적용해보는 연구가 필요하다. 각 도메인별로 유효하리라고 예상되는 입력 및 출력 디자인 옵션들을 선별하고 이들의 조합에 따라 모델들을 구현하여 성능 비교를 하되 옵션별 유효성을 검증해보는 방법론은 체계적이고 합리적인 접근을 가능하게 해준다. 한국어에 대한 적용을 위해서는 본 연구의 모델들을 굴절어(inflectional language)인 영어와 다른 특성을 갖는 한국어에 어떻게 적용해야 하고 어느 정도 성능을 기대할 수 있는지에 대한 연구가 이루어져야 할 것이다. 한국어에 특화된 BERT 응용 모델들을 ACSC에 적용한 연구는 아직 발표되지

않고 있다.

다음으로, 본 연구를 통해 도출할 수 있는 실무적 시사점은 다음과 같다. 먼저, 속성기반 감성분석 모델의 성능 향상으로 비즈니스 적용 가능성이 높아짐으로써 기존의 감성분석 적용 영역에서 더욱 정교한 분석 및 비즈니스 전략 도출이 가능해질 것으로 기대된다. 일반적인 문장 수준의 감성분석은 쇼핑몰 상품평 및 영화평 분석, 각종 이슈에 대한 국민여론 분석, 소비자 리뷰 감성분석 결과를 반영한 추천, SNS 및 뉴스 기사에 대한 감성분석 결과를 반영한 기업 및 주식시장 동향 분석 등의 다양한 분야에 적용되어 왔다 (Lee et al., 2018a; Lee et al., 2018b; Park and Kim, 2019). 이러한 도메인에서 속성 수준의 감성분석과 함께 암시적 속성까지 분석함으로써 새로운 인싸이트를 도출할 수 있는 가능성이 열리고 있는 것이다. 그런데 이러한 속성기반 감성분석의 비즈니스 가치를 실현하기 위해서는 데이터셋 구축이라는 문제를 해결하기 위한 논의 및 투자가 필요하다. 딥러닝 모델을 감성분석에 적용함으로써 기존의 사전 및 규칙 기반 감성분석에 수반된 상당한 양의 수작업 노력이 절감되었지만, 도메인에 적합한 데이터셋 구축은 여전히 적지 않은 부담으로 남아 있다. 현재 속성기반 감성분석 연구를 위한 영어 벤치마크 데이터셋은 본 연구에서 사용한 것들을 포함하여 몇 개 존재하지만 사이즈와 종류 면에서 많이 부족한 상황이다. 특히, 한국어에 대한 속성기반 감성분석 연구를 위한 공인된 데이터셋은 찾아보기 어렵다. 각 속성에 대한 다양한 감성 패턴을 포함하는 양질의 데이터셋을 효율적으로 구축하기 위한 방안을 모색해야할 시점인 것 같다.

마지막으로, 본 연구에서 얻은 결과들이 미래 속성기반 감성분석 연구에 실질적인 도움이 되고 삶의 질을 높여주는 다양한 지능적 서비스들이 창출되는 토대가 되기를 기대한다.

참고문헌(References)

Araque, O., G. Zhu, and C.A. Iglesias, “A Semantic Similarity-based Perspective of Affect Lexicons for Sentiment Analysis,” *Knowledge-Based Systems,* Vol.165, (2019), 346~359.

Chen, P., Z. Sun, L. Bing, and W. Yang, “Recurrent Attention Network on Memory for Aspect Sentiment Analysis,” *Proceedings of Empirical Methods on Natural Language Processing,* (2017), 463~472.

Davidov, D., O. Tsur, and A. Rappoport, “Enhanced Sentiment Learning Using Twitter Hashtags and Smileys,” *Proceedings of the 23rd International Conference on Computational Linguistics,* (2010), 241~249.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” (2018), arXiv:1810.04805.

Do, H.H., PWC. Prasad, A. Maag, and A. Alsadoon, “Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review,” *Expert Systems with Applications,* Vol.118, (2019), 272~299.

Dosoula, N., R. Griep, Rick den Ridder, R. Slangen, Ruud van Luijk, K. Schouten, and F. Frasincar, “Sentiment Analysis of Multiple Implicit Features per Sentence in Consumer Review Data,” *Proceedings of the 12th International Baltic Conference on Databases and Information Systems,* (2016), 241~254.

Dragoni, M., M. Federici, and A. Rexha, "An Unsupervised Aspect Extraction Strategy for Monitoring Real-time Reviews Stream," *Information Processing and Management,* (2018).

Gao, Z., A. Feng, X. Song, and X. Wu, "Target-Dependent Sentiment Classification with BERT," *IEEE Access,* Vol.7, (2019), 154290~154299.

Hai, Z., K. Chang, and J.-j. Kim, "Implicit Feature Identification via Co-occurrence Association Rule Mining," *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing,* (2011), 393~404.

Hannach, H.E. and M. Benkhalifa, "WordNet based Implicit Aspect Sentiment Analysis for Crime Identification from Twitter," *International Journal of Advanced Computer Science and Applications,* Vol.9, No.12(2018), 150~159.

Hoang, M., Oskar Alija Bihorac, and Jacobo Rouces. "Aspect-Based Sentiment AnalysisUusing BERT," *Proceedings of the 22nd Nordic Conference on Computional Linguistics,* (2019), 187~196.

Howard, J. and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference* (Long Papers), Vol.1, (2018), 328~339.

Khalil, T. and S. R. El-Beltagy, "NileTMRG at SemEval-2016 Task 5: Deep Convolutional Neural Networks for Aspect Category and Sentiment Extraction," *Proceedings of the 10th International Workshop on Semantic Evaluation,* (2016), 271~276.

Lee, S., B. Seo, and D. Park, "Development of Music Recommendation System based on Customer Sentiment Analysis," *Journal of Intelligence and Information Systems,* Vol. 24, No. 4 (2018a), 197~217.

Lee, S. W., C. W. Choi, D. S. Kim, W. Y. Yeo, and J. W. Kim, "Multi-Category Sentiment Analysis for Social Opinion Related to Artificial Intelligence on Social Media," *Journal of Intelligence and Information Systems,* Vol. 24, No. 4 (2018b), 51~66.

Li, X., L. Bing, W. Zhang and W. Lam, "Exploiting BERT for End-to-End Aspect-based Sentiment Analysis," *Proceedings of the 2019 EMNLP Workshop W-NUT: The 5th Workshop on Noisy User-generated Text,* (2019), 34~41.

Liu, B., *Sentiment Analysis and Opinion Mining, Springer, Berlin,* 2012.

Liu, Q., H. Zhang, Y. Zeng, Z. Huang, and Z. Wu, "Content Attention Model for Aspect Based Sentiment Analysis," *Proceedings of the 2018 World Wide Web Conference,* (2018), 1023~1032.

Ma, D., S. Li, X. Zhang, H. Wang, "Interactive Attention Networks for Aspect-Level *Sentiment Classification," Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence,* (2017), 4068~4074.

Park, H. and K. Kim, "Sentiment Analysis of Movie Review Using Integrated CNN-LSTM Model," *Journal of Intelligence and Information Systems,* Vol. 25, No. 4 (2019), 141~154.

Park, H., M. Song, and K. Shin, "Sentiment Analysis of Korean Reviews Using CNN: Focusing on Morpheme Embedding," *Journal of Intelligence and Information Systems,* Vol. 24, No. 2 (2018), 59~83.

Park, H., M. Song, and K. Shin, "Deep Learning Models and Datasets for Aspect Term Sentiment Classification: Implementing Holistic Recurrent Attention on Target-dependent Memories," *Knowledge-Based Systems,* Vol.187, (2020), 104825.

Peng, H., Y. Ma, Y. Li, and E. Cambria, "Learning Multi-grained Aspect Target Sequence for Chinese Sentiment Analysis," *Knowledge-Based Systems,* Vol.148, (2018), 167~176.

Peters, M., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep Contextualized Word Representations," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* Vol.1 (Long Papers), (2018), 2227~2237.

Pontiki, M., D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AlSmadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O.D. Clercq, V. Hoste, M. Apidianaki, X. Tannier, N.V. Loukachevitch, E.V. Kotelnikov, N. Bel, S. María J. Zafra, and G. Eryigit, "SemEval-2016 task 5: Aspect Based Sentiment Analysis," *International Workshop on Semantic Evaluation,* (2016). 19~30.

Pontiki, M., D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval 2015 task 12: Aspect Based Sentiment Analysis," *International Workshop on Semantic Evaluation,* (2015), 486~495.

Pontiki, M., D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 Task 4: Aspect Based Sentiment Analysis," *International Workshop on Semantic Evaluation,* (2014), 27~35.

Quan, C. and F. Ren, "Unsupervised Product Feature Extraction for Feature-oriented Opinion Determination," *Information Sciences,* Vol.272, (2014), 16~28.

Radford, A. and T. Salimans, "Improving Language Understanding by Generative Pre-Training," (2018).

Rajpurkar, P., J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000C Questions for Machine Comprehension of Text," (2016), arXiv:1606.05250.

Rietzler, A., S. Stabinger, P. Opitz, and S. Engl, "Adapt or Get Left Behind: Domain Adaptation through Bert Language Model Finetuning for Aspect-target Sentiment Classification," (2019), arXiv:1908.11860 [cs.CL].

Ruder, S., P. Ghaffari, and J.G. Breslin, "INSIGHT-1 at SemEval-2016 Task 5: Deep Learning for Multilingual Aspect-based Sentiment Analysis," *Proceedings of the 10th International Workshop on Semantic Evaluation,* (2016).

Schouten, K. and F. Frasincar, "Survey on Aspect-Level Sentiment Analysis," *IEEE Transactions on Knowledge and Data Engineering,* Vol. 28, No. 3(2016), 813~830.

Song, M., H. Park, and K. Shin, "Attention-based long short-term memory network using sentiment lexicon embedding for aspect-level sentiment analysis in Korean," *Information Processing & Management,* Vol.56, No.3(2019), 637~653.

Song, Y., J. Wang, Z. Liang, Z. Liu, T. Jiang, "Utilizing BERT Intermediate Layers for Aspect Based Sentiment Analysis and Natural

Language Inference," (2020), arXiv:2002.04815v1 [cs.CL].

Sun, C., L. Huang, and X. Qiu, "Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence," (2019), arXiv:1903.09588.

Tang, D., B. Qin, and T. Liu, "Aspect Level Sentiment Classification with Deep Memory Network," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* (2016), 214~224.

Tang, D., B. Qin, X. Feng, and T. Liu, "Effective LSTMs for Target-dependent Sentiment Classification," *International Conference on Computational Linguistics,* (2016), 3298~3307.

Tubishat, M., N. Idris, and M.A.M. Abushariah, "Implicit Aspect Extraction in Sentiment Analysis: Review, Taxonomy, Opportunities, and Open Challenges," *Information Processing and Management,* Vol.54, No.4(2018), 545~563.

Vaswani, A., 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,* Vol.2017-Decem, (2017), 5999~6009.

Wang, Y., M. Huang, L. Zhao and X. Zhu, "Attention-based LSTM for Aspect-level Sentiment Classification," *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* (2016), 606~615.

Xiaomei, Z., Y. Jing, Z. Jianpei, and H. Hongyu, "Microblog Sentiment Analysis with Weak Dependency Connections," *Knowledge-Based Systems,* Vol.142, (2018), 170~180.

Xu, H., B. Liu, L. Shu, and P. S. Yu, "Bert Post-training for Review Reading Comprehension and Aspect-Based Sentiment Analysis," (2019), arXiv:1904.02232.

Zeng, B., Heng Yang, Heng Yang, Ruyang Xu, Wu Zhou, Xuli Han, "LCF: A Local Context Focus Mechanism for Aspect-Based Sentiment Classification," *Applied Sciences,* Vol.9, No.16(2019), 3389.

Zhao, W., Z. Guan, L. Chen, X. He, D. Cai, B. Wang, and Q. Wang, "Weakly-Supervised Deep Embedding for Product Review Sentiment Analysis," *IEEE Transactions on Knowledge and Data Engineering,* Vol.30, No.1(2018), 185~197.

Zhu, J., H. Wang, M. Zhu, B.K. Tsou, and M. Ma, "Aspect-based Opinion Polling from Customer Reviews," *IEEE Transactions on Affective Computing,* Vol.2, (2011), 37~49.

Zhu, P., Z. Chen, H. Zheng, T. Qian, "Aspect Aware Learning for Aspect Category Sentiment Analysis," ACM *Transactions on Knowledge Discovery from Data,* Vol.13, No.6(2019).

APPENDIX

〈Table A-1〉 Results of 12 Implemented Models on Restaurant Datasets

Dataset	Model	Max_ Seq_ Len	Mean_ Acc	Mean_ Mac_ F1	Acc_ Std	Mac-F1_ std
Rest. 2014	Bert_Quest_Second_CLS_MEAN	97	90.62	84.10	0.44	0.84
	Bert_Quest_Second_MEAN	97	90.59	83.92	0.36	0.80
	Bert_Asp_Second_CLS_MEAN	91	90.58	84.32	0.54	1.14
	Bert_Asp_First_CLS_MEAN	91	90.45	83.87	0.71	1.21
	Bert_Asp_Second_MEAN	91	90.39	83.93	0.52	0.86
	Bert_Quest_First_CLS_MEAN	97	90.34	84.06	0.49	1.07
	Bert_Quest_First_MEAN	97	90.06	83.27	0.50	0.94
	Bert_Asp_Second_CLS	91	90.05	83.20	0.39	1.32
	Bert_Quest_First_CLS	97	89.95	83.21	0.30	0.68
	Bert_Asp_First_MEAN	91	89.87	82.93	0.41	0.91
	Bert_Asp_First_CLS	91	89.84	83.33	0.70	1.17
	Bert_Quest_Second_CLS	97	89.68	82.63	1.65	2.46
Rest. 2016	Bert_Quest_First_CLS_MEAN	105	89.09	73.55	0.37	3.86
	Bert_Asp_First_CLS_MEAN	99	88.96	72.02	0.47	2.66
	Bert_Quest_First_MEAN	105	88.85	72.96	0.34	2.77
	Bert_Asp_First_CLS	99	88.78	72.45	0.62	3.27
	Bert_Asp_Second_CLS	99	88.77	73.29	0.43	2.20
	Bert_Quest_Second_CLS_MEAN	105	88.77	72.48	0.54	2.11
	Bert_Asp_First_MEAN	99	88.66	72.56	0.37	2.25
	Bert_Asp_Second_CLS_MEAN	99	88.61	73.50	0.41	2.30
	Bert_Asp_Second_MEAN	99	88.61	72.46	0.41	3.96
	Bert_Quest_Second_MEAN	105	88.56	73.86	0.64	2.16
	Bert_Quest_First_CLS	105	88.54	72.91	0.37	3.15
	Bert_Quest_Second_CLS	105	87.58	70.19	1.85	3.91

〈Table A-2〉 Results of 12 Implemented Models on Laptop Datasets

Dataset	Model	Max_ Seq_ Len	Mean_ Acc	Mean_ Mac_ F1	Acc_ Std	Mac-F1_ std
Lap. 2015	Bert_Asp_Second_CLS_MEAN	92	84.46	69.06	0.48	1.71
	Bert_Quest_Second_MEAN	98	84.35	71.58	0.63	1.94
	Bert_Quest_First_MEAN	98	84.25	68.97	0.72	3.40
	Bert_Quest_First_CLS_MEAN	98	84.07	70.30	0.40	2.24
	Bert_Quest_Second_CLS_MEAN	98	84.04	69.63	0.65	3.50
	Bert_Quest_Second_CLS	98	83.99	70.24	0.74	3.15
	Bert_Asp_First_MEAN	92	83.94	70.18	0.83	3.11
	Bert_Asp_Second_MEAN	92	83.94	70.85	0.70	1.37
	Bert_Asp_Second_CLS	92	83.89	70.15	0.60	2.20
	Bert_Asp_First_CLS_MEAN	92	83.56	68.52	0.28	3.47
	Bert_Asp_First_CLS	92	83.56	67.93	0.75	3.42
	Bert_Quest_First_CLS	98	83.49	69.20	0.88	3.11
Lap. 2016	Bert_Asp_Second_CLS_MEAN	92	83.61	65.52	0.76	3.97
	Bert_Quest_First_CLS_MEAN	98	83.55	66.65	0.61	3.46
	Bert_Quest_First_MEAN	98	83.41	63.95	0.57	3.61
	Bert_Quest_Second_MEAN	98	83.40	66.38	0.46	3.30
	Bert_Quest_Second_CLS_MEAN	98	83.36	67.39	0.46	2.50
	Bert_Quest_First_CLS	98	83.33	65.95	0.64	3.11
	Bert_Asp_First_MEAN	92	83.29	63.68	0.49	3.83
	Bert_Asp_Second_MEAN	92	83.26	65.15	0.48	3.21
	Bert_Asp_First_CLS_MEAN	92	83.18	63.52	0.62	2.97
	Bert_Quest_Second_CLS	98	83.15	64.84	0.60	3.87
	Bert_Asp_First_CLS	92	83.15	64.43	0.59	3.66
	Bert_Asp_Second_CLS	92	83.07	65.71	0.74	3.20

Abstract

Aspect-Based Sentiment Analysis Using BERT: Developing Aspect Category Sentiment Classification Models

Hyun-jung Park* · Kyung-shik Shin**

Sentiment Analysis (SA) is a Natural Language Processing (NLP) task that analyzes the sentiments consumers or the public feel about an arbitrary object from written texts. Furthermore, Aspect-Based Sentiment Analysis (ABSA) is a fine-grained analysis of the sentiments towards each aspect of an object. Since having a more practical value in terms of business, ABSA is drawing attention from both academic and industrial organizations. When there is a review that says "The restaurant is expensive but the food is really fantastic", for example, the general SA evaluates the overall sentiment towards the 'restaurant' as 'positive', while ABSA identifies the restaurant's aspect 'price' as 'negative' and 'food' aspect as 'positive'. Thus, ABSA enables a more specific and effective marketing strategy.

In order to perform ABSA, it is necessary to identify what are the aspect terms or aspect categories included in the text, and judge the sentiments towards them. Accordingly, there exist four main areas in ABSA; aspect term extraction, aspect category detection, Aspect Term Sentiment Classification (ATSC), and Aspect Category Sentiment Classification (ACSC). It is usually conducted by extracting aspect terms and then performing ATSC to analyze sentiments for the given aspect terms, or by extracting aspect categories and then performing ACSC to analyze sentiments for the given aspect category.

Here, an aspect category is expressed in one or more aspect terms, or indirectly inferred by other words. In the preceding example sentence, 'price' and 'food' are both aspect categories, and the aspect category 'food' is expressed by the aspect term 'food' included in the review. If the review sentence includes 'pasta', 'steak', or 'grilled chicken special', these can all be aspect terms for the aspect category 'food'. As such, an aspect category referred to by one or more specific aspect terms is called an explicit aspect. On the other hand, the aspect category like 'price', which does not have any specific aspect terms

* HI AI & Computing Research Center, Korea University
** Corresponding author: Kyung-shik Shin
School of Business, Ewha Womans University
52 Ewhayeodae-gil, Seodaemun-gu, Seoul, 03760, Korea
Tel: +82-2-3277-2799, Fax: +82-2-3277-2766, E-mail: ksshin@ewha.ac.kr

but can be indirectly guessed with an emotional word ‘expensive,’ is called an implicit aspect. So far, the ‘aspect category’ has been used to avoid confusion about ‘aspect term’. From now on, we will consider ‘aspect category’ and ‘aspect’ as the same concept and use the word ‘aspect’ more for convenience. And one thing to note is that ATSC analyzes the sentiment towards given aspect terms, so it deals only with explicit aspects, and ACSC treats not only explicit aspects but also implicit aspects.

This study seeks to find answers to the following issues ignored in the previous studies when applying the BERT pre-trained language model to ACSC and derives superior ACSC models. First, is it more effective to reflect the output vector of tokens for aspect categories than to use only the final output vector of [CLS] token as a classification vector? Second, is there any performance difference between QA (Question Answering) and NLI (Natural Language Inference) types in the sentence-pair configuration of input data? Third, is there any performance difference according to the order of sentence including aspect category in the QA or NLI type sentence-pair configuration of input data?

To achieve these research objectives, we implemented 12 ACSC models and conducted experiments on 4 English benchmark datasets. As a result, ACSC models that provide performance beyond the existing studies without expanding the training dataset were derived. In addition, it was found that it is more effective to reflect the output vector of the aspect category token than to use only the output vector for the [CLS] token as a classification vector. It was also found that QA type input generally provides better performance than NLI, and the order of the sentence with the aspect category in QA type is irrelevant with performance. There may be some differences depending on the characteristics of the dataset, but when using NLI type sentence-pair input, placing the sentence containing the aspect category second seems to provide better performance. The new methodology for designing the ACSC model used in this study could be similarly applied to other studies such as ATSC.

Key Words : Aspect-Based Sentiment Analysis, ABSA, Aspect Category Sentiment Classification, BERT, NLP

Received : September 23, 2020 Revised : November 2, 2020 Accepted : November 23, 2020
Corresponding Author : Kyung-shik Shin

저 자 소 개

박현정

현재 고려대학교 Human-inspired 복합지능연구센터 연구교수로 재직 중이다. KAIST 경영과학과에서 학사와 석사 학위를, 서울대학교 경영학과에서 경영정보시스템 전공으로 박사 학위를 취득하였다. 주요 연구분야는 비즈니스 애널리틱스(Business Analytics), 데이터 마이닝, 인공지능 및 자연어처리, 빅데이터 분석, 소셜 네트워크 분석, 가상화(Virtualization) 및 가상 협업(Virtual Collaboration) 등이다.

신경식

현재 이화여자대학교 경영대학 경영학부 교수로 재직 중이다. 연세대학교 경영학과를 졸업하고, 미국 George Washington University에서 MBA, KAIST에서 경영공학 Ph.D.를 취득하였다. 주요 연구분야는 데이터 마이닝과 비즈니스 인텔리전스, 빅데이터 분석, 비즈니스 애널리틱스(Business Analytics), 인공지능 응용과 지식공학, 가상화(Virtualization) 및 가상 협업(Virtual Collaboration) 등이다.