JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 25, No. 5, pp. 1159-1166, May. 2024

Check for updates

이커머스 도메인에서의 동일 태그 교체 데이터 증강 기법을 활용한 개체명 인식

장 동 호[1] · 부 석 준[2] · 서 영 건[2*]
[1]경상국립대학교 컴퓨터공학과 학생
[2]경상국립대학교 컴퓨터공학과 교수

Named Entity Recognition in E-commerce Domain using the Same-Tag Replacement Data Augmentation Technique

Dong-Ho Jang[1] · Seok-Jun Buu[2] · Yeong Geon Seo[2*]

[1]Student, Department of Computer Engineering, Gyeongsang National University, Jinju 52828, Korea
[2]Professor, Department of Computer Engineering, Gyeongsang National University, Jinju 52828, Korea

[요 약]

데이터 증강은 한국어 NER 분야에서 도메인 특화 데이터 부족으로 발생하는 어려움을 해결하기 위한 중요한 전략이다. 본 연구에서는 이커머스 도메인에서 한국어 개체명 인식 모델을 향상시키기 위한 데이터 증강 방법으로 '음절 단위 문장 BIO 태깅 및 동일 태그 교체(Same-Tag Replacement)' 알고리즘을 제안하고 실험하였다. 이 방법은 한국어 NER 데이터셋에서 문장을 음절 단위로 분리하고 BIO 태그를 부착한 뒤, 동일한 개체 유형에 속하는 단어를 무작위로 교체하여 데이터를 증강한다. 실험 결과, 작은 데이터셋(N=500)의 데이터를 증강했을 때 weighted-average f1-score가 최대 50%까지 개선된 것을 확인하였다. 이는 이커머스 도메인에서 자연어 처리 모델 성능 향상을 위한 실용적이고 효과적인 전략으로 주목받을 것으로 기대된다.

[Abstract]

Data augmentation is an effective strategy to address the challenge of insufficient domain-specific data in the field of Korean named entity recognition(NER). In this study, we developed a method for improving the Korean NER model in the e-commerce domain using character-level begin inside outside(BIO) sentence tagging and a same-tag replacement algorithm. This method involves splitting sentences into character units and then attaching BIO tags. Subsequently, words belonging to the same entity type are randomly replaced to augment the data. Experimental results revealed that the weighted-average F1-score improved by up to 50% when small datasets(N=500) were augmented. This method is expected to be recognized as a practical and effective strategy for enhancing the performance of language processing models in the e-commerce domain.

색인어 : 데이터 증강, NER, 동일테그교체, 단어 무작위 교체, 자연어 처리

Keyword : Data Augmentation, NER, Same-Tag Replacement, Random Word Replacement, Natural Language Processing

http://dx.doi.org/10.9728/dcs.2024.25.5.1159

Received 21 March 2024; **Revised** 12 April 2024
Accepted 12 April 2024

***Corresponding Author; Yeong Geon Seo**

Tel: +82-55-772-1392
E-mail: young@gnu.ac.kr

Ⅰ. 서 론

현대의 디지털 시대에서는 자연어 처리 기술이 빠르게 발전하면서, 다양한 분야에서 이를 응용하는 연구가 활발히 진행되고 있다. 특히, 자연어 처리 기술은 상호작용성이 높은 인공지능 챗봇의 구축에 큰 영향을 미치고 있다. 이러한 챗봇은 사용자들과 원활한 의사소통을 가능케 하며, 다양한 업무나 서비스 분야에서 적용되고 있다. 이와 함께, 쇼핑몰 분야에서는 지능형 챗봇을 통한 상품 추천, 주문 처리, 문의 응대 등이 점차 중요한 역할을 차지하고 있다. 이러한 지능형 챗봇이 정확하게 동작하려면, 주어진 문장에서 중요한 정보를 정확히 추출해 내는 개체명 인식(NER : Named Entity Recognition) 모델[1]-[3]이 필수적이다. 하지만, 이커머스 도메인의 특성상 생성 언어모델을 파인튜닝하기 위한 고품질 학습데이터를 확보하는 것은 쉬운 일이 아니다. 다양한 상품, 서비스, 그리고 사용자의 다양한 언어와 표현이 이뤄지는 이커머스 도메인에서는 많은 양의 도메인 특화 데이터가 필요하게 되는데, 이는 실제로 구하기가 어렵고 비용이 많이 드는 작업 중 하나이다.

데이터셋이 부족한 경우, 개체명 인식 성능이 떨어질 수 있다. 이러한 상황에서는 데이터를 인위적으로 증가시키는 방법이 유용하게 활용될 수 있다. 기존의 한정된 데이터셋을 다양한 방법으로 확장함으로써 모델이 더 다양한 상황에 대응할 수 있게 도와준다. 본 연구에서 제안하는 '음절 단위 문장 BIO 태깅 및 동일 태그 교체(Same-Tag Replacement)' 알고리즘은 데이터를 효과적으로 증강시키면서 모델의 성능을 향상시킬 수 있는 방법을 제시한다.

이 알고리즘은 한국어 NER 데이터셋에서 문장을 음절 단위로 나눈 뒤 BIO(Begin Inside Outside) 태그를 부착하고, 동일한 개체 유형에 속하는 단어를 무작위로 교체하여 데이터를 증강하는 방법이다. 예를 들어, “삼성 노트북 추천”이라는 문장에서 “삼성”이 LOC(LOCation) 태그를 갖고 있을 경우, 전체 학습 데이터에서 무작위로 “삼성”과 같은 LOC 태그를 갖는 “엘지”를 선택하여 교체하여 증강한다. 이 방법을 통해 데이터셋의 양을 증가시키면서도 원래의 의미를 유지할 수 있다.

본 연구를 통해 제안된 데이터 증강 방법이 데이터가 부족한 상황에서도 모델이 효과적으로 학습되는지 확인할 것이다. 또한, 이 방법이 모델의 일반화 능력을 향상시키는 데에 어떠한 영향을 미치는지를 정량적으로 측정하고자 한다. 논문의 구성은, 먼저 자연어 처리 분야에서 선행 연구된 데이터 증강 기법을 소개한다. 다음으로, 제안하는 데이터 증강 방법에 대한 상세한 설명과 적용 가능성을 논의한다. 마지막으로, 결론에서는 제안 방법의 효과와 한계, 그리고 향후 연구 방향에 대해 논의할 것이다.

Ⅱ. 관련 연구

데이터 증강 기법은 주어진 데이터 양이 제한적인 상황에서 학습에 필요한 다양한 데이터를 확보하기 위해 인위적인 변화를 주는 방법론이다. 자연어 처리 분야에서는 문장 내의 단 하나의 단어만 변경되어도 전체 문장의 의미가 크게 변할 수 있다. 또한, 문장 내의 단어 배열 순서가 변경되면 문법적인 오류가 발생할 수 있다. 단어의 변경이나 문장의 재배열은 모델이 학습한 문맥과 의미를 왜곡할 수 있기 때문에, 데이터 증강 시에는 원래 문장의 의미를 유지하면서 새로운 데이터를 생성하는 것이 중요하다. 특히, 개체명 인식과 같은 작업에서는 개체의 신뢰성과 일관성을 유지하는 것이 핵심적이다. 이런 어려움에도 불구하고 자연어 처리 분야에서의 데이터 증강 기법은 간단한 방법부터 Transformer 모델[4]을 활용하는 방법까지 다양하게 연구되어왔다.

2-1 간단한 데이터 증강 방법

차원 EDA(Easy Data Augmentation)은 간단하면서도 효과적인 데이터 증강 방법 중 하나로 알려져 있다. [5]는 특별한 언어 모델을 사용하지 않고, 단지 원본 데이터를 이용하는 증강 기법을 사용하여 텍스트 분류 문제에서 효과를 입증하였다. 표 1은 기존의 문장에서 무작위로 단어를 추가, 삭제, 변경, 혹은 교체하는 방법을 설명한 것으로, 국내외에서도 이를 통한 연구가 많이 진행되었다. 실제로, [6]과 [7]은 적은 양의 범용 NER 데이터셋에서 EDA 증강 기법을 적용한 데이터로 학습하였을 때, 큰 데이터셋을 사용한 것만큼 성능이 향상될 수 있음을 확인하였다.

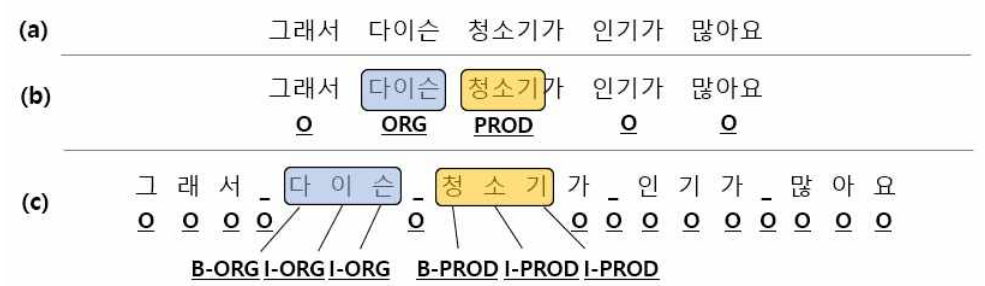
표 1. 한글에 적용된 EDA 기술과 예
Table 1. EDA techniques and examples applied to Korean language

Type	Sentence
None	나는 자전거 타는 것을 좋아한다.
SR	나는 오토바이 타는 것을 좋아한다.
RI	나는 자전거 타는 것을 오늘 좋아한다.
RS	나는 좋아 타는 것을 자전거한다.
RD	자전거 타는 것을 좋아한다.

*We don't translate into English for the accurate conveyance of the word's meaning

2-2 역 번역

역 번역(Back translation)은 언어 간 번역을 통해 데이터를 증강하는 방법으로, 초기 텍스트를 목표 언어로 번역한 다음 해당 번역된 텍스트를 다시 원래 언어로 번역하는 과정을 포함한다. [8]은 이 과정을 통해 원래의 텍스트와 완전히 다른 형태의 문장을 생성하여, 모델이 더 많은 다양성과 언어적 특성을 학습할 수 있게 도움을 주었다. 이러한 방법은 특히 제한된 데이터셋에서 모델의 성능을 향상하기 위한 효과적인 전략 중 하나로 사용되고 있다. 표 2는 역 번역의 예를 보이고 있다.



*We don't translate into English for the accurate conveyance of the word's meaning

그림 1. BIO 태깅의 예. (a) 원 데이터 (b) BIO 태그된 데이터 (c) 음절 단위로 분할한 후 BIO 태그된 데이터

Fig. 1. Example of BIO tagging (a) original data, (b) bio tagged data, (c) BIO tagged data with algorithm for syllable-level segmentation

표 2. 한글에 적용된 역번역 예

Table 2. Examples of back translation applied to Korean language

Type	Sentence
Original sentence	The weather is beautiful today, and I'm enjoying a walk in the park.
Translation sentence	오늘 날씨가 아주 좋아서, 나는 공원에서 산책을 즐기고 있어요.
Back translation sentence	Today's weather is so nice, and I'm enjoying a walk in the park.

*We don't translate into English for the accurate conveyance of the word's meaning

2-3 딥러닝 모델 사용

딥러닝 모델을 활용한 데이터 증강은 기존의 모델 구조를 활용하여 새로운 훈련 데이터를 생성하는 방법이다. 이러한 방법은 주로 생성 모델(Generative model)을 활용하여 다양한 변형을 가진 텍스트를 생성하는 데 적용된다. 특히, GPT(Generative Pre-trained Transformer)와 같은 사전 훈련된 언어 모델을 이용하여 훈련 데이터를 보강하는 방법이 널리 사용되고 있다. 주어진 텍스트에 대해 GPT 모델을 사용하여 문장을 자동으로 생성하고, 이를 훈련 데이터에 추가함으로써 데이터를 보강할 수 있다. 이를 통해 모델은 보다 다양한 문맥과 표현을 학습하게 되어 일반화 성능을 향상시킬 수 있다. 실제로, [9]는 GPT-3와 같은 대규모 언어 모델을 활용하여 현실적인 텍스트 샘플을 생성하고 이를 전이 학습에 활용하는 데이터 증강 기술을 제안했고, 다양한 분류 작업에서 기존의 텍스트 증강 기술보다 더 뛰어난 성능을 보여주었다.

Ⅲ. 동일 태그 교체 데이터 증강 기법

3장에서는 음절 단위 문장의 BIO 태킹 기법을 제안하고 원래 문장의 의미를 유지하면서 새로운 데이터를 생성하는 동일 태그 교체 알고리즘을 제안한다.

3-1 음절 단위 문장 BIO 태깅

본 연구에서는 BIO 태그를 사용하여 개체명을 예측하는 지도학습 기반의 개체명 인식 방법론을 사용하였다. 그림 1은 모델 학습에 사용된 BIO 태그의 예시로, (a)는 원본 데이터, (b)는 BIO 태그가 적용된 형태, 그리고 (c)는 문장을 음절 단위로 분할한 뒤 BIO 태그를 적용하는 알고리즘을 거친 문장이다. 문장을 음절 단위로 분리해주는 이유는 각 음절이 어떤 유형의 개체명에 속하는지 정확하게 레이블링해주기 위함이다. 예를 들어, "삼성 전자"와 "삼성전자"라는 단어가 나왔을 때, 띄어쓰기 여부에 따라 BIO 태깅이 제대로 되지 않는 문제가 발생할 수 있다. 그러나 음절 단위로 분리하면 띄어쓰기에 상관없이 각 음절이 정확하게 레이블링 되어, 모델이 개체명을 더욱 정확하게 학습할 수 있게 된다. 다음 알고리즘은 이러한 음절 단위 문장 BIO 태깅 알고리즘의 의사코드를 나타내고 있다.

```
function DataPreprocessing(df )
    texts ← []
    tags ← []
    for row in df do
        text ← row['']
        bio_tags ← { 'MNY': row[''], 'NOH': row[''], 'PNT':
         row[''], 'ORG': row[''], 'DAT': row[''], 'PROD': row[''] }
        for entit_ type, values in bio_tags do
            for value in values do
                if value ≠ None then
                    entity start ← find start position(text, value)
                    if entity start ≠ −1 then
                        entity end ← entity start + length(value)
                        tag list[entity start : entity end] ←
    ['B-'+entity_type] + ['I-' + entity_type] * (length(value) - 1)
                    end if
                end if
            end for
        end for
        texts.append(tokenize(text))
        tags.append(tag_list)
    end for
    return texts, tags
end function
```

표 3. 데이터 증강의 예
Table 3. Example of data augmentation

Type	Example															
Original Sentence	다	이	슨	_	청	소	기	가	_	인	기	가	_	많	아	요
	B-LOC	I-LOC	I-LOC	O	B-PROD	I-PROD	I-PROD	O	O	O	O	O	O	O	O	O
Label-wise Tag Replacement	엘	지	_	냉	장	고	가	_	인	기	가	_	많	아	요	
	B-LOC	I-LOC	O	B-PROD	I-PROD	I-PROD	O	O	O	O	O	O	O	O	O	

*We don't translate into English for the accurate conveyance of the word's meaning

3-2 동일 태그 교체

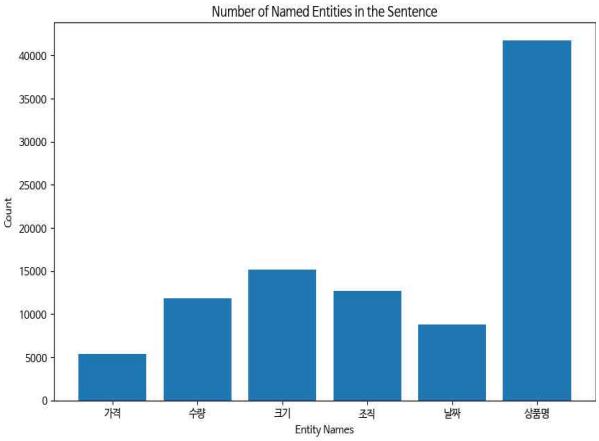
제안 방법에서는 데이터 증강 시 원래 문장의 의미를 유지하면서 새로운 데이터를 생성하기 위해 '동일 태그 교체 알고리즘을 사용하였다. 동일 태그 교체는 학습데이터 내 태그를 활용하여 문장에서 태그를 선택하고, 선택된 태그와 일치하는 태그를 학습데이터 내에서 선택하여 교체하는 방법이다. 문장에서 선택된 태그는 'O' 태그를 제외한 모든 태그를 대상으로 하며, 전체 학습데이터에서 선택되는 태그는 임의로 선택한다. 예를 들면, 표 3의 원본 문장에서 '다이슨'이라는 단어와 '청소기'가 임의로 선택되면, 전체 학습데이터에서 '다이슨'과 같은 조직(ORG) 태그를 가지는 '엘지'를 선택하여 교체하고, '청소기'와 같은 상품(PROD) 태그를 가지는 '냉장고'를 선택하여 교체한다. 이를 통해서 "그래서 엘지 냉장고가 인기가 많아요"와 같은 데이터가 생성된다. 다음 알고리즘은 '동일 태그 교체' 알고리즘의 의사코드를 나타내고 있다.

```
1:function Same_Tag_Replacement(text, tags, ratio)
2:     argument_text ← text.copy()
3:     argument_tags ← tags.copy()
4:     num_tokens ← sum(1 for tag in tags
                    if tag.startswith('B-'))
5:     num_replacement_tokens
                    ← max(1,int(num_tokens ×ratio))
6:     non_O_tags ← [tag for tag in argument_tags
                        if tag.startswith('B-')]
7:     idx_pairs = random.choice(list(combinations
         (range(len(non_O_tags)), num_replacement_tokens)))
8:     tag_pair = [non_O_tags[idx] for idx in idx_pairs]
9:     for idx, tag in idx_pairs, tag_pair do
10:       entity_type ← tag.split('-')[1]
          start_idx ← argument_tags.index(tag)
11:       end_idx ← idx +1
12:       while end_idx < len(argument_tags) and
                    argument_tags[end_idx].startswith('I-') do
13:           end_idx ← end_idx + 1
14:       end while
15:       del argument_text[start_idx : (end_idx - 1)]
16:       del argument_tags[start_idx : (end_idx - 1)]
17:       random entity ←
           random.choice(entity_dict.get(entity_type, ['UNK']))
18:     argument_text[start_idx : start_idx] ←
                    list(random entity.replace(' ', '_')
19:     argument_tags.insert(start_idx, f'B - entity_type')
20:     for _ in len(random_entity) - 1 do
21:         argument_tags.insert(start_idx +1,
                    f'I-entity type')
22:     end for
23:   end for
24:   return argumented_text, argument_tags
25:end function
```

Ⅳ. 성능 평가 및 결과

4-1 데이터셋

본 장에서는 AI-Hub에서 수집한 소상공인 고객 주문 질의응답텍스트(https://www.aihub.or.kr/)에 대한 증강 기법 실험 결과를 포함하였다. 이 데이터는 소상공인 상점에서 고객 질문에 대한 대화를 녹취하여 얻은 음성 파일을 기반으로 한다. 가구/인테리어, 건강, 디지털 가전, 뷰티 등 다양한 카테고리가 존재하지만, 본 논문에서는 디지털 가전 훈련 데이터셋 만을 사용하여 실험하였다. 디지털 가전 데이터는 총 75,000개의 문장으로 구성되어 있으며, 가격(MNY), 수량(NOH), 크기(PNT), 조직(ORG), 날짜(DAT), 상품명(PROD)과 같이 다양한 엔티티 범주에 대한 태깅이 이루어져 있다(참고: 그림 2).



*We don't translate into English for the accurate conveyance of the word's meaning

그림 2. 문장 내의 개체명의 개수
Fig. 2. Number of named entities in the sentences

그러나 데이터의 품질 문제로 인해 태그가 전혀 되어 있지 않거나 매우 부족한 일부 문장을 모두 사용하는 대신, 최소한 2개 이상의 태그가 지정된 문장만을 추출하여 약 50,000개의 문장을 사용하였다. 또한 원본 데이터 크기에 따른 성능을 확인하기 위해 데이터를 소(Small), 중(Medium), 대(Large)로 나누어 각각 5회 반복 실험을 진행하여 각 모델이 개체명 인식에서 어떤 성능 향상을 보이는지 확인하였다.

4-2 평가 지표

개체명 인식 문제에서 모델의 성능을 평가하기 위해, 개체명은 해당 개체명에 속하는 모든 단어가 올바르게 예측되어야 한다. 이를 확인하기 위해 본 논문에서는 정밀도(precision), 재현율(recall), 그리고 f1-score를 사용했다. 표 4는 본 실험에서 사용하는 전체 데이터셋의 태그 수이다. 표에서 보는 것과 같이 실험에 사용하는 데이터셋의 태그 불균형이 심하여, 본 실험에서는 weighted-average f1-score 값으로 성능을 평가하였다. 이를 계산하기 위해, Seqeval 파이썬 라이브러리를 사용했다(Nakayama). Seqeval을 사용하기 위해, 우리는 예측된 값들의 리스트와 실제 값들의 리스트를 제공하기만 하면 된다. Seqeval이 사용하는 식은 아래와 같으며, 여기서 tp는 true positives, fp는 false positives, fn은 false negatives를 나타낸다.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$F1-Score = 2 * \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}} \quad (3)$$

4-3 기본 모델 생성

실험을 위해 KoELECTRA 모델[10]을 기본 모델로 선택하였다. KoELECTRA 모델은 transformer 아키텍처를 기반으로 하며, 다양한 하이퍼 파리미터와 미세 조정을 통해 한국어 특화 모델로 발전시켰다. 학습 데이터셋의 특성을 고려하여 모델의 성능을 극대화하기 위해, 다양한 실험을 통해 최적의 하이퍼 파라미터를 탐색하였다. 표 5는 사용된 기본 모델의 주요 하이퍼 파라미터이다.

표 4. 훈련 데이터셋에 사용된 태그의 수
Table 4. Number of tags in the training dataset

Entity	Tag	Count
Date	B-DAT	8,058
	I-DAT	24,919
Product Price	B-MNY	5,329
	I-MNY	18,841
Product Quantity	B-NOH	12,134
	I-NOH	16,820
Product Size	B-PNT	15,290
	I-PNT	50,546
Brand	B-ORG	10,308
	I-ORG	22,689
Product Name	B-PROD	44,857
	I-PROD	168,785

표 5. 한글에 적용된 EDA 기술과 예
Table 5. EDA techniques and examples applied to Korean language

Type	Sentence
Model	KoELECTRA
Learning Rate	3e-5
Weight Decay	0.01
Num Train Epochs	5
Train Batch Size	8

4-4 실험 결과

실험에서 KoELECTRA 모델을 기반으로 한 방법론을 적용하여 다양한 크기의 학습 데이터셋에 대해 5회 반복 실험을 진행하였고, 그 결과의 평균과 표준 편차를 표 6에서 확인할 수 있다. 표에서 밑줄 친 굵은 글씨는 제시한 방법론을 적용하지 않은 데이터셋 과의 성능 차이를 의미한다. 실험 결과,

표 6. 데이터 증강 기법을 사용한 KoELECTRA의 성능. N은 데이터셋 크기
Table 6. Performance of KoELECTRA model by using data augmentation method. N is the size of the dataset.

Method	Weighted Average F1-Score(%)								
	KoELECTRA-Small			Dialog-KoELECTRA-Small			KoELECTRA-Base		
	Small (N=500)	Medium (N=2,500)	Large (N=5,000)	Small (N=500)	Medium (N=2,500)	Large (N=5,000)	Small (N=500)	Medium (N=2,500)	Large (N=5,000)
Without Data Augmentation	0 ± 0	45.3 ± 0.3	60.4 ± 0.7	20.1 ± 0.3	54.6 ± 2.6	64.5 ± 2.4	41.2 ± 0.1	68.1 ± 2.3	73.2 ± 2.5
Same-Tag Replacement	49.2 ± 2.1 (+49.2)	64.2 ± 3.6 (+18.9)	63.5 ± 2.5 (+3.1)	50.2 ± 2.9 (+30.1)	58.2 ± 3.1 (+3.6)	65.7 ± 2.8 (+1.2)	60.6 ± 0.8 (+19.4)	68.3 ± 2.1 (+0.2)	68.7 ± 2.2 (-4.5)

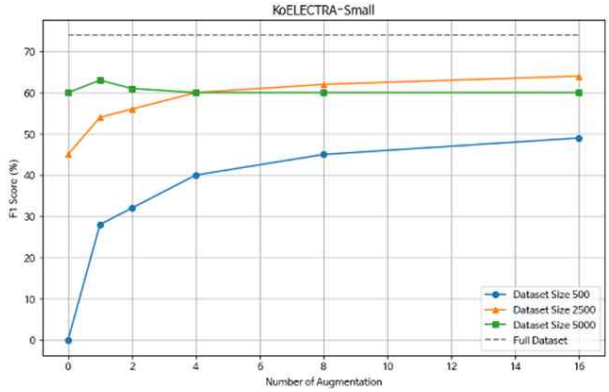


그림 3. 데이터 증강 개수에 따른 KoELECTRA-small의 성능
Fig. 3. Performance of KoELECTRA-small by number of data augmentation

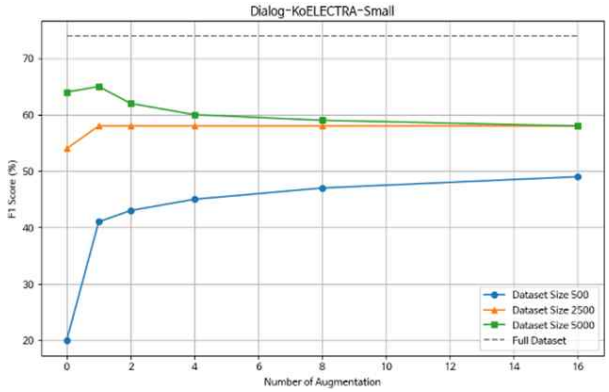


그림 4. 데이터 증강 개수에 따른 대화 KoELECTRA-small의 성능
Fig. 4. Performance of dialog KoELECTRA-small by number of data augmentation

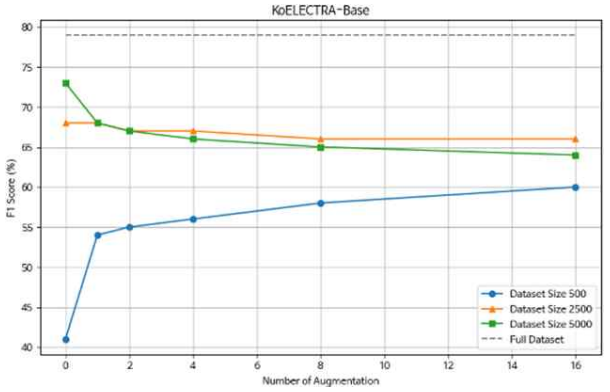


그림 5. 데이터 증강 개수에 따른 KoELECTRA-base의 성능
Fig. 5. Performance of KoELECTRA-base by number of data augmentation

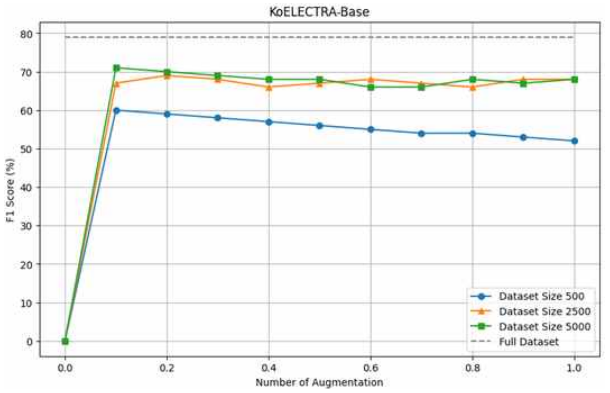


그림 6. 데이터 교체 개수에 따른 KoELECTRA 모델의 성능
Fig. 6. Performance of KoELECTRA models by number of data replacement

KoELECTRA-Small과 Dialog-KoELECTRA-Small 모델은 비교적 가벼운 모델임에도 불구하고 작은 데이터셋에서 데이터를 증강했을 때 큰 폭으로 모델의 성능이 개선되었다. 반면에 KoELECTRA-Base 모델은 비교적 성능 향상이 제한적인 것으로 나타났다. 이어서, 실험에서 적용된 데이터 증강의 수를 조절하여 모델의 성능에 미치는 영향을 관찰한 결과를 그림 3에서 확인할 수 있다. 실험 결과, 세 모델 모두 작은 규모의 데이터셋(N=500)에서는 데이터 증강이 모델의 성능을 크게 향상시켰다. 이는 데이터셋이 작을수록 모델이 학습할 수 있는 문맥이 제한되어 있기 때문에 데이터 증강이 모델의 성능 향상에 긍정적인 영향을 미쳤다는 것을 알 수 있다.

그러나 특정 개수 이상으로 데이터 증강을 적용할 경우, 모델의 성능 향상이 감소하는 경향이 나타났다. 이는 데이터 양이 증가함에 따라 모델이 학습할 수 있는 유의미한 패턴이 이미 충분히 반영되었거나 잘못된 학습 문장이 생성되어 성능에 부정적인 영향을 미칠 가능성이 있음을 시사한다.

다음으로, 데이터 교체의 수를 조절하여 모델의 성능에 미치는 영향을 살펴보았다. 실험 결과, 큰 규모의 데이터셋(N=2500, 5000)에서는 데이터 교체 비율이 성능에 큰 영향을 미치지 않았지만 작은 규모의 데이터셋(N=500)에서는 데이터 교체 비율이 커질수록 성능이 떨어지는 현상이 발견되었다. 이러한 결과는 작은 규모의 데이터셋에서는 데이터 교체의 양이 증가함에 따라 모델이 학습한 유용한 정보가 손실되어 성능이 저하된다는 것을 시사한다. 반면에 큰 규모의 데이터셋에서는 충분한 정보가 포함되어 있기 때문에 데이터 교체의 영향이 미미한 것으로 나타났다. 따라서, 작은 규모의 데이터셋을 다룰 때에는 신중한 데이터 교체 비율 설정이 중요하다는 것을 알 수 있다.

4-5 오류 분석

데이터를 증강했을 때 모델의 성능이 오히려 떨어지는 원인을 파악하기 위해, 오류 분석을 수행했다. 오류 분석은 모델의 강점과 약점을 파악하는 유용한 도구로 사용된다. 먼저, 모델이 어떤 토큰에서 가장 약점을 보이는지를 파악하기 위해 토큰 분류의 오차 행렬을 시각화했다(그림 7). 분석 결과, 모델은 B-DAT의 평균 손실이 0.21로 가장 높았고, 그 다음으로 B-ORG가 평균 손실 0.19로 두 번째로 높았다. 이는 모델

표 7. 저품질의 데이터셋의 예
Table 7. Examples of low-quality datasets.

Type	Example															
Original Sentence	1	0	0	인	치	_	해	상	도	로	_	시	청	_	가	능
True Label	O	O	O	O	O	O	O	O	B-PROD	I-PROD	I-PROD	I-PROD	O	O	O	O
Predicted Label	B-PNT	I-PNT	I-PNT	I-PNT	I-PNT	O	O	O	O	O	O	O	O	O	O	O

*We don't translate into English for the accurate conveyance of the word's meaning

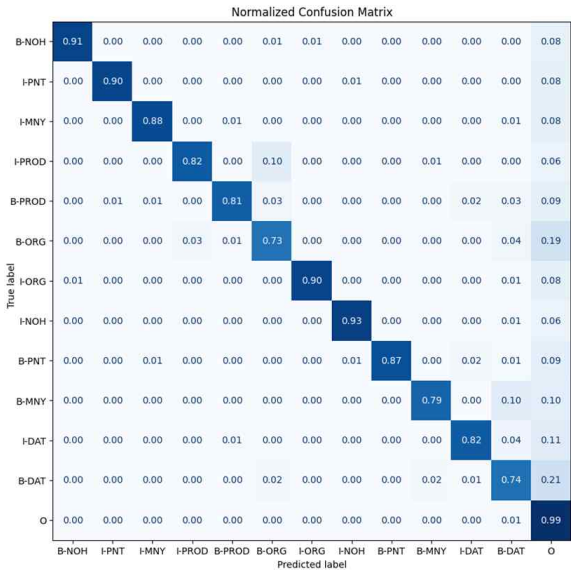


그림 7. 정규화된 오차 행렬
Fig. 7. Normalized confusion matrix

이 날짜(DAT)와 브랜드명(ORG)의 시작 부분을 결정하는 데 어려움을 겪고 있음을 나타낸다.

더 자세한 원인을 분석하기 위해, 높은 손실을 내는 시퀀스들을 알아본 결과 원본 데이터의 일관성 및 정확성 부족으로 인한 것으로 보인다. 대표적인 예로, 표 7을 살펴보면 “100인치 해상도로 시청 가능”이라는 문장에서 모델은 “100인치”라는 단어를 크기(PNT)로 정확하게 예측했지만 실제로 원본 라벨에서는 ‘O’ 태그가 부착되어 있어, 모델의 성능이 낮게 나오는 것을 확인하였다. 또한, 일부 상품명이 모델의 학습에 혼동을 주고 있는 것으로 확인되었다. 예를 들어, 이 문장에서 “도로 시” 라는 상품명이 모델에게 혼란을 일으키고 있는 것으로 나타났다. 이로 인해, 교체되지 말아야 할 단어까지 알고리즘에 의해 교체되어 모델이 잘못된 문장을 학습하는 것으로 나타났다.

V. 결 론

본 연구에서는 '음절 단위 문장 BIO 태깅 및 동일 태그 교체 알고리즘을 소개하고, 다양한 규모의 학습 데이터셋에서의 성능 평가를 수행하였다. 실험 결과, 원본 데이터셋의 일관성 및 정확성 부족으로 인해 데이터를 증강했을 때 모델의 성능 저하를 불러일으켰다. 향후 연구에서는 다음과 같은 방향으로 발전할 수 있을 것으로 보인다.

첫째로, 전처리 및 데이터 품질 관리에 중점을 두어, 특히 작은 규모의 데이터셋을 증강하는 경우에도 모델이 효과적으로 학습할 수 있도록 데이터를 정제하고 일관성을 유지하는 작업이 필요하다. 둘째로, 이전의 오류 분석에서 언급된 특정 도메인의 단어 문제에 대한 해결책을 찾아내고, 해당 도메인에 특화된 전처리 기법을 개발하여야 한다. 그렇지 않으면, 교체되지 말아야 할 단어까지 알고리즘에 의해 교체되어 모델이 잘못된 문장을 학습할 가능성이 커진다. 만약, 이러한 문제들이 해결된다면 정확한 개체명 인식이 중요한 이커머스 도메인에서 자연어 처리 모델의 성능 향상에 큰 도움이 될 것으로 기대된다.

감사의 글

이 연구는 과학기술정보통신부의 재원으로 한국지능정보사회진흥원의 지원을 받아 구축된 "소상공인 고객 주문 질의-응답 텍스트"을 활용하여 수행된 연구입니다. 본 연구에 활용된 데이터는 AI 허브(aihub.or.kr)에서 다운로드 받으실 수 있습니다.

참고문헌

[1] J. Li, A. Sun, J. Han, and C. LI, “A Survey on Deep Learning for Named Entity Recognition : Extended Abstract,” in *Proceedings of 2023 IEEE 39th International Conference on Data Engineering*, Anaheim: CA, pp. 3803-3804, April 2023. https://doi.org/10.1109/ICDE55515.2023.00335

[2] D. Nadeau and S. Sekine, “A Survey of Named Entity Recognition and Classification,” *Lingvisticae Investigationes*, Vol. 30, No. 1, pp. 3-26, January 2007. https://doi.org/10.1075/li.30.1.03nad

[3] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural Architectures for Named Entity Recognition,” in *Proceedings of the 2016 Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego: CA, pp. 260-270, arXiv:1603.01360, June 2016. https://doi.org/10.18653/v1/N16-1030

[4] V. Kumar, A. Choudhary, and E. Cho, "Data Augmentation using Pre-trained Transformer Models," in *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, Suzhou, China, pp. 18-26, December 2020. https://doi.org/10.48550/arXiv.2003.02245

[5] J. Wei and K. Zou, "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Hong Kong, China, pp. 6382-6388, November 2019. https://doi.org/10.18653/v1/D19-1670

[6] X. Dai, and H, Adel, "An Analysis of Simple Data Augmentation for Named Entity Recognition," in *Proceedings of the 28th International Conference on Computational Linguistics,* Barcelona, Spain(Online), pp. 3861-3867, arXiv:2010.11683, December 2020. https://doi.org/10.18653/v1/2020.coling-main.343

[7] G. S. Cho and S. B. Kim, "Korean Named Entity Recognition Using Data Augmentation Techniques," *Journal of the Korean Institute of Industrial Engineers*, Vol. 48, No. 2, pp. 176-184, April 2022. https://doi.org/10.7232/JKIIE.2022.48.2.176

[8] S. Edunov, O. Myle, A. Michael, and G. David, "Understanding Back-Translation at Scale," arXiv:1808.09381, August 2018. https://doi.org/10.48550/arXiv.1808.09381

[9] K. M. Yoo, D. Park, J. Kang, S.-W. Lee, W. Park, "GPT3Mix: Leveraging Large-scale Language Models for Text Augmentation," in *Proceedings of Findings of the Association for Computational Linguistics*, Punta Cana, Dominican Republic, pp. 2225-2239, arXiv.1808.09381, November 2021. https://doi.org/10.18653/v1/2021.findings-emnlp.192

[10] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators," arXiv:2003.10555, March 2020. https://doi.org/10.48550/arXiv.2003.10555

장동호(Dong-Ho Jang)

2019년～현　재: 경상국립대학교 컴퓨터공학과 재학
※관심분야 : AI, 머신러닝, LLM

부석준(Seok-Jun Buu)

2023년 : 연세대학교
컴퓨터과학과 졸업(공학박사)

2023년 9월～현　재: 경상국립대학교 컴퓨터공학과 조교수
※관심분야 : AI, 머신러닝, LLM, 사이버보안,
뉴로-심볼릭 인공지능

서영건(Yeong Geon Seo)

1987년 : 경상대학교 전산과(이학사)
1997년 : 숭실대학교 전산과(공학박사)

1989년～1992년: 삼보컴퓨터
1997년～현　재: 경상국립대학교 컴퓨터공학과 교수
2022년～현　재: 경상국립대학교 정보전산처장
※관심분야 : 컴퓨팅 사고, 의료 영상 처리, SLAM,
영상 인식, 컴퓨터 네트워크