http://www.jsebs.org
ISSN: 2288-3908

The Journal of Society for e-Business Studies
Vol.26, No.1, February 2021, pp.107-126
https://doi.org/10.7838/jsebs.2021.26.1.107

CNN기반 상품분류 딥러닝모델을 위한 학습데이터 영향 실증 분석

Empirical Study on Analyzing Training Data for CNN-based Product Classification Deep Learning Model

이나경(Nakyong Lee)*, 김주연(Jooyeon Kim)**, 심준호(Junho Shim)***

초 록

전자상거래에서 상품 정보에 따른 신속하고 정확한 자동 상품 분류는 중요하다. 최근의 딥러닝 기술 발전은 자동 상품 분류에도 적용이 시도되고 있다. 성능이 우수한 딥러닝 모델 개발에 있어, 학습 데이터의 품질과 모델에 적합한 데이터 전처리는 중요하다. 본 연구에서는, 텍스트 상품 데이터를 기반으로 카테고리를 자동 유추할 때, 데이터의 전처리 정도에 따른 영향력과 학습 데이터 선택 범위 영향력을 CNN모델을 사례 모델로 이용하여 비교 분석한다. 실험 분석에 사용한 데이터는 실제 데이터를 사용하여 연구 결과의 실증을 담보하였다. 본 연구가 도출한 실증 분석 및 결과는 딥러닝 상품 분류 모델 개발 시 성능 향상을 위한 레퍼런스로서 의의가 있다.

ABSTRACT

In e-commerce, rapid and accurate automatic product classification according to product information is important. Recent developments in deep learning technology have been actively applied to automatic product classification. In order to develop a deep learning model with good performance, the quality of training data and data preprocessing suitable for the model are crucial. In this study, when categories are inferred based on text product data using a deep learning model, both effects of the data preprocessing and of the selection of training data are extensively compared and analyzed. We employ our CNN model as an example of deep learning model. In the experimental analysis, we use a real e-commerce data to ensure the verification of the study results. The empirical analysis and results shown in this study may be meaningful as a reference study for improving performance when developing a deep learning product classification model.

키워드 : 상품분류, CNN 모델, 딥러닝, 학습데이터, 사전처리
Product Classification, CNN Model, Deep Learning, Training Data, Preprocessing

본 연구는 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2020R1F1A1075952).
본 논문은 한국전자거래학회 2020 추계학술대회에서 발표된 동일 제목의 우수논문이 학회지에 추천되어 확장 수정된 것임.

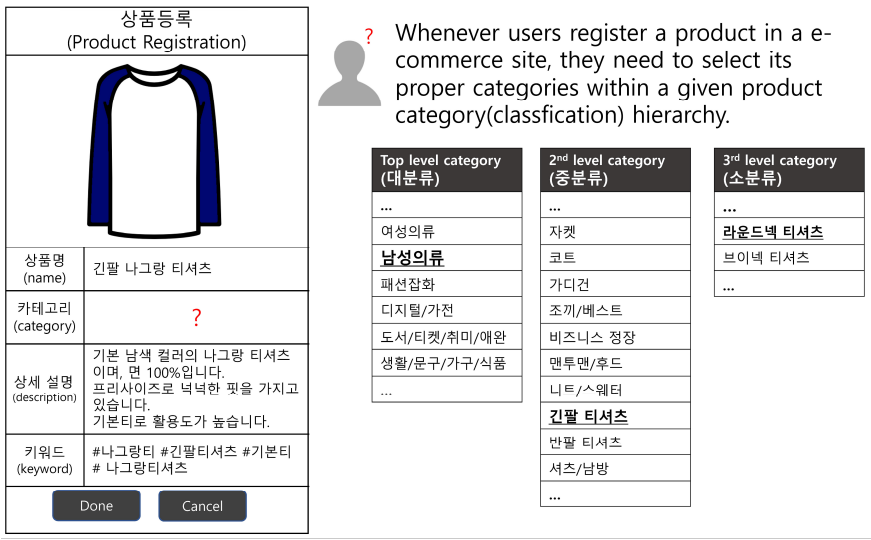
* First Author, Undergraduate student, Division of Computer Science, Sookmyung Women's University (lnk7424@sm.ac.kr)
** Undergraduate student, Division of Computer Science, Sookmyung Women's University(parang@sm.ac.kr)
*** Corresponding Author, Professor, Division of Computer Science, Sookmyung Women's University (jshim@sm.ac.kr)
Received: 2021-01-06, Review completed: 2021-01-26, Accepted: 2021-02-01

1. 서 론

인터넷의 발달로 전자상거래 시장이 더욱더 활성화되고 있다. 매일 수많은 상품이 온라인상에 등록되고 거래가 이루어진다. 각 상품은 상품명, 상품이 속한 카테고리, 상품가격, 상품에 대한 상세설명, 상품의 특징을 나타내는 키워드 등의 텍스트와 이미지를 포함한 다양한 정보로 구성된다. <Figure 1>은 전자상거래 쇼핑몰에서 상품을 등록할 때 사용자가 상품에 대한 정보와 카테고리를 선택하는 프로세스를 보여주는 예시 화면이다.

주어진 상품을 상품이 속해야 할 카테고리에 정확하게 분류하는 것은, 원하는 상품을 정확히 검색하여 구매 또는 판매에 이르도록 하는 효율성 측면뿐 아니라, 고객 만족도 더 나아가 쇼핑몰 수익에 영향을 미치는 중요한 요소이다. 각 쇼핑몰은 각기 다른 상품 분류 체계를 가지고 있으므로, 예를 들어 판매하고자 하는 상품의 카테고리를 판매자가 직접 입력해야 하는 중고 물품 거래인 경우, 판매자가 상품의 카테고리를 정확히 선택하여 입력하는 데는 많은 노력과 시간이 들게 되면서, 그 정확성도 보장할 수 없게 된다. 사람에 의해 수동으로 만들어져야 하는 대부분의 상품 정보와 달리, 상품 카테고리는 기타 정보를 이용해 유추할 수 있다. 이를 활용한 자동 카테고리 분류 기능은 전자상거래 시장에서 상품 등록 시 소모되는 시간과 비용을 줄이는 장점을 가진다.

전자상거래에서 상품을 자동 분류하려는 연구는 여러 방면으로 이루어져 왔다. 전자 카탈로그 데이터베이스를 이용한 전통적인 색인 기반 연구에서부터, 비교적 근래의 상품정보 온톨로지를 이용한 연구와 최근의 기계학습 기반 딥러닝 기술 적용 연구는 대표적 예이다[13, 14, 15, 21]. 딥러닝은 다양한 모델과 그에 따른 기술로서 발전해왔다. 하지만 기계학습을 기반으로 하는 학습 과정을 기반으로 한 딥러닝의 특성상,



〈Figure 1〉 An Example of Product Classification(or categorization) Process in e-Commerce

학습 데이터의 전처리 과정을 통한 품질 향상과 모델 성능 향상을 위한 학습 데이터의 선정 등은 매우 중요하다. 다양한 딥러닝 모델 중에서 CNN(Convolutional neural network)은 다중 분류 및 텍스트 분류에서 상대적으로 훈련 시간 대비 높은 성능 등의 장점을 가진 대표적인 모델 중 하나로 활발히 사용되고 있다[11, 20]. 딥러닝 기술을 이용한 상품자동분류 기존 연구들에서는 딥러닝 모델별 성능 차이에 따른 모델 선택 및 그에 따른 성능향상에 집중했지만, 상품자동분류 모델 개발에 있어 입력으로 사용되는 학습데이터가 성능에 미치는 영향을 체계적으로 분석 제시하고 있지는 않다.

이에 본 연구는 CNN 기반 상품자동분류 딥러닝 모델을 예시로, 모델 개발에 있어 입력으로 사용되는 학습데이터의 변화가 성능에 미치는 영향을 실제 국내 전자상거래 상품 데이터를 통한 실험을 통해 분석 제시한다. 학습데이터의 변화가 딥러닝 모델 성능에 영향을 미치는 실험 분석은 두 가지 측면에서 이루어진다. 첫 번째로 학습데이터로 사용되는 상품 정보 데이터의 개수 및 조합, 즉 학습 데이터로서의 속성 선택이 모델 성능에 미치는 영향을 분석한다. 두 번째로 입력 데이터의 단계별 전처리 과정을 통한 데이터 품질이 모델 성능향상에 미치는 영향을 분석한다.

이를 위한 본 논문의 구성은 다음과 같다. 먼저, 제 2장에서는 CNN과 딥러닝 기술을 이용한 최근의 상품 자동 분류 관련 연구를 소개한다. 제 3장에서는 본 연구의 실험 분석에 사용되는 CNN 모델 구조를 소개하고, 제 4장에서는 학습과 실험에 사용되는 데이터의 특성과 전처리 단계 과정을 소개한다. 제 5장에서는 학습데이터의 변화가 딥러닝 모델 성능에 영향을 미치는 실험 분석을 앞서 소개한 두 측면으로 각각 설명한다. 마지막으로 제 6장에서는 결론과 연구 방향을 제시한다.

2. 관련연구

2.1 딥러닝과 CNN

딥러닝은 머신 러닝의 한 분야로 역전파(Back-propagation) 알고리즘을 사용한다. 기계가 이전 계층(Layer)의 표현(Representation)에서 각 계층의 표현을 계산하는 데 사용되는 내부 매개 변수를 어떻게 변경해야 하는지 표시함으로서 대규모 데이터 세트에서 복잡한 구조를 발견한다. 다수의 처리 계층으로 구성된 계산 모델이 데이터 표현을 학습하는 것이다. 이를 통해 음성 인식, 개체 감지 및 약물 발견 및 유전체학 등 많은 분야에서 최첨단 기술이 크게 발전하였다. 그 중에서도 심층 합성곱 신경망(Deep Convolutional Net)은 이미지, 비디오, 음성 및 오디오 처리 능력을 향상시켰고, 순환 신경망(Recurrent Net)은 텍스트 및 음성과 같은 순차적 데이터 관련 기술을 발전시켰다[12].

CNN(Convolutional Neural Network)은 딥러닝 모델의 한 종류로 이미지 분류에 많이 쓰이는 모델이다. 기존 영상처리의 필터 역할을 하는 합성곱(Convolution)과 신경망을 결합한 구조이다. 합성곱레이어(Convolution Layer)가 필터를 통해 이미지의 특징을 추출하면 풀링 레이어(Polling Layer)에서 특징을 강화하고 이미지의 크기를 줄인다. 이 과정을 반복하며 이미지의 특징(Feature)을 추출한다. 해당 이미지 특징을 Fully Connected Layer에 보내

분류를 수행한다. 최근에는 텍스트 분류에도 활발히 쓰이고 있다. 이미지의 특징을 추출하는 것처럼 CNN 모델을 자연어 처리(Natural Language Processing)에 적용하면 텍스트의 특정 지역에서 특징(Feature)을 추출한다[9].

2.2 상품 자동 분류와 딥러닝

상품 카테고리 분류는 각 상품을 알맞은 세부 카테고리로 연결하는 것이다. 각 상품은 상품명, 카테고리, 가격 등의 정보를 수반하는데, 수동으로 만들어야하는 상품명, 상세 정보와 달리 카테고리는 기타 메타 데이터를 이용해 추론할 수 있다. 상품 자동 분류는 이를 이용해 수동으로 카테고리를 분류할 때 발생하는 시간과 경제적 손실을 줄일 수 있다[6, 10, 19]. 전자상거래 시장에서 정확한 상품 카테고리 분류는 수익 및 고객 만족도와 연관되며, 그 중요성이 더욱 커지고 있다[16, 17]. 국내외의 여러 쇼핑몰에서 정확도 높은 상품 자동 분류 기술을 개발하고 있다.

상품 분류 문제는 상품의 정보 중 텍스트 데이터만을 사용할 경우, 텍스트 분류(Text Classification) 문제로 재해석 할 수 있다. 기존에 텍스트 분류에 많이 쓰이던 SVM, 나이브 베이즈 등을 상품 분류에 적용하여 사용했다[2, 3, 4, 7, 16]. <Table 1>은 상품 자동 분류 문제에 대한 연구에서 어떤 분류기(Classifier)를 사용했는지 정리한 것이다. 최근에는 딥러닝 모델을 많이 활용하고 있으며, 상품 사진인 이미지 데이터도 입력 데이터로 사용하는 등 여러 시도가 이어지고 있음을 알 수 있다[6, 23]. 딥러닝 모델을 사용한 몇 가지 예시를 소개하면 다음과 같다.

Rakuten에서는 상품 분류를 위한 데이터 세트와 베이스라인을 제공하며 데이터 챌린지를 개최했다[15]. 상품 정보 중 제목 데이터를 이용해 3,008개의 카테고리 중 하나로 분류했다. 전자 상거래 환경에서 롱테일(long-tail) 현상은 중요한 문제가 아니지만, 이로 인해 성능

〈Table 1〉 Related Works On Deep-Learning based Product Classification

Works	Deep learning models or classifiers
Cortez et al.[3]	feature weight optimization method, KNN, CFC
Dalal et al.[4]	Naïve Bayes, Decision Tree, Neural Network, Support Vector Machine, Hybrid approach
Aanen et al.[1]	wordNet based algorithm(word sense disambiguation procedure)
Ha et al.[6]	RNN based DeepCN
Zahavy et al.[23]	Text Classification-CNN Image Classification-VGG16
Xia at al.[20]	CNN based A-CNN
Skinner[18]	LSTM
Yu et al.[22]	Fasttext, TextCNN, VDCNN, TextRNN, AbLSTM
Goumy et al.[5]	Single Classifier, Top Down Classifier, CNN+Glove pre trained word embedding
Krishnan et al.[11]	Multi-CNN, Multi-LSTM

지표로 가중치를 부여한 정밀도(Precision), 재현율(Recall), F1-score를 사용하였다. 참가 팀들은 최소 0.7226에서 최대 0.8510의 F1-score를 기록했다. Skinner[18]에서는 해당 라쿠텐 데이터 세트를 사용하여 LSTM을 기반으로 레이어와 파라미터에 변화를 주며 성능을 비교했다. 제목 데이터를 사용했으며 가중치를 부여한 Precision, Recall, F1-score로 성능을 비교했다. 0.7 후반에서 0.8 초반의 F1-score를 기록했다.

Krishnan and Amarthaluri[11]에서는 상품 카테고리 분류를 계층적 모델과 플랫 모델을 사용하여 비교하고 특정 경우에 플랫 모델이 더 나은 성능을 발휘함을 증명했다. 계층적 모델로는 bag of word를, 플랫 모델로는 multi-CNN과 multi-LSTM을 사용했다. top-n accuracy로 성능을 비교했으며 플랫 모델이 더 높은 성능을 보임을 증명했다.

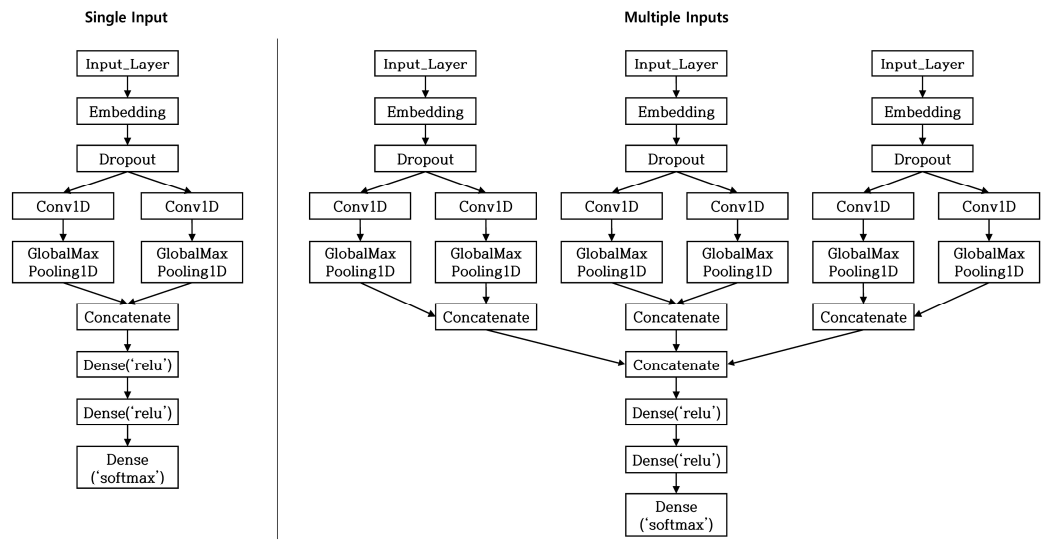
Xia et al.[20]는 굴절어인 영어가 아닌 교착어인 일본어로 구성된 제목을 이용해 전처리를 진행하고 상품을 분류했다. 상품을 분류하는 모델로 CNN 기반의 ACNN 모델을 제안했다. GBT 모델을 비교군으로 정하여 CNN 기반의 모델이 학습 시간은 짧지만, 비슷한 성능을 보여줌을 증명했다. 18개의 카테고리로 상품을 분류했으며 95.90~96.27 사이의 Micro-f1을 기록하였다.

Ha et al.[6]은 자체 쇼핑몰의 한국어 데이터를 사용하여 연구로는 상품 분류를 위한 새로운 딥 러닝 모델로 DeepCN이라는 RNN 기반의 모델을 제안하였다. 상품명, 브랜드명 등 다양한 텍스트 데이터를 사용했으며, 이미지 데이터의 사용이 긍정적인 영향을 미침을 증명했다. 12개의 카테고리로 분류하며 데이터 불균형을 고려해 상대 정확도로 성능을 비교했다.

이상의 연구에서와같이, 같은 데이터를 사용하여 전처리한 뒤 여러 가지 딥러닝 모델별로 비교하거나, 이미지 데이터를 추가하는 연구는 활발히 이루어져 왔다. 본 연구에서는 모델별 성능 평가가 아니라, 딥러닝 모델 개발에 있어 학습 데이터의 중요성 자체에 주목한다. 이를 위하여 한국어 텍스트 데이터를 중심으로 학습 데이터의 변화가 모델 성능에 미치는 영향력을 구체적으로 분석 제시한다는 점에서 기존 연구들과 차별성이 있다.

3. 모 델

본 연구에서 고안한 상품자동분류를 위한 딥러닝 모델은 CNN(Convolutional Neural network) 기반이다. 일반적으로 CNN 모델은 입력(Input)과 출력(Output) 사이에 합성곱 레이어(Convolutional Layer)와 풀링 레이어(Pooling Layer)를 가진 다층 신경망 구조를 가진다. 여러 차례의 pilot 모델학습 과정을 거친 결과, 본 연구에서 학습 데이터와 모델 간 영향 분석을 위해 개발한 CNN 모델의 구조는 <Figure 2>와 같다. 그림에서는 단일 입력(Single Input)일 때의 모델 구조와 다중 입력(Multi Input)일 때의 모델 구조를 구분하여 설명하고 있다. 모델의 입력(Input)은 상품에 대한 텍스트 데이터이며, name, keyword, description 중 하나 이상의 속성(Attribute)이 들어갈 수 있다. 모델의 출력(Output)은 해당 상품이 속한 중분류 카테고리이다.



〈Figure 2〉 Our CNN Model Structures(Left: When the Model Receives a Single Input, Right: When the Model Receives Multiple Inputs)

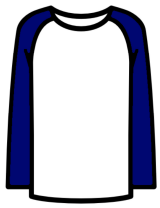
각 속성은 입력 데이터로써 200차원의 임베딩 레이어(Embedding Layer)를 거친다. 해당 파라미터는 Krishnan and Amarthaluri[11] 등 관련 연구에서 200을 사용한 것을 참고하여 정하였다. 드롭아웃(Dropout) 정규화를 거친 뒤 합성곱(Convolution)을 수행한다. name과 keyword는 커널 사이즈가(kernel size)가 2와 3인 합성곱 레이어(Convolutional layer)를 사용하고, description은 커널 사이즈가(kernel size)로 3, 4, 5를 사용한다. 해당 파라미터 세팅은 Krishnan and Amarthaluri[11]에서 1, 2, 3, 4, 5를 사용한 것과 Xia et al.[20]에서 1, 3, 4, 5를 사용한 것을 참고하였다. 합성곱 레이어(Convolutional layer)의 결과를 글로벌 맥스 풀링(Global Max Pooling)한 뒤, Softmax 활성화 함수를 사용해 독립된 레이어를 연결(Concatenation)한다. 그 후 Relu, Softmax 활성화 함수를 사용한 Dense 레이어를 거치면 분류 결과가 출력된다. 다수의 속성을 사용할 경우 Dense 레이어를 한 번 더 연결(Concatenation)하는 과정을 수행한 뒤 분류 결과를 출력한다.

4. 데이터

4.1 특징

모델 개발에 사용한 학습 데이터는 국내 유수의 중고물품 전자상거래 A사의 2019년 하반기와 2020년 상반기 사이의 실제 등록 상품 데이터를 사용하였다. 중고거래 특성상 누구나 자유롭게 판매자가 될 수 있어 상품 게시글의 형식이 간단하고 자유로우며, 자연어에 가까운 특징이 있다.

<Figure 3>은 등록된 상품의 예시이며, 학습에 사용된 속성명과 간단한 설명을 포함하고 있다. 상품 정보 중 학습에 사용된 것은 상품명인 name, 상품을 나타내는 키워드인 keyword,



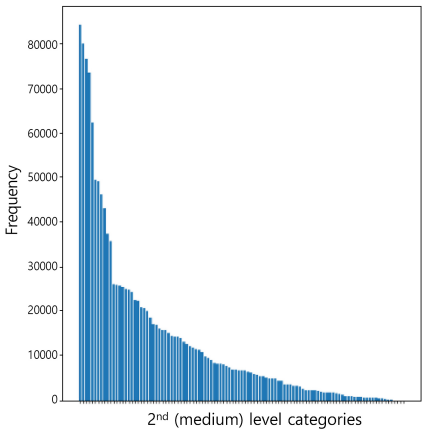
상품명 (name)	긴팔 나그랑 티셔츠	**name :** product name
카테고리 (category)	남성 의류 / 긴팔 티셔츠 / 라운드넥 티셔츠	**category_id :** product categories within the category hierarchy (top/2nd/3rd levels)
상세 설명 (description)	남색 컬러의 나그랑 티셔츠이며, 면 100%입니다. 프리사이즈로 넉넉한 핏을 가지고 있습니다. 기본티로 활용도가 높습니다.	**description :** product description
키워드 (keyword)	#나그랑티 #긴팔티셔츠 #기본티 #나그랑티셔츠	**keyword :** tags and keywords associated with the product

〈Figure 3〉 An Example of Product Data and its Attributes: *Name, Category, Description, and Keyword*

상품의 상세 설명인 description, 그리고 카테고리인 category_id이다. name은 상품명이자 게시글의 제목으로 짧은 문장 혹은 단어 단위로 구성되어 있다. keyword는 검색 용이성을 높이기 위한 태그 혹은 키워드로 구성된 데이터이다. 사용자가 단어 혹은 어절 단위로 상품과 관련된 텍스트를 입력한 것을 문자열로 연결하였다. 상세 설명인 description은 다른 속성에 비해 데이터 길이가 길고 문장 위주로 구성되어 있다. 상품에 대한 자세한 설명은 물론, 상태, 배송 방법 등을 함께 명시한다.

상품은 상품 분류 체계에 따라 상품이 속할 분류, 즉 카테고리(category_id)가 지정된다. 상품 분류 체계는 전자상거래 사이트나 시스템에 따라 상이한 구조를 가진다. 가장 대중적으로 사용되는 분류체계는 대분류-중분류-소분류의 계층적(hierarchical) 구조로 알려져 있다. 예를 들어 <Figure 3>에서 제시된 '긴팔 나그랑 티셔츠' 상품은 '남성 의류' 대분류 밑의, '긴팔 티셔츠' 중분류, 다시 그 아래의 '라운드넥 티셔츠' 소분류에 최종적으로 분류되어 등록되어진 예시이다. 본 연구에서 사용된 실데이터의 상품 분류체계는 대분류 기준 10여 개의 카테고리를 사용하며, 대분류 밑의 중분류의 개수는 120여 개이다. 우리는 대분류, 중분류 체계에서 자동 분류 실험을 수행하여 그 학습 데이터와 모델 성능 영향 평가를 진행하였다. 두 분류체계가 비슷한 실험 결과를 보였기에, 본 논문에서는 별도의 언급이 없는 경우라면 설명 용이성을 위하여 중분류 기준의 실험 결과를 제시한다.

총 상품 데이터 개수는 1,401K개이다. 상품의 개수는 카테고리별로 상이한 분포를 가지는 것으로 알려져 있다[15, 20, 22]. 본 연구에서 사용된 데이터도 이러한 특성을 따르고 있었다. <Figure 4>는 카테고리별 상품 데이터의 분포를 중분류 체계를 기준으로 나타낸 것이다. 데이터의 개수가 카테고리별로 상당히 큰 차이를 보임을 알 수 있다. 참고로, 중분류를 기준으로 카테고리별 단순 평균 데이터 수는 11K개다. <Figure 5>는 각 속성의 토큰 길이별 분포를 나타낸 것이다. 데이터 길이별 분포 역시 타 연구에서 사용한 데이터와 비슷한 특징을 가지고 있다. 이러한 데이터 특성이 모델 학습과 성능에 미치는 영향에 대해서는 다음 절인 데이터 전처리 과정에서 좀 더 살펴본다.

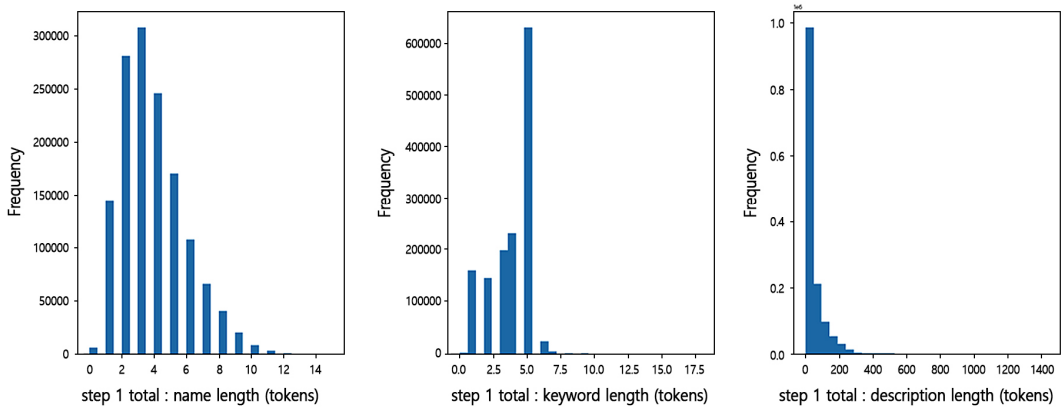


〈Figure 4〉 Number of Products for Each Category (based on the 2nd (Medium) Levels)

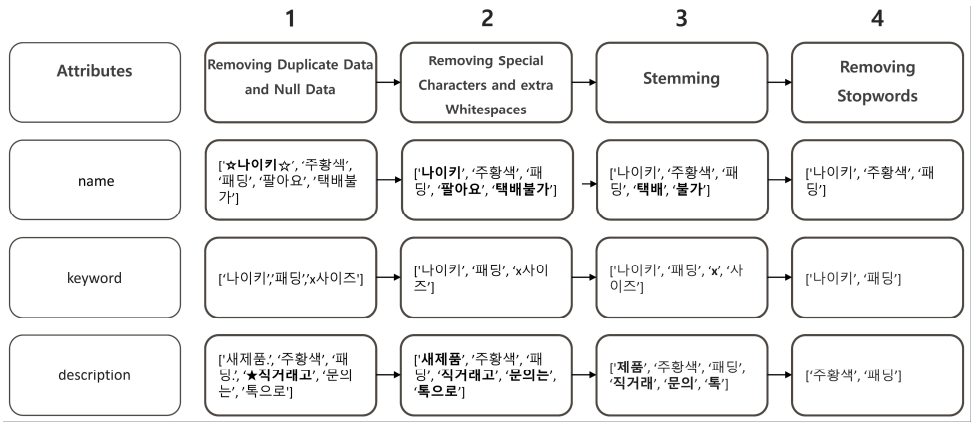
4.2 전처리 과정

학습 데이터의 전처리에 따른 CNN 딥러닝 모델 영향력을 비교 분석하기 위해, 본 연구에서는 학습 텍스트 데이터를 사용할 때 적용하는 전처리 과정을 총 4단계로 나누었다. <Figure 6>은 토큰화를 거친 텍스트 데이터의 형태를 예시로 나타낸 것이다.

첫 번째로 중복 데이터와 null이 포함된 데이터를 제거했다[15]. 본 연구의 기본 데이터이다. 두 번째는 특수문자를 제거한 데이터이다[6, 20]. 여기서 특수문자는 문자와 숫자를 제외한 모든 문자를 의미한다. 예를 들어, name 부분을 보면 '☆나이키☆'라는 토큰이 있는데 이처럼 특수문자가 포함되어 있으면, '☆나이키☆'와 '나이키'는 다른 단어가 된다. 이런 현상을 없애기 위해 특수문자를 제거했다. 세 번째는 형태소 분석기를 적용한 데이터이다[9]. 형태소는 의미를 가지는 가장 작은 단위로, okt 형태소 분석기를 적용하여 명사인 형태소만 추출하였다. description 부분을 보면 '새제품'이 형태소 분석기 적용 후 '제품'이라는 명사만 남은 것을 확인할 수 있다. okt 형태소 분석기는 오픈소스 기반의 트위터 형태소 분석기다. 네 번째로 불용어를 선정해 제거했다. 불용어는 고빈도어 중에서 문법적 기능을 수행하는 형태소와 의미를 구성하는 핵심 품사를 제외한 것이다. 그 이후 데이터를 직접 살펴 목록을 조정한다[8]. 본 연구에서는 상품 분류의 특성을 고려하며 해당 기준에 따라 '판매', '제품', '직거래' 등의 형태소를 불용어로 선정하였다.



〈Figure 5〉 Name, Keyword and Description Length (in tokens) Distributions



〈Figure 6〉 Data Preprocessing Steps and Tokenized Result Data Examples for Each Step

〈Table 2〉 Number of Tokens that Each Preprocessing Step Produces

preprocessing step \ attribute	name	keyword	description
Removing Duplicate Data and Null Data	489,007	531,069	2,042,127
Removing Special Characters and extra WhiteSpaces	436,353	529,130	1,812,243
Stemming	99,545	82,635	203,811
Removing Stopwords	99,450	82,541	203,715

<Table 2>는 각 단계의 데이터에 토큰화(tokenization)를 거친 뒤 만들어진 토큰의 개수를 나타낸 것이다. <Table 2>를 보면 전처리 각 단계를 거칠 때마다 토큰의 수가 많이 줄어드는 것을 확인할 수 있다. 특히, 형태소 분석기를 적용했을 때 토큰의 수가 가장 많이 줄어들었다. 어미, 접사와 조사 혹은 잘못된 띄어쓰기로 인해 다른 토큰으로 구분되었던 것들이 형태소 분석기를 거친 후 명사인 어간만 남아 토큰 수는 감소하고 각 토큰의 빈도수는 향상하였다.

데이터는 띄어쓰기를 기준으로 토큰화를 진행하였다[11, 15]. 속성별 최대 토큰 길이로 name과 keyword는 최대 토큰 길이를 사용했고 description은 200으로 제한하였다. description의 평균 토큰 길이는 47.88이지만 최대토큰 길이는 1441로 최대 길이로 사용할 경우 학습이 불가능했다. 동일한 모델로 최대 길이와 200으로 제한한 경우를 500K개의 데이터를 사용해 비교한 결과 성능에서 약 0.003의 차이를 보였으나 학습 시간은 5시간 17분이 감소하였다. 따라서 본 연구의 모든 실험에서 description의 데이터들을 200으로 제한한 후 학습을 진행하였다.

전체 데이터 세트에서 10%를 랜덤 샘플링하여 test 데이터로 사용하였다[20]. 나머지 데이터의 90%는 train에, 10%는 validation에 사용하였다. 즉, train : valid : test = 81 : 9 : 10이다. 학습 데이터의 변화가 딥러닝 모델 성능에 영향을 미치는 실험 분석은 1) 학습데이터로 사용되는 상품 정보 데이터의 개수 및 조합, 즉 학습 데이터로서의 속성 선택이 모델 성능에 미치는 영향 분석과 2) 입력 데이터의 단계별 전처리과정을 통한 데이터 품질이 모델 성능향상에 미치는 영향 분석을 수행하였다.

5. 실 험

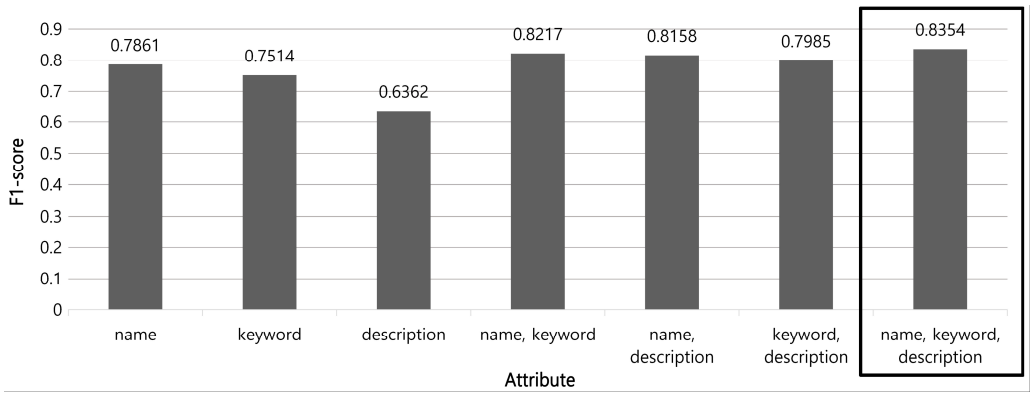
5.1 학습 데이터로서의 속성 선택이 모델 성능에 미치는 영향

우리는 모델에 학습데이터로 입력되는 상품 정보 속성이 딥러닝 모델의 성능에 어떻게 영향을 미치는지 파악하기 위해서 name, keyword, description의 세 가지 속성을 단일 입력, 속성 간 조합을 통한 다중 입력으로 구분하여 실험을 진행하였다. 즉, 학습 입력데이터로서 속성의 선택을, name, keyword, description, (name, keyword), (name, description), (keyword, description), (name, keyword, description)등 일곱가지 경우를 모두 살펴보았다. 성능 메트릭은 딥러닝 분류 예측 성능에 일반적으로 사용되는 Precision, Recall, F1-score를 사용하였다. 실험 결과는 <Table 3>에 제시하였다. 이 중 F1- score는 Xia et al.[20], Lin et al.[15]의 상품분류 관련 대회 및 연구에서 주요 성능 메트릭으로 제시하고 있어, 비교 참조 용이성을 위해 <Figure 7>에 그래프 형식으로 재구성하였고, 편의상 F1-score를 기준으로 성능 분석 설명을 제시한다.

〈Table 3〉 Precision, Recall, F1-Score by Attribute Selection

attribute \ performance metric	Precision	Recall	F1-score
name	0.7882	0.7908	0.7861
keyword	0.7607	0.7567	0.7514
description	0.6610	0.6395	0.6362
name, keyword	0.8243	0.8251	0.8217
name, description	0.8172	0.8189	0.8158
keyword,description	0.8011	0.8021	0.7985
name, keyword, description	0.8367	0.8380	0.8354

<Figure 7>에서 볼 수 있듯이, 입력 데이터로서 하나의 속성을 사용한 경우에는 name, keyword, description 순서로 성능이 우수한 것으로 나타났다. <Figure 1>과 <Figure 3>의 예시 그림에서 볼 수 있듯이, 소비자에게 가장 먼저 노출되는 게시글의 제목인 name과 검색 용이성을 높이는 keyword에 상품 분류의 핵심이 되는 데이터가 포함되어 있음을 알 수 있다. 반면, 상품의 상세 설명인 description은 name과 keyword보다 평균 0.1 이상 차이 나는 낮은 성능을 보였다. 상품 상세 설명은 다른 속성에 비해 상품 카테고리 분류에 사용되는 학습데이터로서의 품질이 떨어지는 것으로 보인다.



〈Figure 7〉 F1-Scores for Different Attribute Selections

입력 데이터로서 2개의 속성을 조합하여 사용한 실험 결과에서는, name과 keyword를 함께 사용했을 때 성능이 가장 좋았으며, 단일로 사용했을 때 다른 속성에 비해 0.1 이상 차이가 났던 description도 다른 속성과 함께 사용했을 때는 다른 조합과의 차이가 0.01 정도로 줄어들 정도로 성능이 크게 향상되었다. 여러 속성의 조합으로 학습을 진행했을 때 multi-input에서는 모두 0.8 내외의 성능을 보였으며, 가장 성능이 좋았던 것은 name, keyword, description을 모두 사용했을 때이었다.

<Table 4>는 서로 다른 속성의 입력 조합이 상품 카테고리별 예측 성능에 구체적으로 어떠한 영향을 주었는지 살피기 위해, 중분류 기준으로 F1-score 값 범위별로 카테고리 개수를 살펴본 분석표이다. 예를 들어, 테이블의 (1, 1)번째 셀의 값은, name 속성만 입력데이터로 사용한 모델은 12개의 중분류 카테고리에 대해서 0.9 이상의 F1-score 값을 넘는 성능을 보인 것을 의미한다. 이에 비해 (1, 3)번째 셀의 값은, description 속성만 입력데이터로 사용한 모델은 2개의 중분류 카테고리에 대해서만 0.9 이상의 F1-score 값을 넘는 성능을 보인 것이다.

조합별로 0.9 이상을 기록한 카테고리들을 살펴봤을 때, 단일 속성에서 0.9 이상의 F1-score를 기록한 카테고리들이 해당 속성들을 조합하여 사용했을 때에도 0.9 이상의 F1-score를 기록하였다. name과 keyword를 함께 사용한 경우 name만 사용했을 때보다 0.0356만큼, keyword만 사용했을 때보다 0.0703만큼 F1-score가 향상되었다. <Table 4>를 보면 name과 keyword 각 한 개의 속성만 사용한 경우, 0.9 이상의 F1-score를 기록한 카테고리의 수는 각각 12개와 13개이다. name과 keyword 두 개를 사용한 경우, 각각 속성을 사용한 0.9 이상 예측 가능한 카테고리들을 모두 포함하여 총 20개의 카테고리가 0.9 이상의 F1-score를 기록하였다. 예를 들어, 국산차(딜러)와 애완(반려) 카테고리는 keyword에만 포함되어 있고 스커트/치마는 name에만 포함된 카테고리였는데, 이 둘을 함께 사용했을 때 세 카테고리 모두 F1-score가 향상되었다. 각각 사용했을 때 F1-score가 모두 높았던 카테고리들이 함께 사용했을 때도 높은 순위에 있었으나, 국산차(딜러)와 같이 keyword에서는 3위라는 높은 순위를 기록하고 name에서는 상대적으로 낮은 41위를 기록했음에도 두 속성을 함께 사용했을 때 2위를 기록한 카테고리도 있었다.

name과 description을 함께 사용했을 때는 name만 사용했을 때보다 0.0297만큼 description만 사용했을 때보다 0.1796만큼 F1-score가 향상되었다. 이 경우에도 단일로 사용했을 때 0.9 이상의 F1-score를 기록한 카테고리들이 조합하여 사용했을 때 0.9 이상을 기록한

〈Table 4〉 Number of Categories(2nd level) by F1-score Ranges

F1-score \ attribute	name	keyword	description	name, keyword	name, description	keyword, description	name, keyword, description
0.9 ≤	12	13	2	20	17	16	24
0.8 ≤	38	34	10	49	48	42	52
0.7 ≤	61	54	27	74	71	66	76
0.6 ≤	83	78	47	91	86	87	88
0.5 ≤	96	95	70	104	102	102	106

카테고리에 모두 속해있었다. description에서 0.9 이상의 F1-score를 기록한 카테고리는 전체 120여 개 카테고리중 단 2개인데, 이중 타이어/휠의 경우 name에서는 0.95174, description에서는 0.9232로 가장 높은 F1-score를 기록하였고, name과 description을 함께 사용했을 때는 0.9470으로 name보다 F1-score가 다소 낮아진 결과를 보였다.

keyword와 description을 함께 사용했을 때는 keyword만 사용했을 때보다 0.0471만큼, description만 사용했을 때보다 0.1623만큼 F1-score가 향상되었다. 따로 사용했을 때 0.9 이상의 F1-score를 기록한 카테고리들이 함께 사용했을 때도 0.9 이상의 F1-score를 기록하였다. 위와 마찬가지로 keyword에서 0.9 이상을 기록한 13개의 카테고리와 description에서 0.9 이상을 기록한 카테고리가 keyword와 description을 함께 사용했을 때 0.9 이상을 기록한 16개의 카테고리에 모두 속해있었다. 한편, 향수/아로마, 모자, 도서/책 등의 각각으로 사용했을 때 0.9 미만의 F1-score를 기록했던 카테고리들은 다중으로 사용할 경우 F1-score가 상대적으로 많이 향상되었다.

F1-score가 가장 높았던 name, keyword, description의 경우 name, keyword에서 0.9 이상을 기록했던 카메라/DSLR카테고리를 제외하고 다른 속성별 실험해서 0.9 이상을 기록했던 모든 카테고리가 24개 안에 포함되어 있었다. 오토바이/스쿠터, 자전거/MTB 등은 해당 조합에서만 0.9 이상을 기록한 특성을 보이었다. 0.9 미만의 F1-score를 기록한 곳에서도 위와 비슷한 양상을 보였다.

결론적으로, 각각의 카테고리에서 F1-score가 높아 조합하여 사용했을 때 F1-score가 향상된 카테고리가 가장 많았지만, 한쪽의 F1-score가 상대적으로 매우 높아 합쳤을 때 F1-score가 향상된 카테고리도 있었다. 반면에 조합하여 사용했을 때 오히려 F1-score가 하락한 카테고리도 존재하였다. 속성별 분류 성능이 다중 입력으로 사용할 때 영향을 미치는 것은 많지만 반드시 성능 향상을 보장하는 것은 아님을 알 수 있다.

또한, F1-score가 높은 카테고리들은 포함된 상품의 특징이 상대적으로 뚜렷한 점이 성능에 영향을 끼친 것으로 보인다. 예를 들어, 특정 상품만 판매하는 제조사의 브랜드명이나 카테고리명 그 자체를 토큰으로 포함하고 있는 데이터가 많았고, 그 외의 원피스, 스커트와 같이 여성의류에만 포함된 카테고리도 F1-score가 높았다. 반면, 0.5 이하의 낮은 F1-score를 기록한 카테고리들은 다음의 두 가지 특징이 있었다. 첫 번째는 각 카테고리에 속한 데이터의 양이 적어 학습이 제대로 이루어지지 않은 경우이다. 중분류를 기준으로 각 카테고리의 평균 데이터 수는 11K이지만, 예를 들어 배우(여) 카테고리와 산업용품 카테고리는 각각 30여 개, 650여 개의 데이터만 존재하였다. 위와 같이 데이터의 양이 적은 경우, 대체로 낮은 성능을 보였다. 두 번째는 카테고리별 상품의 특징이 두드러지지 않는 경우이다. 성별, 나이 등에 따라 남성의류, 여성의류 등으로 나누어져 있는데, 해당 카테고리의 중분류에 의류 관련 중분류가 따로 존재하여 이를 구분할 수 있는 특징이 두드러지지 않는 경우이다. 예를 들어 긴팔 티셔츠, 반팔 티셔츠, 반바지 등과 같이 여성의류와 남성의류에 모두 속해 있는 카테고리나 남주니어의류(7세~), 여주니어의류(7세~), 여아의류(3~6세), 남아의류(3~6세)와 같은 카테고리의 성능이 상대적으로 낮았다.

5.2 입력데이터 전처리 과정이 모델 성능에 미치는 영향

이 절에서는, 입력데이터의 전처리 과정이 딥러닝 모델의 성능에 어떻게 영향을 미치는지 파악하기 위해서 속성별 전처리 단계를 기준으로 실험 분석을 제시한다. 사용한 성능 메트릭은 이전 절과 같이 Precision, Recall, F1-score를 사용하였다. name의 실험 결과를 <Table 5>로, keyword의 실험 결과를 <Table 6>으로, description의 실험 결과를 <Table 7>로 제시한다. 그리고 앞 절과 동일하게, F1-score 성능 메트릭만 결과만 <Figure 8>에 그래프 형식으로 재구성하였고, 편의상 F1-score를 기준으로 성능 분석 설명을 제시한다.

<Figure 8>를 보면 데이터 전처리과정의 첫 번째 단계, 즉, 중복, null데이터를 제거한 경우, name은 0.7429, keyword는 0.7381, description은 0.6004의 F1-score를 각각 기록했다. 이어, 전처리과정의 두 번째 단계인, 무분별한 공백문자와 특수문자를 제거하게 되면, name은 0.0065, description은 0.0075만큼 F1-score가

〈Table 5〉 Precision, Recall, F1-Score by Each Preprocessing Step when the Input Attribute is ***name***

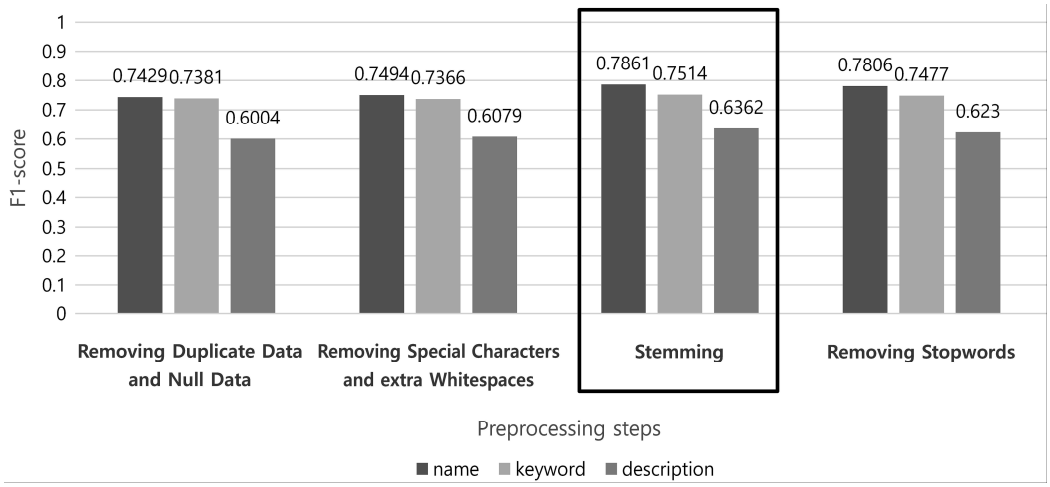
preprocessing step \ performance metric	Precision	Recall	F1-score
Removing Duplicate Data and Null Data	0.7542	0.7445	0.7429
Removing Special Characters and extra WhiteSpaces	0.7632	0.7492	0.7494
Stemming	0.7882	0.7908	0.7861
Removing Stopwords	0.7822	0.7849	0.7806

〈Table 6〉 Precision, Recall, F1-Score by Each Preprocessing Step when the Input Attribute is ***keyword***

preprocessing step \ performance metric	Precision	Recall	F1-score
Removing Duplicate Data and Null Data	0.7467	0.7418	0.7381
Removing Special Characters and extra WhiteSpaces	0.7457	0.7398	0.7366
Stemming	0.7607	0.7567	0.7514
Removing Stopwords	0.7555	0.7523	0.7477

〈Table 7〉 Precision, Recall, F1-Score by Each Preprocessing Step when the Input Attribute is ***description***

preprocessing step \ performance metric	Precision	Recall	F1-score
Removing Duplicate Data and Null Data	0.6142	0.6060	0.6004
Removing Special Characters and extra WhiteSpaces	0.6223	0.6124	0.6079
Stemming	0.6610	0.6395	0.6362
Removing Stopwords	0.6529	0.6243	0.6230



〈Figure 8〉 F1-Scores for Each Preprocessing Step

1단계에 비해 증가했고, 반면에 keyword는 0.0015만큼 소폭 하락했다. 세 속성 모두 성능이 가장 좋았을 때는 전처리 단계의 3단계인 형태소 분석기를 이어서 적용했을 때이다. name은 0.0367, keyword는 0.0148, description은 0.0283만큼 향상되었다. 이는 같은 의미를 가졌으나, 띄어쓰기, 어미, 조사 등에 의해 서로 다른 형태로 사전에 등록된 것이 형태소 분석기를 통해 제거되어 성능 향상에 도움을 준 것으로 보인다. 실험 결과의 특이한 결과는, 4번째 단계인 불용어를 제거했을 경우이다. 실험 결과에서는 세 속성 모두 F1-score가 3단계보다 오히려 다소 감소하였다. 이것은 입력 데이터에서 데이터 품질을 높이기 위해서 특정 토큰을 불용어를 선정하여 임의로 제거하는 경우, 선별에 있어 신중해야 하며, 카테고리별 특성에 유의해야 함을 의미할 수 있다. 이를 위해, 속성별, 전처리 단계별로 F1-score 변화를 심층 분석한 <Table 8>, <Table 9>, <Table 10>을 살펴보자.

<Table 8>에서는 name 속성에 대하여, 전처리 두 번째 단계부터 단계별로, 중분류 카테고리 기준으로 F1-score가 증가, 감소, 동일한 카테고리의 개수와 평균 증가율, 평균 감소율을 보여준다. 특수문자를 제거했을 때 성능이 많이 향상된 중분류 카테고리 가운데 하나는 운동화/캐주얼화(0.1974)이다. 해당 카테고리는 사이즈나 정품 유무, 상품 번호 등을 명시할 때 '[245]', '(245)'와 같이 특수문자로 인해 각각 다른 토큰이 되었다. 특수문자가 제거된 후에는 '245'라는 하나의 토큰이 만들어져 각 단어의 빈도수가 올라간 것을 확인하였다. 참고로 본 절에서는, 카테고리의 증가율을 괄호안에 표기하였는데, 앞서 표기는 운동화/캐주얼화의 F1-score가 0.1974만큼 향상되었다는 표기이다.

<Table 8>에 따르면 F1-score가 향상된 카테고리가 가장 많은 단계는 92개 카테고리가 증가세를 보인 형태소 분석기를 적용했을 경우이다. 성능이 많이 향상된 카테고리로서, 맨투맨/후드티(0.2579), 원피스(0.129) 등 있는데, 해당 카테고리의 데이터를 살펴본 결과, 같은 상품명, 브랜드명이지만 띄어쓰기로 인해 다른

단어로 등록되었던 것들이 형태소 분석기를 거치면서 하나의 형태소로 통일되게 되어, 마찬가지로 토큰의 수는 줄어들고, 토큰 당 빈도수는 상승하였다. 불용어를 제거했을 때 성능이 향상된 것은 카오디오/영상(0.2)을 예로 살펴볼 수 있다. 해당 카테고리는 불용어 제거 전에 '국내', '발송', '제품' 등 변별력이 떨어지는 단어들이 다수 포함되어 있었고, 이를 제거한 후 성능이 향상되었다. 하지만, 앞서 언급하였듯이 <Table 8>을 보면 변별력이 없어 보이는 단어를 선정해 F1-score가 낮아진 카테고리가 높아진 카테고리에 비해 더 많았다. 불용어 제거 후 F1-score가 하락한 카테고리에는 '연락', '수', '교환' 등의 단어가 상대적으로 다수 포함되어 있었다. 한편, '판매', '상태' 등 겹치는 불용어도 있는 것으로 보아 불용어 선정에 유의해야 하며, 불용어 제거가 단순하게 성능 향상을 보장하지는 않음을 알 수 있다.

<Table 9>는 keyword 속성에 대하여, 전처리 두 번째 단계부터 각 단계별로, 중분류 카테고리 기준으로 F1-score가 증가, 감소, 동일한 카테고리의 개수와 평균 증가율, 평균 감소율을 보여준다. 다른 속성인 name과 description과 달리 특수문자, 공백문자를 제거했을 때 F1-score가 하락한 카테고리가 상대적으로 더 많다. 띄어쓰기로 구분되고 주로 단어 단위로 등록되어 다른 속성에 비해 특수문자가 적고 무분별한 공백문자는 없었다. 성능이 향상된 카테고리도 있긴 하지만, F1-score가 하락한 카테고리가 더 많고 평균 증가율보다 감소율이 더 크며 <Table 6>을 보면 전체 성능도 감소했음을 알 수 있다. 형태소 분석기 적용과 불용어 제거에서는 name 속성과 비슷한 양상을 보였으나, 문장으로 구성된 다른 카테고리에 비해 형태소 분석기를 적용했을 때의 평균 증가율이 가장 낮다. 형태소 분석기를 적용했을 때 성능이 많이 향상된 것은 빅사이즈(0.1691), 썬케어(0.1443) 등이었고, 불용어를 제거했을 때 성능이 향상된 것은 솔로(남)(0.1047), 카오디오/영상(0.1020) 등이었다.

<Table 10>은 description 속성에 대하여, 전처리 두 번째 단계부터 중분류 카테고리 기준 단계별 F1-score가 증가, 감소, 동일한 카테고리의 개수와 평균 증가율, 평균 감소율을 보여준다. description은 상세 설명이어서 데이터의 길이가 길고 가장 자연어에 가까운 형태였다. 토큰의 개수가 매우 많아 특수문자 제거 전후의 빈도 차이는 작았지만, 공백 문자가 가장 많이 제거된 속성이었다. 또한, 형태소 분석기 적용 시 증가하고 감소한 카테고리의 개수는 다른 속성과 비슷하지만, 평균 증가율이 가장 높다. 이때 성능이 많이 향상된 카테고리는 인라인/스케이트(0.2204), 언더웨어/속옷(0.1581), 기타구기 스포츠(0.1198) 등이었다. description 데이터는 문장 비율이 높아 형태소 분석기의 영향을 많이 받았다. 4장의 데이터 전처리별 토큰수를 제시한 <Table 2>를 보면, name에서 336,808(77.1%)개, keyword에서 446,495(84.3%)개 만큼 줄어든 반면, description은 토큰이 1,608,432(88.7%)만큼 감소했음을 근거로 한다. 또한, 여러 개의 문장으로 구성된 만큼, 형태소 분석기를 적용했을 때 어미, 접사, 조사가 제거되어 명사의 빈도수가 높아졌다. 불용어 제거는 name, keyword와 같은 양상을 보였는데, 카오디오/영상(0.0615) 등은 name, keyword처럼 불용어를 제거했을 때 성능이 향상되었다.

〈Table 8〉 F1-Score Change Statistics By Each Preprocessing Step when the Input Attribute is ***Name***

preprocessing step / metric	Removing Special Characters and extra White Spaces	Stemming	Removing Stopwords
No of increased categories	68	92	44
No of decreased categories	50	26	74
No of unaffected categories	1	1	1
Average increase rate	0.0221	0.0431	0.0244
Average decrease rate	-0.0220	-0.0390	-0.0199

〈Table 9〉 F1-Score Change Statistics by Each Preprocessing Step when the Input Attribute is ***Keyword***

preprocessing step / metric	Removing Special Characters and extra White Spaces	Stemming	Removing Stopwords
No of increased categories	53	92	58
No of decreased categories	63	24	58
No of unaffected categories	3	3	3
Average increase rate	0.0178	0.0280	0.0157
Average decrease rate	-0.0210	-0.0247	-0.0173

〈Table 10〉 F1-Score Change Statistics by Each Preprocessing Step when the Input Attribute is ***Description***

preprocessing step / metric	Removing Special Characters and extra White Spaces	Stemming	Removing Stopwords
No of increased categories	79	90	42
No of decreased categories	39	28	76
No of unaffected categories	1	1	1
Average increase rate	0.0288	0.0451	0.0169
Average decrease rate	-0.0241	-0.0338	-0.0329

5. 결 론

본 연구는 CNN 딥러닝 모델을 사용하여 상품 데이터의 주류를 이루는 텍스트 데이터를 중심으로 전처리 과정과 입력 데이터로서 선택하는 속성에 따른 상품분류 모델의 성능 변화를 분석하였다. 먼저 사용한 입력 데이터의 개수 및 조합에 따른 변화를 비교했을 때, 세 가지 속성을 모두 사용하였을 때의 성능이 가장 좋았다. 또한, description이 name과 keyword에 비해 성능이 차이 나게 낮았음에도 다른 속성과 함께 사용했을 때 성능이 가장 많이 향상되었다. 입력데이터로 사용되는 속성이 늘어날 때마다 단일 입력일 때 성능이 높았던 카테고리들의 분류 성능이 다중 입력일 때도 높은 성능을 보이는 것을 확인하였다. 전처리를 적용했을 때 자연어에서는 특수문자, 공백문자 제거가 성능 향상에 도움이 되지만, keyword와 같이 오히려 성능이 떨어지는 경우가 있다는 것을 알 수 있었다. 모든 속성이 공통으로 가장

높은 성능을 보인 것은 형태소 분석기를 적용했을 때이다. 형태소 분석기 전후의 데이터를 비교한 결과, 교착어이면서, 띄어쓰기가 불분명한 한국어는 형태소 분석기를 적용했을 때 카테고리별로 상품의 특징을 드러내는 형태소의 빈도수가 향상된 것을 확인하였다. 반면, 불용어는 데이터를 직접 살펴 변별력이 없는 단어를 선정하더라도, 성능 향상을 쉽게 보장하기는 어렵다는 것을 알 수 있었다.

전자상거래 플랫폼이 활성화되면서 앞으로 더 많은 상품이 다양한 형태로 온라인에서 거래될 것이다. 복합 쇼핑몰은, 중고 거래와 같이 소비자와 판매자의 경계가 불분명한 곳에서, 자동 카테고리 분류 기능은 서비스 이용자 모두에게 큰 편의성을 제공한다. 상품 자동 분류는 전자상거래 분야에서 매우 중요한 기술이며, 플랫폼이 활성화됨에 따라 그 중요성이 더욱 커지고 있다. 딥러닝, 머신러닝의 다양한 모델을 활용하여 성능을 높이려는 노력은 계속되고 있지만, 사용하는 기술의 장점을 살리기 위해선 학습 데이터의 품질도 좋아야 한다. 본 연구가 도출한 실증 분석 및 결과는 딥러닝 상품 분류 모델 개발 시 성능 향상을 위한 레퍼런스로서 의의가 있다. 또한, 텍스트 데이터가 상품 정보의 주류를 이루는 만큼, 자연어 처리 기술을 더 발전시키면 같은 모델을 사용하더라도 더 좋은 성능을 보일 수 있을 것이다.

References

[1] Aanen, S. S., Vandic, D., and Frasincar, F., “Automated product taxonomy mapping in an e-commerce environment,” Expert Systems with Applications, Vol. 42, No. 3, pp. 1298-1313, 2015.

[2] Abels, S. and Hahn, A., “Automatic Classification and Re-Classification of Product Data in e-Business,” 2005 Symposium on Applications and the Internet Workshops (SAINT 2005 Workshops), pp. 350-353, 2005.

[3] Cortez, E., Rojas Herrera, M., da Silva, A. S., De Moura, E. S., and Neubert, M., “Lightweight Methods for Large-Scale Product Categorization,” Journal of the American Society for Information Science & Technology, Vol. 62, No. 9, pp. 1839-1848, 2011.

[4] Dalal, M. K. and Zaveri, M. A., “Automatic Text Classification: a Technical Review,” International Journal of Computer Applications, Vol. 28, No. 2, pp. 37-40, 2011.

[5] Goumy, S. and Mejri, M.-A., “Ecommerce Product Title Classification,” In SIGIR 2018 Workshop on eCommerce, 2018.

[6] Ha, J. W., Pyo, H., and Kim, J., “Large-scale item categorization in e-commerce using multiple recurrent neural networks,” In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 107-115, 2016.

[7] Khoo, A., Marom, Y., and Albrecht, D., “Experiments with sentence classification,” Proceedings of the Australasian Language Technology Workshop 2006, pp.

18-25, 2006.

[8] Kil, H.-H., “The Study of Korean Stop-words list for Text mining,” URIMAL-GEUL : The Korean Language and Literature, Vol. 78, pp. 1-25, 2018.

[9] Kim, Y., “Convolutional neural networks for sentence classification,” Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1746-1751, 2014.

[10] Kozareva, Z., “Everyone likes shopping! multi-class product categorization for e-commerce,” Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1329-1333, 2015,

[11] Krishnan, A. and Amarthaluri, A., “Large Scale Product Categorization using Structured and Unstructured Attributes,” KDD ’19. Mar 2019.

[12] LeCun, Y., Bengio, Y., and Hinton, G., “Deep Learning,” Nature, Vol. 521, pp. 436-444, 2015.

[13] Lee, D. J., Hwang, I. B., and Lee, S.-G., “Efficient Management of Statistical Information of Keywords on E-Catalogs,” The Jounal of Society for e-Business Studies, Vol. 14, No. 4, pp. 1-17, 2009.

[14] Lee, T., Lee, I.-H., Lee, S. K., Lee, S.-G., Kim, D. K., Chun, J. H., Lee, H., and Shim, J. H., “Building an Operational Product Ontology System,” Electronic Commerce Research and Applications, Vol. 5, No. 1, pp. 16-28, 2006.

[15] Lin, Y.-C., Das, P., Trotman, A., and Kallumadi, S., “A Dataset and Baselines for e-Commerce Product Categorization,” 2019 ACM SIGIR International Conference on Theory of Information Retrieval, pp. 213-216, 2019.

[16] Shen, D., Ruvini, J.-D., and Sarwar, B., “Large-scale item categorization for e-commerce,” Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM '12), pp. 595-604, 2012.

[17] Shen, D., Ruvini, J.-D., Mukherjee, R., and Sundaresan, N., “A study of smoothing algorithms for item categorization on e-commerce sites,” Neurocomputing, Vol. 92, pp. 54-60, 2012.

[18] Skinner, M., “Product Categorization with LSTMs and Balanced Pooling Views,” In SIGIR 2018 Workshop on eCommerce, 2018.

[19] Suzuki, S., Iseki, Y., Shiino, H., Zhang, H., Iwamoto, A., and Takahashi, F., “Convolutional Neural Network and Bidirectional LSTM Based Taxonomy Classification Using External Dataset at SIGIR eCom Data Challenge,” In SIGIR 2018 Workshop on eCommerce, 2018.

[20] Xia, Y., Levine, A., Das, P., Di Fabbrizio, G., Shinzato, K., and Datta, A., “Large-Scale Categorization of Japanese Product Titles Using Neural Attention Models,” In Proceedings of the 15th Conference of

the European Chapter of the Association for Computational Linguistics, Vol. 2, pp. 663-668, 2017.

[21] Yeon, J. H., Lee, D. J., Shim, J. H., and Lee, S.-G., "Product Review Data and Sentiment Analytical Processing Modeling," The Jounal of Society for e-Business Studies, Vol. 16, No. 4, pp. 125-137, 2011.

[22] Yu, W., Sun, Z., Liu, H., Li, Z., and Zheng, Z., "Multi-level Deep Learning based E-commerce Product Categorization," In SIGIR 2018 Workshop on eCommerce, 2018.

[23] Zahavy, T., Magnani, A., Krishnan, A., and Mannor, S., "Is a picture worth a thousand words? A Deep Multi-Modal Fusion Architecture for Product Classification in e-commerce," CoRR, Vol. abs/1611.09534, 2016.

저 자 소 개

이나경 (E-mail: lnk7424@sookmyung.ac.kr)
2017년~현재 숙명여자대학교 컴퓨터과학전공 (학부 재학중)
관심분야 딥러닝, 데이터베이스, 웹 등

김주연 (E-mail: parang@sookmyung.ac.kr)
2017년~현재 숙명여자대학교 컴퓨터과학전공 (학부 재학중)
관심분야 딥러닝, 머신러닝 등

심준호 (E-mail: jshim@sookmyung.ac.kr)
1986년~1990년 서울대학교 계산통계학과 (학사)
1990년~1994년 서울대학교 계산통계학과 전산과학전공 (석사)
1994년~1998년 Northwestern University, Electrical & Computer Engineering (박사)
1999년~2001년 Drexel University, College of Information Science and Technology 조교수
2001년~현재 숙명여자대학교 소프트웨어학부 교수
관심분야 데이타베이스, 빅데이터, e-비즈니스 기반기술, 웹