OPEN ACCESS

기술과학 분야 문헌의 DDC 자동 분류에 관한 연구*

A Study on Automatic DDC Classification of Documents in Technology

강 우 진 (Woojin Kang)**
나 상 오 (Sango Na)***
이 종 욱 (Jongwook Lee)****

목 차

1. 서 론
2. 개념적 배경
3. 연구 설계
4. 연구 결과
5. 논의 및 결론

초 록

본 연구는 기계학습 모델을 활용하여 DDC 기술과학(600) 분야 문헌의 자동 분류를 수행하고, 제목 중심 접근의 한계를 보완하기 위해 문헌 설명 정보를 분류 자질로 추가하여 그 효과를 검증하였다. 이를 위해 Omkuji, FastText, BERT 총 3가지 기계학습 모델을 기반으로 분류모델을 설계하였으며, 강목과 요목 수준의 분류기호를 예측하여 모델별・수준별(강목, 요목) 결과를 비교 분석하였다. 성능 평가 지표로는 정확도(Accuracy)와 F_1-score를 사용하였다. 분석 결과 BERT-FastText-Omikuji 순으로 우수한 성능을 보였으며, Omikuji 분류모델의 요목 수준 F_1-score를 제외한 모든 조건에서 설명 정보를 추가했을 때 성능이 향상됨을 확인하였다. 특히 문헌 설명 정보를 추가한 후 BERT 기반 분류를 수행하였을 때, 요목 수준에서 79.52%의 정확도를 기록하여 선행연구와 비교하여 약 8.62%p의 성능 향상을 확인하였다. 이 밖에도 분류 자질의 차이뿐만 아니라, 학습에 활용되는 문헌의 양이 많을수록 분류 성능이 전반적으로 향상되는 경향이 확인되었다. 이러한 결과는 자동분류 성능의 지속적인 개선을 위해 학습 문헌의 양적 확보와 함께 분류 자질 정보의 질적 보완이 병행되어야 함을 시사한다. 후속 연구에서는 분석 대상 주제 범위를 DDC 전 분야로 확대하고, 국내 도서관에서 널리 활용되는 KDC에도 적용 가능성을 검토할 필요가 있다. 나아가 다양한 분류모델을 추가로 적용하고 분류 자질을 확장함으로써, 이에 따른 분류 성능의 변화를 살펴볼 필요가 있을 것이다.

ABSTRACT

This study investigates the automatic classification of documents in the Dewey Decimal Classification (DDC) Technology class (600) using machine learning models, with the aim of overcoming the limitations of title-based classification approaches. To enhance classification performance, descriptive document information, such as summaries and introductions, was incorporated as additional classification features. Three machine learning models—Omikuji, FastText, and BERT—were employed, and classification performance was evaluated at both the main class and division levels. Accuracy and F_1-score were used as evaluation metrics. The results demonstrate that BERT consistently outperformed FastText and Omikuji across most experimental conditions. With the exception of the division-level F_1-score of the Omikuji model, all models showed improved performance when descriptive information was added. In particular, the BERT-based model achieved an accuracy of 79.52% at the division level, representing an improvement of approximately 8.62 percentage points compared to previous studies. The findings also indicate that classification performance generally improves as the volume of documents used in model training increases, underscoring the importance of data scale in addition to feature selection. These results suggest that competitive automatic classification performance can be achieved through appropriate model selection and enriched classification features, even within single-model approaches. Future research should expand the scope to all DDC classes and examine the applicability of the proposed approach to the Korean Decimal Classification (KDC), as well as explore additional features and alternative machine learning models.

키워드: 자동분류, 듀이십진분류법(DDC), 기술과학 분야, 기계학습, 분류 자질
Automatic Classification, Dewey Decimal Classification (DDC), Technology, Machine Learning, Classification Features

* 본 연구는 국회도서관의『인공지능 시대 국회도서관 지식정보 구조화 방안 연구(2024)』의 일부를 수정・보완한 것임.
** 경북대학교 대학원 문헌정보학과 박사과정(rkddnwls1234@knu.ac.kr / ISNI 0000 0005 0659 7247) (제1저자)
*** 경북대학교 대학원 문헌정보학과 석박사통합과정(gg4518@knu.ac.kr / ISNI 0000 0005 0490 1663) (공동저자)
**** 경북대학교 문헌정보학과 부교수(jongwook@knu.ac.kr / ISNI 0000 0004 6830 6145) (교신저자)
논문접수일자: 2026년 1월 20일 최초심사일자: 2026년 1월 29일 게재확정일자: 2026년 2월 3일
한국문헌정보학회지, 60(1): 173-194, 2026. http://dx.doi.org/10.4275/KSLIS.2026.60.1.173

※ Copyright © 2026 Korean Society for Library and Information Science
This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 (https://creativecommons.org/licenses/by-nc-nd/4.0/) which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.

www.kci.go.kr

1. 서 론

정보자원의 양이 급증하고 있는 가운데 이들 자원에 대한 주제 접근을 효율적으로 제공하기 위해 자동화된 주제 색인의 필요성이 확대되고 있다. 자동 주제색인은 주제 접근을 효율화하는 것 외에도 기존 메타데이터를 풍부하게 하고, 색인의 일관성을 높이는 데 유용할 수 있다. 메타데이터의 풍부화는 정보자원 간의 연결성을 강화시키는 데도 도움이 된다(노지현 외, 2023). 이는 곧 정보자원에 대한 검색 가능성과 접근성을 높이는 데 기여하는 것이다(Golub, 2021). 특히, 주제명표목표나 시소러스, 분류표와 같은 기존의 정보조직 체계를 활용하여 접근점을 부여하는 것은 이용자의 요구에 부합하는 자료를 효과적으로 탐색하는 데 중요한 역할을 한다. 통제어휘로 작성된 접근점을 부여함으로써 이용자로 하여금 혼란을 줄이고, 필요한 정보자원을 명확히 식별하고 선택할 수 있도록 하는 것이다(노지현 외, 2023). 이에 많은 도서관에서는 입수된 자료의 서지 레코드에 주제명이나 디스크립터, 분류기호를 부여하고 있다.

도서관 자료에 부여되는 분류기호는 자료의 주제를 제시하여, 주제 접근점으로 활용되며, 서가 상의 자료 배열의 기준이 되기도 한다. 특히, 분류기호에 따른 자료 배가는 유사한 주제의 자료를 한 데 묶어줌으로써 이용자의 브라우징(Browsing)에도 유익하게 활용되고 있다. 우리나라 도서관에서는 자료 분류에 있어 한국십진분류법(Korean Decimal Classification, 이하 KDC) 또는 듀이십진분류법(Dewey Decimal Classification, 이하 DDC) 등을 주로 사용한다. KDC는 국내 공공도서관에서 주로 사용하고 있으며 DDC는 국가도서관이나 대학도서관 등에서 빈번하게 적용되고 있다(곽철완, 2014).

국내외에서는 도서관 자료에 대한 분류 기호 부여 업무를 자동화하기 위한 노력이 꾸준히 이루어지고 있다. 문헌의 내용을 대상으로 한 자동분류 연구는 1960년대부터 시작되었으며, 국내 문헌정보학 분야에서는 1980년대 초반부터 연구가 수행되었다(김정현, 2011). 1990년대부터는 컴퓨터의 성능이 향상되며 기계학습(Machine Learning, 이하 ML)이 문헌 자동분류 연구에 도입되기 시작하였고, 이때부터 실험적인 수준을 넘어서 활발한 연구가 이루어져 왔다(김판준, 2022; 정영미, 2012; Sebastiani, 2002). 최근에는 해외에서 생성형 AI와 기계학습을 활용한 연구가 이루어지고 있다. 스웨덴, 핀란드와 같은 유럽 국가에서는 ML 기반의 서지 데이터 자동분류 서비스를 발표하고 이에 대한 성능을 평가하는 연구가 이루어졌다. 구체적으로 단일 알고리즘을 적용하는 것뿐만 아니라, 이를 결합하는 기법을 활용하여 Annif라는 새로운 모델을 개발하고, 실제 서비스 운영에 일부 도입하여 실용성을 입증한 사례를 확인할 수 있다(Golub et al., 2024; Suominen et al., 2022).

테스트 단계의 활용이지만 생성형 AI의 사용도 시도되고 있다. ChatGPT, Copilot, Gemini와 같은 AI에 도서 분류기호(DDC, LCC)와 주제명 색인을 요청하고, 이를 기존의 목록 메타데이터와 비교하는 시도가 주를 이루고 있다(Bodenhamer, 2023; Noruzi, 2024). 하지만 생성형 AI를 사용한 자동분류는 아직까지 실효성이 부족하다는 한계가 있다. 문헌정보학 분야의 업무에서 생성형 AI는 작업 효율을 높이는 보조도구로서 충분히 가치 있지만 그 결과물로 사실

www.kci.go.kr

이 아닌 정보를 생성하는 환각(Hallucination) 현상이 종종 발생하며, 이로 인해 전문가의 검증이 필수적이라는 평가를 받고 있다(Formanek, 2024). 특히 기존의 도서관 메타데이터(분류기호, 주제명)와 생성형 AI의 자동분류 결과물을 비교하였을 때, 분류기호 값은 일치율이 현저히 떨어져 단독으로 활용하기 어렵다는 지적이 있다(Noruzi, 2024). 즉, 생성형 AI를 활용한 분류기호 생성 시도는 있으나 보조표를 활용한 분류기호 조합 방식의 복잡성과 규칙을 정확히 반영하지 못하는 등의 한계로 아직까지는 ML 모델을 적용한 실험과 사례가 주요하다.

이에 본 연구에서는 문헌에 대한 DDC 분류기호를 자동으로 생성하기 위하여 기계학습(ML) 모델을 적용하였다. 기존 연구나 서비스에서 자주 활용되어 온 FastText와 Omikuji 모델을 포함하여 최근에 개발된 딥러닝 기반 언어모델인 BERT를 활용하여 분류 성능을 비교, 분석하였다. 선행연구에서는 주로 문헌의 제목이나 목차를 분류 자질로 활용해 왔으나(이용구, 2020; 이용구, 2023a), 본 연구에서는 문헌의 요약이나 소개와 같은 설명 정보를 외부 데이터와 결합하여 보완하고 이를 분류 자질로 활용하였다. 또한 분석 대상은 기존 DDC 자동분류 연구(Golub et al., 2024)에서 가장 좋은 성능을 보인 바 있는 '기술과학'(600) 분야로 설정하였다.

2. 개념적 배경

2.1 기계학습 기반 문헌 자동 분류

문헌 자동 분류 설계에 활용되는 대표적인 모델로는 Omikuji, FastText, BERT(Bidirectional Encoder Representations from Transformers) 등이 있다(Suominen et al., 2022; 김인후, 김성희, 2022; 이용구, 2020; 2023a; 2023b). 먼저, Omikuji는 대규모 다중 레이블 분류를 지원하기 위해 개발된 Python 라이브러리로, 그 핵심에는 Parabel 분류기가 있다. Parabel은 전통적인 다중 클래스 분류 방식 중 하나인 OvA (One-vs-All) 분류기의 한계를 보완하기 위해 제안된 방법이다. OvA 분류기는 각 범주에 속할 확률을 개별적으로 학습, 예측하기 때문에 높은 정확도를 가지지만, 범주의 수가 증가할수록 훈련 시간과 예측 시간이 비례적으로 증가하여 범주가 많은 대규모 분류에서는 비효율적이라는 문제가 있다. 이를 해결하기 위해 Parabel은 계층형 이진 트리 구조를 도입하였다. 즉, 유사한 범주들을 묶어 두 개의 그룹으로 나눈 뒤, 다시 각 그룹을 이진 분할하는 과정을 반복하여 트리를 구축한다. 이렇게 하면 범주 수가 매우 많더라도 OvA 방식과 비슷한 수준의 정확도를 유지하면서도 훈련 및 예측 시간을 획기적으로 줄일 수 있다(Prabhu et al., 2018). Omikuji에서 이러한 트리 구조를 구현할 때는 범주 간 유사도를 고려하는데, 이는 주로 문헌의 내용을 토큰화(tokenization)한 뒤 TF-IDF 가중치를 적용하여 만든 문서-단어 행렬의 벡터값을 기반으로 계산한다.

Omikuji는 분류모델 구조에 트리 구조를 도입한 것이 특징이라면 FastText와 BERT를 활용한 분류는 문헌을 벡터화하는 방법, 즉 분류 자질 표현 방식이 특징적이다. 기존의 벡터화 방식은 Omikuji 분류기와 같이, 문헌 내 텍스트를 문장이나 단어 단위로 토큰화한 뒤 TF-IDF

www.kci.go.kr

가중치를 적용하여 문서-단어 행렬로 변환하는 방법이 주로 활용되었다. 하지만 문헌 집합 내 토큰 수가 늘어날수록 차원이 커지고, 기존의 문서-단어 행렬은 대부분의 값이 '0'인 희소행렬(sparse matrix)이어서 자원 소모와 학습 및 예측 시간이 증가하는 문제가 있다. 이에 비해 FastText와 BERT는 각 단어를 개별 차원으로 표현하는 대신에 토큰의 출현 정보를 바탕으로 대부분의 차원이 '0'이 아닌 밀집 벡터(dense vector) 형태로 학습한다. 이후 학습된 토큰 벡터들을 조합하여 문서를 하나의 벡터로 표현하며, 이러한 방식은 희소행렬 문제를 완화할 뿐 아니라 새로운 토큰이 등장해도 기존의 문서 벡터 차원의 크기를 유지할 수 있다.

FastText는 Facebook(현 Meta)에서 개발한 라이브러리로 Word2Vec의 단어 임베딩 방식의 한계를 보완한 모델로 단어를 n-그램(n-gram) 단위의 하위 단어(subword)로 분해하여 학습한다. 이를 통해 복수형이나 접사 변형 등 단어의 형태 변화를 보다 유연하게 처리할 수 있는 장점이 있다. FastText에서 얻어진 임베딩 벡터는 선형 모델(Linear Model)과 같은 분류기와 결합하여 분류모델을 구성할 수 있다(Joulin et al., 2016).

BERT는 구글(Google)에서 개발한 트랜스포머(Transformer) 아키텍쳐 기반 딥러닝 언어모델이다. FastText가 주어진 학습 데이터에서 임베딩을 직접 학습하는 것과 달리, BERT는 대규모 말뭉치(Corpus)에서 사전학습된 언어모델(Pre-trained Language Model, PLM)을 활용한다(Devlin et al., 2017). 이후 특정 과제에 맞게 미세조정(Fine-tuning) 과정을 거쳐 분류에 적용된다. BERT는 입력 단어를 워드피스(WordPiece) 알고리즘을 사용해 하위 토큰으로 분해한 뒤, 각 토큰에 대해 토큰 임베딩, 세그먼트 임베딩, 위치 임베딩을 더하여 트랜스포머의 셀프 어텐션(self-attention) 구조로 문맥 정보를 학습한다. 이를 통해 기존의 문서-단어 행렬 기반 Bag of Words(BoW) 방식이나 FastText보다 더 깊은 맥락을 반영한 벡터 표현을 생성할 수 있다. 마지막으로, 이렇게 얻어진 임베딩 벡터는 선형 모델이나 신경망 분류기(Neural Network Classifier)과 같은 분류기와 결합되며, 출력층에서는 소프트맥스(softmax) 또는 시그모이드(sigmoid) 함수를 통해 범주별 확률을 예측하는 방법으로 BERT 기반 분류모델을 구축하여 활용한다.

2.2 문헌 자동분류 최근 연구 동향

최근 해외에서 이루어진 연구는 도서 분류기호와 주제명(Subject Headings, SH)을 자동 부여하려는 시도가 주를 이루고 있으며, ML 모델을 주요하게 활용하고 있으며, 최근 발전한 생성형 AI를 점차 적용하는 방향으로 전개되고 있다. 기계학습 모델로는 Suominen et al.(2022)이 개발한 Annif가 대표적이다. 핀란드를 비롯하여 스웨덴, 독일 등 다양한 국가의 국립도서관에서 사용되는 Annif는 여러 단일 모델(예: 어휘 기반의 Maui, MLLM 및 연관성 기반의 FastText, Omikuji)을 통합하고, 이들의 결과를 결합하는 앙상블(Ensemble) 기법을 사용한다. 이는 다양한 문서 유형과 데이터셋, 분류 체계와 관계없이 대부분의 상황에서 단일 모델보다 나은 성능을 보였으며, 핀란드의 대학 리포지터리와 도서 유통사 등 실제 환

www.kci.go.kr

경에 성공적으로 도입되어 실용성 또한 입증되었다.

Golub et al.(2024)은 Annif의 앙상블 기법을 스웨덴 국가 종합목록(Libris)의 대규모 서지 데이터(약 23만건)에 적용하여 DDC 분류기호 자동 부여를 시도하고 그 실용적 가치를 입증하고자 하였다. 연구 결과, 요목 수준에서 62.83%의 정확도를 달성하였으며, 특히 서지 데이터에 주제 정보가 포함되어 있는 경우에는 최대 66.82%까지 정확도가 향상된 것을 확인하였다. 주류별로 구분하여 성능을 확인한 결과 600(기술과학)이 70.9%으로 가장 높은 정확도를 보인 반면, 200(철학 및 심리학)은 56.2%로 가장 낮은 성능이 확인되었다.

생성형 AI를 활용한 자동분류 또한 시도되고 있다. Bodenhamer(2023)는 생성형 AI(ChatGPT)의 도서관 메타데이터(DDC 분류기호, LCSH 주제명, 키워드) 생성 능력을 평가하기 위해 표준화된 질의 목록을 4개월간 반복적으로 요청하였다. 그 결과, 주제명은 LCSH와 비교적 유사하게 나타났으나, DDC 분류기호에서는 많은 오류가 발생하여 작업 종류에 따라 성능 편차가 큼을 확인하였다. Noruzi(2024)는 4권의 오픈액세스 도서를 대상으로 AI(ChatGPT, Copilot, Gemini)가 부여한 주제명(LCSH)과 분류기호(LCC, DDC)를 미국 의회도서관의 기존 목록과 비교하였다. AI가 부여한 값들은 주제적 관련성은 있었지만, 사람이 부여한 값과의 일치성은 매우 낮아, 생성형 AI를 단독으로 활용하기에는 한계가 있다는 점을 지적하였다.

국내에서도 서지 데이터를 활용하여 자료의 주제를 자동으로 부여하는 자동분류 연구가 다수 수행되었다. 일반적으로 자동분류 연구는 (1) 분류 대상인 '범주', (2) 분류를 위해 활용되는 자료의 속성인 '분류자질', (3) 분류자질을 이용하여 범주를 예측하는 '분류기' 세 가지 요소를 중심으로 이루어진다.

우선 범주의 경우, 자료에 할당되었던 분류기호가 주로 활용되었다. 대표적으로 국립중앙도서관 및 공공도서관에서 사용하는 KDC, 대학도서관에서 활용하는 DDC, 그리고 주제명표목표가 주요한 범주로 활용되었다(이용구, 2020; 2023a; 2023b). 이외에도 정부 기록물을 대상으로 정부기능분류체계에 따른 단위과제 자동분류 연구(김영범, 장우권, 2023)와 학술논문을 학술연구분야 분류표에 따라 자동분류하는 연구(김인후, 김성희, 2022) 등이 수행된 바 있다.

분류자질로는 자료의 내용을 파악할 수 있는 속성(예: 제목, 초록, 목차)이 주로 활용되었으며, 일부 연구(김영범, 장우권, 2023)에서는 외적인 요소(예: 기관명, 담당자명, 생산일자)도 포함하였다. 도서 자동분류의 경우, 도서관이 자체적으로 확보할 수 있는 서지 정보 중 주제를 반영할 수 있는 요소로 제목을 활용한 연구가 주를 이루고 있다(이용구, 2023a; 2023b). 그러나 제목의 길이가 짧다는 한계를 보완하기 위해 목차 정보를 추가하여 학습하는 연구도 진행되었으며, 이 과정에서 성능 향상도 확인된 바 있다(이용구, 2020). 학술논문의 경우 제목뿐만 아니라 초록(Abstract)을 분류자질로 활용하는 연구가 수행되었으며, 일부 연구에서는 기관명, 관리부서명 등 외적 요소를 포함한 자동분류도 시도되었다(김영범, 장우권, 2023).

분류자질의 표현(Feature representation) 방식은 대부분 텍스트 데이터의 수치화를 기반으로 한다. 전통적으로는 TF-IDF 가중치 기반의

www.kci.go.kr

벡터화 방법이 적용되었으며(김영범, 장우권, 2023), 최근에는 자연어처리(Natural Language Processing, NLP) 기법의 발전에 따라 딥러닝 기반 임베딩 기법이 점차 적극적으로 활용되고 있다(김인후, 김성희, 2022; 이용구, 2020; 2023a; 2023b). 임베딩 기법 중에서는 Word2Vec, GloVe (Global Vectors), FastText가 제목과 같은 짧은 텍스트에 적용되었으며(이용구, 2023a), 긴 텍스트에서는 텍스트 내 문맥 반영에 좋은 성능을 가지고 있는 트랜스포머 기반 BERT가 활용되었다(김인후, 김성희, 2022; 이용구, 2020; 2023b).

분류기로는 다양한 기계학습 및 딥러닝 기반 모델이 적용되었다. 전통적으로 k-NN(k-Nearest Neighbor) 분류기가 활용되었으며(이용구, 2020; 2023a), 추가적으로 SVM(Support Vector Machine), 나이브 베이즈(Naive Bayes), 결정트리(Decision Tree), 랜덤 포레스트(Random Forest), XGBoost 등의 기계학습 모델이 적용되었다(김영범, 장우권, 2023). 한편, BERT 기반 연구에서는 BERT 임베딩 값을 활용하여 미세조정하거나 딥러닝 분류기에 적용하여 성능을 평가하는 방식이 사용되었다(김인후, 김성희, 2022; 이용구, 2023b).

이처럼 자동분류 연구는 최근 들어 딥러닝과 자연어처리 기술이 빠르게 발전하면서 보다 정교한 자동분류 모델을 적용하고 있으며, 기존의 TF-IDF 기반 방식에 비해 임베딩 기법을 적용하는 연구가 활발하게 진행되고 있다.

3. 연구 설계

본 연구에서는 문헌의 제목에 설명 정보(요약, 소개)를 추가한 텍스트를 분류자질로 활용하고, DDC 분류기호를 예측 대상으로 하여 분류기를 학습하였다. 특히, 문헌의 제목만을 활용한 경우와 제목에 문헌에 관한 설명 정보를 보강한 사례를 구분하였다. 예측 범주는 강목과 요목 수준으로 설정하여 분석을 수행하였다. 자동분류 모델은 선행연구를 바탕으로 Omikuji, FastText, BERT를 활용하였다.

3.1 데이터 개요

자동분류 관련 선행연구(Golub et al., 2024)에서 DDC 기술과학(600) 분야가 가장 우수한 분류 성능을 보인 바 있다. 특히, Golub et al. (2024)의 연구에서는 600 분야는 타 주제분야에 비해 통제어휘 기반의 주제명이 명확히 부여되어 있으며, 레코드의 수가 많은 편이어서 DDC 요목 수준에서 가장 높은 정확도(70.9%)를 기록하였다. 이러한 결과에 근거하여 본 연구에서는 연구 범위를 DDC 600 분야로 설정하였으며, 이를 위해 국회도서관으로부터 해당 분야로 분류된 문헌의 서지 레코드 데이터(270,729건)를 MARC(MAchine Readable Catalog) XML(eXtensible Markup Language) 형태로 입수하여 분석을 진행하였다.

해당 데이터에서 예측 범주인 DDC 분류기호는 082(듀이십진분류기호) 필드에서 추출하였다. 분류자질로 활용될 수 있는 문헌의 제목은 245(책임표시사항) 필드에서 추출하였다. MARC 레코드에는 제목은 입력되어 있으나, 내용에 관한 요약이나 소개 등과 같은 정보는 거의 입력되어 있지 않았다. 그렇지만 이러한 제목은 간결하게 작성되어 있어 학습에 있어서

www.kci.go.kr

충분한 정보를 제공할 수 없는 한계가 존재한다(Peng et al., 2021). 따라서 본 연구에서 자동분류 모델의 성능 향상을 위해 문헌에 대한 설명 정보를 추가하고자 하였다. 이를 위해 각 MARC 레코드에서 020(국제표준도서번호) 필드에서 ISBN(International Standard Book Number)을 추출한 뒤, 이를 기반으로 외부 데이터를 결합하여 자료에 대한 부가적인 설명(요약, 소개 등) 정보를 추가하였다.

구체적으로 한국문화정보원의 '문화 빅데이터 플랫폼'(https://www.bigdata-culture.kr/)에서 국립중앙도서관이 제공하는 '도서별 상세정보' 데이터를 수집하여 MARC 레코드에서 추출한 ISBN 정보와 매칭하여 문헌 설명 정보를 추가하였다. 하지만 2022년부터 '도서별 상세정보'에 대한 추가적인 데이터 업로드가 중단된 상황임에 따라 2022년 이후 발행된 문헌에 대한 설명자료를 추가할 수는 없었다. 이에 문화체육관광부의 '도서관 정보나루'에서 제공하는 '도서 상세 조회 API(Application Protocol Interface)'를 활용하여 2022년 이후 발행된 문헌에 대한 설명 정보를 보완하였다. 이러한 과정을 거쳐 112,569권의 문헌에 대한 설명 정보를 추가할 수 있었다(〈표 1〉 참조).

3.2 데이터 전처리

앞서 수집한 데이터에서 분류기를 적용하기 위해서는 우선 분류자질에 해당하는 텍스트를 모두 벡터 형태로 변환할 필요가 있으며, 동시에 잡음(noise), 불용어(stopwords) 등을 제거하여 효과적이고 정확한 결과를 얻을 수 있도록 해야 한다. 이를 위해 분류자질로 활용되는 문헌의 제목과 설명 정보에서 특수문자, 숫자, 한자 등을 모두 제거하였다. 이는 불필요한 정보가 문헌을 표현하고 분류하는 과정에서 악영향을 미치는 것을 방지하기 위함이다.

본 연구에서 적용한 BERT는 사전학습모델에 포함된 토크나이저(tokenizer)를 활용하므로 별도의 토큰화 과정이 필요하지 않다. 반면, Omikuji와 FastText와 같은 다른 모델의 경우 텍스트를 벡터화하기 위해 분류기 적용 전에 사전에 토큰화 과정이 요구된다. 이에 본 연구에서는 Python의 Mecab 형태소 분석기를 활용하여 전치사와 접속사 등을 불용어로 처리하여 제외하고, 명사, 형용사, 동사에 해당하는 토큰만을 추출하여 분석에 활용하였다.

이후 문헌의 제목과 설명 정보를 결합한 후, 전체 길이가 10자 이하이거나 토큰화 결과 유효한 토큰이 존재하지 않는 경우를 제외하였다.

〈표 1〉 문헌 설명 정보 추가 예시

ISBN	제목	설명
9788937810398	영리한 눈	'머리에서 발끝까지'는 생활 속에서 겪는 경험과 흥미로운 정보, 그리고 재미있는 그...
9788901044002	대전 세계 엑스포, 감동과 환희	대전 세계 엑스포 개최 10주년 기념 회고집. 대전 세계 엑스포 조직위원장을 맡았던...
9788976332721	지식재산 전쟁	캐논, 히타치, 제록스 등 선진기업들의 특허경영과 최근 벌어진 PDP, 반도체의 특...

www.kci.go.kr

이는 문헌의 분류에 활용할 자질이 부족하거나 제목 및 설명만으로 맥락을 파악하기 어려운 텍스트를 분석 대상에서 배제하기 위함이다. 그 결과, 설명 정보가 포함되어 내용에 대한 맥락 정보가 존재하는 문헌(86,663건, 65.7%)과 설명 정보는 없으나 제목의 길이가 충분하여 일정 수준의 맥락을 제공한다고 판단되는 문헌(45,357건, 34.3%)을 포함하여 총 132,020건의 문헌으로 구성된 데이터셋을 구축하였다.

본 데이터셋은 분류기호별 문헌 수가 균등하지 않은 문제가 있어 오버샘플링(Oversampling)과 언더샘플링(Undersampling) 기법을 고려하였다. 그렇지만 언더샘플링을 강목 수준에 적용하는 경우 전체 데이터의 약 6%만 활용 가능하며, 오버샘플링은 레코드 수가 적은 분류기호의 샘플 수를 증가시키는 과정에 특정 패턴이 과도하게 학습되는 과적합(overfitting)의 위험이 존재한다. 이에 본 연구에서는 데이터의 손실과 과적합을 방지하기 위해 불균형한 분포를 유지한 상태로 분석을 수행하였다.

3.3 자동 분류모델 적용

본 연구에서 적용한 자동 분류모델은 3개로 각각 Omikuji, FastText 그리고 BERT를 활용하였다. 이들은 지도학습(supervised learning) 기법의 일종으로 학습 문헌과 실험 문헌에 대한 분리가 필요하다. 이에 본 연구에서는 전체 문헌을 학습 문헌과 실험 문헌을 9:1의 비율로 분리하였으며, 이 과정에서 문헌의 분류기호 분포를 고려하였다. DDC 분류기호 강목 수준에서는 10개의 강목에 대해 132,020건의 데이터셋을 활용하였다. 요목 수준에서는 97개 요목 중 3개 요목이 각각 1건의 문헌만을 포함하고 있어, 이들을 제외한 후 94개의 요목에 대한 132,017건의 자료를 자동 분류 분석에 활용하였다.

Omikuji를 적용하기 위해 전처리 단계에서 수행한 토큰화 결과를 TF-IDF 가중치를 적용하여 분류자질로 변환하였다. 이때 분석의 효과성을 높이기 위하여 전체 문서의 50% 이상 등장하는 고빈도 단어를 제거하였으며, 이후 상위 빈도 5,000개의 단어를 제외한 나머지 단어는 저빈도 단어 즉, 불용어로 간주하여 제거하였다. FastText 모델에서는 각 문헌의 토큰을 300차원의 벡터로 변환한 후, 이들의 평균값(Average pooling)을 산출하여 문헌 단위의 벡터 표현을 도출하였다. 이렇게 생성된 문헌 임베딩은 임베딩 결과에 선형 계층과 소프트맥스 함수를 결합한 분류기에 입력되어 각 분류기호에 대한 확률 분포를 산출하도록 하였다(Joulin et al., 2016). 모델 학습은 기본 설정된 파라미터 값을 활용하였으며, 학습 에포크(epoch) 5회를 적용하였다.

BERT는 사전학습모델의 종류에 따라 영문뿐 아니라 국문 등 다양한 언어의 텍스트 처리에 대응할 수 있다(이용구, 2023b). 본 연구에서는 다국어 BERT 사전학습모델인 'bert-base-multilingual-uncased'를 활용하여 문헌의 임베딩 값을 추출하였다. 토큰화 과정은 별도의 추가 작업 없이 해당 사전학습모델에서 제공하는 토크나이저를 사용하여 원문 텍스트를 입력하는 방식으로 수행하였다. 자동 분류를 위해서는 앞서 FastText와 같이 BERT에서 추출한 임베딩 벡터를 입력으로 하는 신경망 기반 분류기를 적용하였다. 입력 텍스트는 BERT의

www.kci.go.kr

임베딩 레이어를 거쳐 벡터화되며, 변환된 임베딩 벡터는 선형 계층에 연결된다. 이후 출력 계층에는 분류 대상인 분류기호을 배치하여 학습하는 미세조정을 수행하였다. 또한 다중 클래스 분류를 위해 선형 계층의 출력에 소프트맥스 함수를 적용하여 각 클래스에 속할 확률을 산출하고, 이들의 합이 '1'이 되도록 정규화하였다. BERT 기반 신경망 분류기 학습 시에도 앞선 FastText 분류기와 동일하게 학습 에포크 5회를 적용하였다.

3.4 자동 분류 결과 평가

학습 문헌을 통해 DDC 분류기호를 학습한 후, 실험 문헌의 텍스트(제목 및 설명 정보)를 분류기에 입력하여 분류기호를 예측한 뒤, 해당 결과에 대하여 분류모델의 성능을 평가하기 위해 분류기호별 F_1-score, F_1-score(macro)와 정확도(Accuracy)를 지표로 활용하였다.

해당 지표를 산출하기 위해서는 예측하고자 하는 범주에 대해 실제 범주와 예측된 범주를 비교하여 2×2 형태의 혼동행렬(Confusion matrix)을 구성하여야 한다. 예를 들어 분류기가 예측하고자 하는 대상이 610이고 이에 대한 분류 결과를 평가한다고 가정할 경우, 분류기가 문헌을 610으로 예측하였고 실제 범주 또한 610인 경우를 'TP'(True Positive)로 정의한다. 반대로 분류기가 610으로 예측하였으나 실제 범주가 610이 아닌 경우는 'FP'(False Positive)에 해당한다. 또한 분류기가 610이 아니라고 예측하였으나 실제 범주가 610인 경우는 'FN'(False Negative)에 해당하며, 예측 결과와 실제 범주가 모두 610이 아닌 경우는 'TN'(True Negative)로 분류한다. 이러한 혼동행렬을 기반으로 정확률(Precision), 재현율(Recall), 정확도 등의 성능 지표를 산출하였다(〈표 2〉 참조).

정확률은 분류기가 특정 범주로 예측한 문헌 중 실제로 해당 범주에 속하는 문헌의 비율을 의미하며, 재현율은 실제로 해당 범주에 속하는 문헌 중 분류기가 올바르게 해당 범주로 예측한 문헌의 비율을 의미한다. 정확도는 전체 문헌 중 분류기가 올바르게 분류한 문헌의 비율을 나타낸다. F_1-score는 정확률과 재현율의 조화평균을 통해 산출되는 지표로 두 지표를

〈표 2〉 지표 산출 방식

지표	산출 방식
정확률 (Precision)	$\frac{True\ Postive(TP)}{True\ Positive(TP)+False\ Positive(FP)}$
재현율 (Recall)	$\frac{True\ Postive(TP)}{True\ Positive(TP)+False\ Negative(FN)}$
정확도 (Accuracy)	$\frac{True\ Postive(TP)+True\ Negative(TN)}{True\ Positive(TP)+False\ Positive(TP)+False\ Negative(FN)+True\ Negative(TN)}$
F_1-score	$\frac{2*Recall*Precision}{Recall+Precision}$
F_1-score (macro)	$\frac{1}{N}\sum_{i=1}^{N} F_1\text{-}score_i$

www.kci.go.kr

종합적으로 고려하여 분류 성능을 평가하기 위해 사용된다.

한편, 개별 범주에 대한 성능뿐만 아니라 전반적인 분류 결과에 대한 평가 또한 필요하다. 정확도는 전체 문헌에 대한 예측 결과를 기준으로 산출되는 지표이지만, 범주 간 문헌 수 분포가 불균형한 경우 문헌 수가 많은 범주의 성능이 과도하게 반영되어 전체 성능이 왜곡될 수 있다는 한계가 있다. 이러한 한계를 보완하기 위해 본 연구에서는 각 범주에 대해 F_1-score를 산출한 후, 이를 평균하여 계산하는 F_1-score(macro)를 함께 산출함으로써 범주 간 불균형의 영향을 최소화하고 전반적인 분류 성능을 평가하고자 하였다.

4. 연구 결과

4.1 서지 레코드 분류 현황

분석 대상인 132,020건의 데이터셋에 대하여 DDC 분류기호 할당 현황을 살펴보았다(〈표 3〉 참조). 우선 강목의 분포를 살펴본 결과 650이 30.1%(39,683건)로 가장 높은 비중을 차지하고 있었고, 뒤이어 610가 22.4%(29,624건), 620이 18.4%(24,336건)이 높은 비중을 차지하고 있었다. 반대로 670과 680은 각각 0.6%(829건), 0.7%(891건)로 적은 비중을 가졌다.

요목 수준 분류기호 할당 현황을 살펴본 결과 658이 가장 많았으며, 그 다음으로 616, 650, 621이 높은 비중을 차지하고 있었다(〈표 4〉 참조). 반대로 레코드가 없거나 매우 적게 할당된 요목도 있었다. 예를 들어, 605, 656, 699의 경우에는 전체 데이터셋 중 1건만 존재하였다. 이에 대해서는 학습 문헌과 실험 문헌의 구분이 불가능하므로 분석에서 제외한 후, 132,017건을 대상으로 분석하였다.

4.2 자동 분류 실험 결과

각 분류모델(Omikuji, FastText, BERT)의 적용 결과를 제시함에 있어 분류자질 즉, 문헌의 제목만을 활용한 경우와 설명 정보를 추가

〈표 3〉 DDC 강목 수준의 분류기호 분포

기호	강목명	레코드 수(건)
600	Technology	2,440(1.8%)
610	Medicine and health	29,624(22.4%)
620	Engineering	24,336(18.4%)
630	Agriculture	11,783(8.9%)
640	Home and family management	16,745(12.7%)
650	Management and public relations	39,683(30.1%)
660	Chemical engineering	3,939(3.0%)
670	Manufacturing	829(0.6%)
680	Manufacture of products for specific uses	891(0.7%)
690	Construction of buildings	1,750(1.3%)
전체		132,020(100%)

www.kci.go.kr

〈표 4〉 DDC 요목 수준의 분류기호 분포(상위 10개)

기호	요목명	레코드 수(건)	비중(%)
658	General management	26,822	20.3
616	Disease	8,748	6.6
650	Management and auxiliary services	7,593	5.8
621	Applied physics	7,132	5.4
641	Food and drink	6,097	4.6
613	Personal health and safety	5,959	4.5
649	Child rearing; home care of people with disabilities and illnesses	5,545	4.2
628	Sanitary engineering	4,821	3.7
615	Pharmacology and therapeutics	4,542	3.4
657	Auxiliary services: Accounting	3,157	2.4
전체		80,416	60.9

한 경우를 구분하였다. 또한 강목, 요목 수준의 정확도와 F_1-score를 중심으로 성능을 비교하였다.

4.2.1 Omikuji

Omikuji 분류모델을 활용하여 강목 수준에서 분류기호에 대한 분류 결과를 살펴보았다. 먼저, 강목별 성능을 살펴본 결과 분류자질의 차이와 관계없이 학습 문헌의 양이 상대적으로 많은 610, 620 그리고 650의 분류 성능은 높게 나오는 한편, 그 수가 적은 670은 가장 낮은 성능을 보였다(〈표 5〉 참조).

강목별 성능은 설명 정보 추가 여부에 따라 차이가 있었다. 대표적으로 610과 640은 설명이 추가되면서 강목별 정확도가 17.5%, 25.5%로, F_1-score는 0.0952, 0.1667로 성능이 향상되었다. 반대로 630과 660은 정확도와 F_1-score가 하락하였다. 전반적으로 두 분류 기호 모두에서 분류기의 예측 빈도가 감소하고, 정확률도 하락한 것을 확인하였다. 특히 660의 정확률 감소 폭이 두드러졌는데, 전반적으로 620으로 잘못 분류되는 사례가 많은 것으로 나타났다. 두 지표가 각각 상승, 하락한 경우도 있었다. 650은 정확도가 낮아진 반면 F_1-score는 높아졌는데, 이는 분류모델에서 해당 범주로의 예측 빈도를 줄이면서도 실제로 해당 범주에 속하는 문헌을 비교적 정확하게 식별하기 시작한 것으로 해석할 수 있다. 이와 대조적으로 620은 정확도는 상승하고 F_1-score는 하락하였는데, 650과 달리 예측 빈도는 늘었으나 범주의 구분 능력은 다소 떨어진 것으로 분석된다.

강목 전체적 수준의 정확도는 제목에서 추출한 토큰만을 분류자질로 활용하여 분류모델을 구축한 경우 72.59%로 나타났으며, 설명 정보를 추가한 경우 78.26%로 향상되었다(〈표 5〉 참조). F_1-score(macro)의 경우, 제목에서 추출한 토큰만을 활용했을 때 0.5804로 나타났으며, 설명 정보를 추가한 경우 0.5989로 향상되었다(〈표 5〉 참조). 이는 강목별로는 설명 정보 추가에 따른 성능의 향상과 저하가 혼재할 수 있으나, 분류모델의 전반적인 예측 성능은 향상되었음을 의미한다.

www.kci.go.kr

〈표 5〉 Omikuji 분류모델 강목별 정확도 및 F_1-score

구분		600	610	620	630	640	650	660	670	680	690	전체
정확도(%)	T	46.35	64.21	83.26	67.39	41.23	92.57	48.59	10.17	26.98	42.86	72.59
	T+D	39.90	81.71	85.49	61.37	66.73	89.40	40.13	9.38	29.85	44.90	78.26
	변화	-6.45	17.5	2.23	-6.02	25.50	-3.17	-8.46	-0.79	2.87	2.04	5.67
F_1	T	0.5687	0.7070	0.8256	0.7339	0.5372	0.7591	0.5626	0.1690	0.3953	0.5455	0.5804
	T+D	0.5147	0.8022	0.8051	0.6803	0.7039	0.8554	0.4830	0.1622	0.4255	0.5570	0.5989
	변화	-0.0540	0.0952	-0.0205	-0.0536	0.1667	0.0963	-0.0796	-0.0068	0.0302	0.0115	0.0185

비고. T: 제목, T+D: 제목+설명

요목 수준에서 Omikuji 분류모델을 적용한 결과, 설명 정보를 추가하였을 때 전체적으로 정확도가 상승한 요목의 수는 30개, 하락한 요목의 수는 39개였으며. F_1-score의 경우 성능이 높아진 요목의 수는 27개, 낮아진 요목의 수는 43개인 것으로 나타났다. 성능이 향상된 요목들은 평균적으로 정확도 14.97%, F_1-score 0.1640이 상승했으나, 하락한 요목들의 정확도는 -13.11%. F_1-score는 -0.1474 감소했다(〈표 6〉 참조).

요목별 성능 변화를 살펴본 결과, 강목 640과 650에 속한 요목들은 상대적으로 성능이 향상되는 경향을 보인 반면, 620과 630 강목에 속한 요목들은 요목 수준에서 전반적인 분류 성능 저하를 보였다. 이들 요목은 동일 강목 내 다른 요목들로 예측되는 경우가 증가하였으며, 이는 강목 수준의 주제 구분은 유지된 반면, 요목 수준의 세분화된 예측이 상대적으로 어려워졌기 때문으로 보인다. 앞서 제시한 상위 분포 10개 요목(〈표 4〉 참조) 가운데 8개 분류기호에서는 정확도와 F_1-score가 모두 증가한 반면, 621은 두 지표 모두 하락하였다. 특히 621의 성능 저하는 620 주제 영역 전반에서 요목 간 분류 경계가 약화되었음을 보여주며, 해당 영역에 대한 분류기의 식별력이 상대적으로 감소하였음을 의미한다. 한편, 가장 높은 비율을 차지한 658의 경우 정확도는 감소하였으나 F_1-score는 증가하였는데, 이는 분류모델이 해당 범주로의 과도한 예측 빈도를 줄이는 동시에 실제로 해당 범주에 속하는 문헌을 이전보다 보다 정확하게 식별하기 시작했음을 의미한다.

빈도수가 적은 분류기호들은 예측 성능이 낮

〈표 6〉 Omikuji 분류모델 설명정보 추가 시 성능 변화

구분		600	610	620	630	640	650	660	670	680	690	전체
정확도(%)	상승	2 (25.36)	5 (7.09)	2 (3.85)	2 (1.98)	6 (25.46)	4 (23.20)	3 (4.24)	2 (13.47)	0 (-)	4 (8.62)	30 (14.97)
	하락	3 (-9.76)	4 (-3.39)	7 (-9.6)	6 (-10.35)	2 (-8.62)	2 (-2.69)	6 (-9.93)	3 (0.21)	3 (-14.25)	3 (-42.28)	39 (-13.11)
F_1	상승	2 (0.3008)	3 (0.3675)	1 (0.0281)	1 (0.0180)	6 (0.2229)	5 (0.1878)	3 (0.0422)	2 (0.2063)	0 (-)	4 (0.1060)	27 (0.1640)
	하락	3 (-0.1508)	6 (-0.0531)	8 (-0.1018)	8 (-0.0778)	2 (-0.1057)	1 (-0.0356)	6 (-0.1067)	3 (-0.2500)	3 (-0.1897)	3 (-0.3014)	43 (-0.1474)

www.kci.go.kr

거나, 예측이 되지 않는 경우도 발생하였다. 이러한 점을 포착하여, 빈도를 기반으로 평균적인 요목별 성능을 살펴보았을 때 빈도수 하위 25%(59건) 미만인 요목의 정확도는 설명 정보 추가 전후 각각 4.49%, 8.26%였고 F_1-score는 0.0652, 0.1로 확인되었다. 반대로 상위 25% 빈도수를 가진 요목들의 경우 정확도는 각각 53.86%, 58.41%였으며, F_1-score는 0.5878, 0.6032로 나타났다. 이는 학습문헌의 수 또한 성능 예측 시 중요한 요소 중 하나로 작용함을 의미한다.

Omikuji 분류모델의 성능을 종합해보면 요목 수준 보다는 강목 수준에서 더욱 뛰어난 분류 성능을 보였으며, 설명 정보 추가에 따른 일부 성능 향상을 확인할 수 있었다. 요목 수준에서 제목 추출 토큰만을 활용한 경우 전체 정확도 58.57%, F_1-score(macro) 0.3577로 나타났으며, 문헌의 설명 정보를 추가한 경우 전체 정확도가 62.65%로 상승하였으나, F_1-score(macro)는 0.3437로 감소하는 양상을 보였다. 이와 같은 결과는 분류자질로 문헌 설명 정보의 토큰까지 포함하면서 분류에 유의미한 정보 외에도 잡음이 증가하거나 주제 특정성이 낮은 토큰이 포함되어, 분류기호 간 균형에 부정적인 영향을 미친 것으로 해석할 수 있다.

4.2.2 FastText

FastText로 구축한 분류모델 또한 강목과 요목별로 적용하여 그 결과를 비교하였다. 우선 강목별 정확도와 F_1-score를 살펴보면 Omikuji 분류모델의 적용 결과와 유사한 결과를 보인다. 학습 문헌의 양이 많은 610, 620 그리고 650의 분류 성능은 뛰어난 반면, 그 수가 적은 670은 비교적 매우 낮은 성능을 보이고 있다(〈표 7〉 참조). 문헌의 설명 정보를 추가한 경우에는 대부분의 강목에서 성능이 향상된 것을 확인할 수 있다. 특히 비교적 낮은 성능을 보이던 강목(670, 680, 690) 또한 설명 정보를 추가했을 때 정확도가 약 3.98%~8.55% 증가하였으며, F_1-score는 약 0.02~0.06까지 증가한 것을 확인할 수 있다.

학습문헌의 수가 적어 가장 낮은 성능을 보이는 670에 대한 FastText 분류모델의 적용 결과를 살펴보면, 설명 정보 추가 전후로 각각 정확도 27.59%, 36.14%, F_1-score 0.3721, 0.4316을 기록하고 있다(〈표 7〉 참조). 같은 강목에 Omikuji 분류모델을 적용한 결과는 정확도 10.17%, 9.38%, F_1-score 0.1690, 0.1622인 것을 확인할 수 있다(〈표 5〉 참조). 즉, 학습문헌의 수가 적은 범주에 대한 분류기호를 예측했을

〈표 7〉 FastText 분류모델 강목별 정확도 및 F_1-score

구분		600	610	620	630	640	650	660	670	680	690	전체
정확도(%)	T	53.78	85.71	81.12	71.03	61.30	82.03	59.41	27.59	41.56	50.31	76.89
	T+D	56.56	86.23	84.39	72.50	72.96	88.18	62.18	36.14	47.19	54.29	81.30
	변화	2.78	0.52	3.27	1.47	11.66	6.15	2.77	8.55	5.63	3.98	4.41
F_1	T	0.6352	0.7424	0.8407	0.7604	0.6836	0.8133	0.6424	0.3721	0.5246	0.6090	0.6624
	T+D	0.6330	0.8104	0.8510	0.7676	0.7476	0.8775	0.6250	0.4316	0.5753	0.6333	0.6953
	변화	-0.0022	0.0680	0.0103	0.0072	0.0640	0.0642	-0.0174	0.0595	0.0507	0.0243	0.0329

비고. T: 제목, T+D: 제목+설명

www.kci.go.kr

때 FastText 분류모델의 성능이 Omikuji에 비해 좋은 것을 확인할 수 있는데, 이는 FastText가 단어 임베딩 기반의 분산 표현을 활용함으로써 학습 문헌이 상대적으로 부족한 범주에서도 의미 정보를 보다 효과적으로 반영할 수 있기 때문으로 분석된다.

전반적인 성능을 살펴보았을 때, 제목에서 추출한 토큰만을 분류자질로 활용한 경우 전체 정확도 76.89%, F_1-score(macro) 0.6624를 기록했으며 설명 정보를 추가한 경우 정확도는 81%, F_1-score(macro)는 0.69까지 상승하였다(〈표 7〉 참조). 이는 곧 설명 정보가 추가되면서 분류자질이 보완되어 분류모델의 전반적인 예측 성능이 향상되었음을 의미한다.

요목 수준의 성능 변화를 살펴본 결과 Omikuji 결과와 대비적으로 모든 강목 아래 요목들의 성능이 향상되는 경향을 보이는 것을 확인할 수 있었다. 특히 분류 성능이 향상된 분류기호보다 감소한 분류기호가 더 많았던 Omikuji 분류모델과 달리 지표가 향상된 요목들의 수가 크게 늘어났으며 전반적으로 성능 향상폭이 하락폭보다 높음을 볼 수 있다(〈표 6〉, 〈표 8〉 참조). 특히 강목 610과 640에 속한 요목들의 성능이 전반적으로 높아지는 경향이 있었다. 이는 FastText의 자질 표현방식이 상대적으로 학습 문헌이 부족한 범주에서도 의미 정보를 효과적으로 반영할 수 있기 때문으로 분석된다.

빈도를 기반으로 평균적인 요목별 성능을 보았을 때 빈도 수 하위 25%(59건) 미만인 요목의 정확도는 설명 정보 추가 전후로 각각 5.87%, 10.21%를 기록하였고, F_1-score는 0.08364, 0.1289로 나타났다. 반면, 빈도 수 상위 25% 요목들의 경우 정확도는 설명 정보 추가 전후로 각각 61.80%, 66.75%를 기록하였으며, F_1-score는 0.6507, 0.68256로 확인되었다. Omikuji 분류모델에 비해서는 높은 성능을 보이고 있지만 빈도 수 하위 25% 요목에 대한 성능 지표가 상위 25%의 요목 대비 현저히 낮은 것을 확인할 수 있으며, FastText 분류모델 또한 학습문헌의 수에 따라 분류 성능에 큰 영향을 받는다는 것을 확인할 수 있다.

FastText 분류모델 또한 요목 수준 보다는 강목 수준에서 더 나은 성능을 보였지만, 문헌의 설명 정보 추가에 따른 성능 증가는 요목 수준에서 더 큰 폭으로 나타났다. 제목에서 추출한 토큰만을 분류자질로 활용한 경우 전체 정확도는 약 63%, F_1-score(macro)는 약 0.4140인 반면, 문헌 설명 정보를 추가한 결과 전체 정확

〈표 8〉 FastText 분류모델 설명정보 추가 시 성능 변화

구분		600	610	620	630	640	650	660	670	680	690	전체
정확도(%)	상승	3 (11.74)	9 (5.99)	7 (6.58)	7 (6.70)	7 (11.84)	4 (9.84)	6 (14.38)	3 (18.40)	3 (23.96)	4 (31.89)	53 (14.54)
	하락	1 (-0.98)	0 (-)	2 (-1.50)	2 (-9.13)	1 (-15.79)	1 (-1.57)	2 (-2.87)	1 (-4.55)	0 (-)	1 (-9.62)	11 (-4.86)
F_1	상승	3 (0.1418)	8 (0.0200)	7 (0.4556)	6 (0.0740)	8 (0.0818)	5 (0.0740)	7 (0.1478)	3 (0.1792)	4 (0.2379)	5 (0.2426)	56 (0.1278)
	하락	2 (-0.0446)	1 (-0.0032)	2 (-0.0114)	4 (-0.0753)	1 (-0.1886)	1 (-0.1886)	2 (-0.0366)	2 (-0.0361)	0 (-)	2 (-0.0666)	17 (-0.0549)

www.kci.go.kr

도가 약 69%에 달했으며 F_1-score(macro)는 0.4678으로 증가했다.

4.2.3 BERT

BERT 기반으로 구축한 분류모델을 적용하여 강목별 정확도와 F_1-score를 살펴본 결과, 문헌의 설명 정보를 추가했을 때 1개 강목(660)을 제외한 모든 강목에서 성능이 향상된 것으로 나타났다(〈표 9〉 참조). 강목별 성능은 앞서 제시한 모델들과 마찬가지로 학습 문헌의 양에 영향을 받는 경향을 보였으나, 기존에 비교적 낮은 성능을 보이던 670과 680에서는 설명 정보를 추가함으로써 정확도가 각각 약 13.25%와 15.73% 증가하였고, F_1-score 또한 약 0.1165와 0.0646 상승하여 해당 강목들에 대한 식별력이 뚜렷하게 개선되었다. 이는 BERT가 문헌을 표현할 때 단어의 단순한 등장 여부뿐 아니라 문맥 정보를 반영하여 문헌 표현을 학습하는 특성을 지니고 있어, 설명 정보 추가를 통해 문헌 간 구분에 필요한 의미적 정보를 보다 효과적으로 활용할 수 있었기 때문으로 분석된다. 또한 강목별 적용 결과를 종합적으로 살펴보면, BERT 기반 분류모델은 Omikuji 및 FastText 분류모델에 비해 문헌 설명 정보 추가에 따른 성능 향상 폭이 상대적으로 크게 나타났으며, 이는 주제 식별력이 보다 효과적으로 개선되었음을 시사한다.

전반적인 모델 성능 또한 문헌의 제목만 활용한 경우에는 전체 정확도가 83.16%로 나타났으며, 문헌 설명 정보를 추가했을 때 90.14%로 증가하였다. 또한, F_1-score(macro)는 제목만 활용하였을 때 0.7190, 문헌 설명 정보를 추가했을 때 0.7829로 나타나며, 앞서 적용한 두 모델에 비해 높은 성능을 보였다(〈표 9〉 참조).

요목 수준의 성능 변화를 살펴본 결과 성능이 향상된 요목들의 수가 감소한 요목들의 수보다 많으며, 그 폭 또한 전반적으로 향상폭이 하락폭보다 큰 것을 확인할 수 있다(〈표 10〉 참조). 특히 강목 610과 640에 속한 요목들의 성능이 전반적으로 높아지는 경향을 보였다. 즉, BERT 또한 FastText와 같이 자질 표현방식인 임베딩 방식이 효과적임을 확인할 수 있었다. 빈도를 기반으로 평균적인 요목별 성능을 보았을 때, 세부적인 요목별 F_1-score를 살펴보면 전체 데이터셋에서 높은 비중을 차지하는 649, 659는 높은 F_1-score를 보이는 반면, 문헌의 수가 10개 미만인 분류기호들은 F_1-score "0"을 기록하는 등 낮은 점수를 보인다. 이는

〈표 9〉 BERT 분류모델 강목별 정확도 및 F_1-score

구분		600	610	620	630	640	650	660	670	680	690	전체
정확도(%)	T	71.72	81.30	89.03	80.22	72.60	91.56	67.77	28.92	48.31	57.71	83.16
	T+D	80.33	92.10	92.93	86.08	86.51	95.51	65.74	42.17	64.04	62.86	90.14
	변화	8.61	10.80	3.90	5.86	13.91	3.95	-2.03	13.25	15.73	5.15	6.98
F_1	T	0.7353	0.8399	0.8814	0.8393	0.7546	0.8611	0.6935	0.3871	0.5584	0.6392	0.7190
	T+D	0.7967	0.9198	0.9119	0.8791	0.8705	0.9501	0.6736	0.5036	0.6230	0.7006	0.7829
	변화	0.0614	0.0799	0.0305	0.0398	0.1159	0.089	-0.0199	0.1165	0.0646	0.0614	0.0639

비고. T: 제목, T+D: 제목+설명

www.kci.go.kr

〈표 10〉 BERT 기반 분류모델 요목별 정확도 및 F_1-score(문헌 설명 추가)

	600	601	602	603	604	605	606	607	608	609
정확도(%)	0.0	40.0	0.0	0.0	0.0	-	20.0	83.3	89.1	60.0
F_1-score	0.0000	0.5714	0.0000	0.0000	0.0000	-	0.3333	0.8213	0.8219	0.5455
	610	611	612	613	614	615	616	617	618	619
정확도(%)	72.7	67.4	63.8	74.5	75.6	69.4	77.4	78.8	69.5	0.0
F_1-score	0.7434	0.7209	0.6431	0.7375	0.7686	0.6893	0.7733	0.7958	0.7036	0.0000
	620	621	622	623	624	625	626	627	628	629
정확도(%)	67.1	86.4	65.4	68.0	83.5	76.7	-	91.7	89.2	75.5
F_1-score	0.6788	0.8444	0.6415	0.7338	0.8284	0.8018	-	0.8982	0.8652	0.7723
	630	631	632	633	634	635	636	637	638	639
정확도(%)	87.3	69.2	63.6	40.6	73.3	72.1	81.3	12.5	36.4	81.7
F_1-score	0.8685	0.7013	0.6667	0.4590	0.7446	0.6782	0.8320	0.2000	0.4444	0.8293
	640	641	642	643	644	645	646	647	648	649
정확도(%)	45.5	84.6	66.7	25.0	0.0	40.0	83.3	79.3	68.4	91.4
F_1-score	0.5172	0.8431	0.7407	0.4000	0.0000	0.5333	0.8283	0.7931	0.6190	0.8918
	650	651	652	653	654	655	656	657	658	659
정확도(%)	68.8	69.4	25.0	0.0	-	0.0	-	94.6	90.6	70.1
F_1-score	0.7185	0.6772	0.4000	0.0000	-	0.0000	-	0.9477	0.8877	0.7426
	660	661	662	663	664	665	666	667	668	669
정확도(%)	58.2	9.1	79.2	24.1	74.1	35.0	11.1	37.5	51.2	66.7
F_1-score	0.5860	0.1429	0.6441	0.2857	0.6667	0.4242	0.2000	0.5455	0.5385	0.6102
	670	671	672	673	674	675	676	677	678	679
정확도(%)	43.8	72.7	0.0	0.0	33.3	0.0	0.0	50.0	0.0	0.0
F_1-score	0.4912	0.6531	0.0000	0.0000	0.3636	0.0000	0.0000	0.6667	0.0000	0.0000
	680	681	682	683	684	685	686	687	688	689
정확도(%)	0.0	40.0	0.0	0.0	16.7	0.0	78.1	42.9	0.0	-
F_1-score	0.0000	0.4058	0.0000	0.0000	0.2222	0.0000	0.7353	0.4615	0.0000	-
	690	691	692	693	694	695	696	697	698	699
정확도(%)	61.5	50.0	60.0	44.4	75.0	0.0	77.8	50.0	0.0	-
F_1-score	0.5981	0.5333	0.6408	0.4706	0.7500	0.0000	0.7778	0.5294	0.0000	-

불균형한 학습 문헌의 양에서 나타나는 문제로 확인된다. 또한, 빈도 수가 하위 25% 미만인 요목들은 설명 정보 추가 전후로 각각 정확도 8.00%, 6.96%, F_1-score 0.0816, 0.08933를 기록하였다. 반대로 빈도 수 상위 25%인 요목들의 경우 정확도 71.26%, 79.49%, F_1-score 0.7218, 0.7914를 기록하였다. BERT 기반 분류모델 또한 다른 두 모델과 같이 학습문헌 수에 따라 분류 성능에 영향을 받는 것을 확인할 수 있다.

종합해보면 BERT 기반 분류모델 또한 강목 수준에서 더 나은 성능을 보인다. 요목 수준에 적용한 결과, 제목만 활용하였을 때 전체적인 정확도 71.85%, F_1-score(macro) 0.4558, 문헌 설명 정보 추가 후 전체적인 정확도 79.5%, F_1-score(macro) 0.4962를 기록하였으며 강목보다 낮은 수준임을 확인할 수 있다.

www.kci.go.kr

4.3 자동 분류모델 비교

3가지 분류모델을 강목에 적용한 결과를 살펴보면 BERT 기반 분류모델이 가장 좋은 성능을 보인다. 분류자질에 문헌 설명 정보를 포함하였을 때를 기준으로 BERT 기반 분류모델은 전체 정확도 90.14%, F_1-score(macro) 0.7829를 기록하며 뛰어난 성능을 보였으며, FastText 분류모델이 전체 정확도 81.30%, F_1-score(macro) 0.6953으로 뒤를 이었다. Omikuji 분류모델은 전체 정확도 78.26%, F_1-score(macro) 0.5989를 기록하며 다른 두 분류모델에 비해 낮은 성능을 보였다(〈표 11〉 참조).

모든 분류모델은 제목만을 활용한 분류자질에 문헌 설명 정보를 추가하였을 때 전체 정확도 및 F_1-score에서 더 나은 성능을 보였다. 전체 정확도는 제목만 활용한 경우에 비하여 평균적으로 5.7%가 증가하였고, F_1-scoree(macro)는 0.038이 증가한 것으로 나타났다. 이때 전체 정확도는 BERT(6.98%)-Omikuji(5.67%)-FastText(4.41%) 순서로 큰 증가폭을 보였으며, F_1-score는 BERT(0.0639)-FastText(0.0329)-Omikuji(0.0185) 순서로 증가폭이 큰 것을 확인할 수 있다.

모든 분류모델에서 가장 뛰어난 성능이 확인된 강목은 650이며, 문헌 설명 정보를 추가한 경우에 정확도와 F_1-score가 가장 높은 것을 확인할 수 있다. 하지만, 분류자질에 차이가 있는 경우 즉, 제목만을 활용하였을 때 다른 강목에서 더 뛰어난 성능을 보인 예외가 있는데, FastText 분류모델의 610(정확도 85.71%), 620(F_1-score 0.8407)과 BERT 기반 분류모델의 620(F_1-score 0.8814)이 그 사례이다.

요목에 적용한 경우에도 Omikuji 분류모델의 F_1-score(macro) 외에는 모든 분류모델의 지표에서 문헌 설명 정보를 추가한 분류자질을 활용했을 때 더 나은 성능이 확인되었다(〈표 12〉 참조). Omikuji 분류모델의 F_1-score(macro) 하락 원인은 설명 정보 추가로 새로운 토큰이 증가했음에도 상위 5,000개 단어만 남기는 필

〈표 11〉 분류모델별 성능 비교(강목)

강목	정확도(%)		F_1-score(macro)	
	제목	제목+문헌 설명	제목	제목+문헌 설명
Omikuji	72.59	78.26	0.5804	0.5989
FastText	76.89	81.30	0.6624	0.6953
BERT	83.16	90.14	0.7190	0.7829

〈표 12〉 분류모델별 성능 비교(요목)

요목	정확도(%)		F_1-score(macro)	
	제목	제목+문헌 설명	제목	제목+문헌 설명
Omikuji	58.57	62.65	0.3577	0.3437
Fasttext	63.74	68.98	0.4140	0.4678
BERT	71.85	79.52	0.4558	0.4962

www.kci.go.kr

터링 기준으로 인해 남아있는 토큰들의 구별력이 충분하지 않아 분류 변별력을 감소시켰기 때문으로 분석된다.

반면, 필터링을 적용하지 않은 FastText와 BERT 기반 분류모델의 성능이 향상된 것은 벡터화 과정에서 밀집행렬을 사용하는 구조적 특성에 기인한다. 이는 희소행렬을 활용하는 Omikuji 분류모델에 비해 문헌 설명 정보가 추가됨에 따라 얻는 성능적 이익이 크기 때문이다.

분류자질에 문헌 설명 정보를 추가하면서 나타난 성능 향상의 폭은 강목보다 요목 수준에서 더 컸다. 정확도는 제목만 활용한 경우에 비해 평균적으로 약 5.6% 가량 증가하였고, F_1-score(macro)는 약 0.027이 증가한 것으로 나타났다.

하지만, 분류 성능 자체만을 비교한다면 요목에 적용한 결과는 강목에 적용한 것에 비해 낮은 성능을 보인다(〈표 11〉, 〈표 12〉 참조). 이는 전반적으로 분류기가 예측해야 할 범주는 증가한 반면, 범주별 학습 문헌의 양은 감소한 것이 주 원인으로 분석된다. 또한, 범주 간 내용적 유사성이 높아짐에 따라 경계가 모호해진 것도 성능이 다소 낮게 측정된 원인으로 판단된다.

5. 논의 및 결론

본 연구는 정보자원 급증과 함께 자동분류의 중요성이 증대되는 가운데, 기존 ML 기반 연구의 한계를 보완하고 분류의 성능을 최적화하고자 시도하였다. 이를 위해 3개의 기계학습 알고리즘과 모델(Omikuji, FastText, BERT)을 기반으로 분류모델을 설계하였으며, 국회도서관으로부터 제공받은 DDC 기술과학(600) 서지레코드 데이터(270,729건)를 대상으로 자동분류를 실시하였다. 특히, 문헌 제목만으로는 분류모델의 학습에 있어 충분한 정보량이 확보되지 않는다는 지적(Peng et al., 2021)에 주목하여 문헌 제목으로 구성된 분류자질 이외에 문헌 설명 정보를 추가한 분류자질을 추가로 구성하였다.

이들 분류자질에 대한 자동분류를 강목과 요목 수준에서 개별적으로 수행하였으며, 설명 정보를 추가한 분류자질이 자동분류 성능에 미치는 영향을 확인하고자 각각의 분류결과를 정확도와 F_1-score를 통해 비교하였다. 자동분류 수행 결과, 분류모델의 기반이 되는 모델에 따라 그 성능의 차이가 발견되었다. BERT, FastText, Omikuji 순으로 높은 정확도와 F_1-score를 기록하였으며, 딥러닝 기반 사전 학습 언어모델인 BERT를 통해 임베딩을 수행하여 표현된 분류자질을 학습한 분류모델이 사전 학습을 하지 않은 두 모델보다 우수한 성능을 보였다.

분류자질의 문헌 설명 정보 추가 여부에 따른 차이 또한 확인할 수 있었다. 강목 수준에서 전체 정확도는 제목만 활용한 경우에 비해 평균 5.7%, F_1-score(macro)는 0.038이 증가하였으며, 요목 수준에서도 전체 정확도는 평균 5.7%, F_1-score(macro)는 0.027이 증가하였다. 이는 문헌과 관련된 텍스트를 추가하였을 때 자동분류 성능이 개선될 수 있음을 보여준다. 다만 예외적으로 Omikuji 분류모델을 요목 수준에서 적용하였을 때 F_1-score가 하락한 것이 확인되었는데, 이는 설명 정보를 추가함으로써 추출된 전체 토큰 수는 증가하였으나, 분류 자질로 활용되는 토큰 수는 상위 빈도 5,000개로 동일하게 고정하였기 때문으로 추측된다. 이 과정

www.kci.go.kr

에서 제거되는 토큰의 수가 증가하여, 분류를 구분할 수 있는 의미 있는 토큰이 상대적으로 더 많이 제거될 가능성이 존재한다. 그 결과 분류모델의 학습 과정에서 활용 가능한 정보가 감소하여 성능 하락이 나타난 것으로 추측된다.

강목별 성능에서는 학습문헌 수가 많은 특정 강목(610, 620, 650)에 대한 성능이 비교적 뛰어난 반면, 그 수가 적은 강목(670)에 대한 성능은 떨어지는 것을 확인하였다. 요목별 성능 또한 문헌 수가 하위 25%에 해당하는 경우 설명 정보를 추가하여도 8.26%의 정확도와, 0.1의 F_1-score 상승을 기록한 반면, 상위 25%에 해당하는 경우에는 58.41%의 정확도와 0.6032의 F_1-score 상승을 확인할 수 있었다. 이를 통하여 학습문헌의 수 또한 분류 성능에 대한 중요한 요소로 작용함을 파악하였다.

강목과 요목 수준의 분류 성능을 비교한 결과, 모든 분류모델에서 강목 수준이 요목 수준보다 전반적으로 우수한 성능을 보이는 것으로 확인되었다. 이는 요목 수준으로 갈수록 분류모델이 예측해야 할 주제 범주의 수가 증가하는 반면, 주제가 세분화되면서 범주별 문헌 수는 감소하고 내용적 유사성은 높아져 분류 경계가 상대적으로 모호해지기 때문인 것으로 보인다. 한편, 기존 선행연구에서는 DDC 기술과학(600) 분야의 요목 수준 분류 정확도가 70.90%로 보고된 바 있으나(Golub et al., 2024), 본 연구에서는 문헌 설명 정보를 추가적으로 활용한 결과 요목 수준에서 최대 79.52%(BERT)의 정확도를 기록하여 약 8.62%p의 성능 향상이 나타났다. 이는 제안한 접근이 요목 수준 분류 성능 개선에 효과적임을 시사한다.

여러 모델을 결합하는 앙상블(Ensemble) 기법으로 개발된 Annif와 같은 분류모델은 일반적으로 단일 분류모델에 비해 좋은 성능을 보인다고 알려져 있다(Golub et al., 2024; Suominen et al., 2022). 문헌 데이터의 주제 범위와 비교 기준에 차이가 있어 직접적인 비교는 제한적이지만, 문헌 유형에 따라 Annif의 F_1-score가 0.3~0.5 수준임을 고려할 때, 본 연구에서 활용한 Omikuji, FastText, BERT 기반 분류모델은 문헌 설명 정보를 추가한 경우 각각 0.3437, 0.4678, 0.4962의 F_1-score를 기록하여 Annif와 대등한 수준의 성능을 보였다(Suominen et al., 2022). 이는 적절한 모델 선정과 함께 분류자질을 보완할 경우, 단일 모델 기반 접근에서도 충분히 경쟁력 있는 자동분류 성능을 달성할 수 있음을 시사한다. 나아가 요목 수준에서 66.82%의 정확도를 기록한 Annif가 핀란드의 대학 및 도서 유통사 등 실제 업무 환경에 도입되고 있다는 점을 고려할 때(Golub et al., 2024), 본 연구에서 제안한 바와 같이 문헌 설명 정보를 활용한 BERT 기반 분류 접근 역시 실무적 활용 가능성이 있는 것으로 판단된다.

이 연구는 DDC의 전 주제 분야가 아닌 기술과학(600)이라는 특정 주제 분야의 서지 데이터만을 대상으로 실시하였다는 한계가 있다. 따라서 주제 분야를 DDC 전 분야 또는 국내도서관에서 주로 사용하는 KDC로 확대하고, 다양한 자료 유형에 대한 자동분류를 시도하여 그 범용성을 검증할 필요가 있다. 또한, 후속 연구에서 추가적인 분류모델을 적용해보고, 문헌 제목과 설명 등 텍스트 중심의 분류 자질에 도서 표지와 같은 이미지를 포함하는 방식으로 분류자질을 추가하여 분류 성능의 변화를 살펴볼 필요가 있을 것이다.

www.kci.go.kr

참 고 문 헌

곽철완 (2014). 도서관 분류법에 관한 국내 연구 동향: 최근 10년간의 연구를 중심으로. 한국비블리아학회지, 25(1), 173-191. http://doi.org/10.14699/kbiblia.2014.25.1.173

김영범, 장우권 (2023). 메타데이터를 활용한 기록물 자동분류 성능 요소 비교. 정보관리학회지, 40(3), 99-118. https://doi.org/10.3743/KOSIM.2023.40.3.099

김인후, 김성희 (2022). 딥러닝 기반의 BERT 모델을 활용한 학술 문헌 자동분류. 정보관리학회지, 39(3), 293-310. https://doi.org/10.3743/KOSIM.2022.39.3.293

김정현 (2011). 한국의 자료조직 분야 연구동향 분석: 1970~2010. 한국도서관·정보학회지, 42(3), 149-164. https://doi.org/10.16981/kliss.42.3.201109.149

김판준 (2022). 자질선정을 통한 국내 학술지 논문의 자동분류에 관한 연구. 정보관리학회지, 39(1), 69-90. https://doi.org/10.3743/KOSIM.2022.39.1.069

노지현, 이미화, 이은주 (2023). 목록이론의 이해와 적용. 서울: 한국도서관협회.

문화체육관광부 [발행년불명]. 도서관 정보나루. 출처: https://www.data4library.kr/

이용구 (2020). 목차 정보와 kNN 분류기를 이용한 사회과학 분야 도서 자동 분류에 관한 연구. 정보관리학회지, 37(1), 1-21. https://doi.org/10.3743/KOSIM.2020.37.1.001

이용구 (2023a). 단행본 서명의 단어 임베딩에 따른 자동분류의 성능 비교. 정보관리학회지, 40(4), 307-327. https://doi.org/10.3743/KOSIM.2023.40.4.307

이용구 (2023b). BERT 모형을 이용한 주제명 자동 분류 연구. 한국문헌정보학회지, 57(2), 435-452. https://doi.org/10.4275/KSLIS.2023.57.2.435

정영미 (2012). 정보검색연구. 연세대학교 출판문화원: 서울.

한국문화정보원 [발행년불명]. 문화 빅데이터 플랫폼. 출처: https://www.bigdata-culture.kr/

Bodenhamer, J. (2023). The reliability and usability of ChatGPT for library metadata. Available: https://hdl.handle.net/20.500.14446/339626

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2017). BERT: pre-training of deep bidirectional transformers for language understanding, arXiv preprint. https://doi.org/10.48550/arXiv.1810.04805

Formanek, M. (2024). Exploring the potential of large language models and generative artificial intelligence(GPT): Applications in Library and Information Science. Journal of Librarianship and Information Science, 1-23. https://doi.org/10.1177/09610006241241066

Golub, K. (2021). Automated subject indexing: An overview. Cataloging & Classification Quarterly, 59(8), 702-719. https://doi.org/10.1080/01639374.2021.2012311

www.kci.go.kr

Golub, K., Suominen, O., Mohammed, A. T., Aagaard, H., & Osterman, O. (2024). Automated dewey decimal classification of swedish library metadata using Annif software. Journal of Documentation, 80(5), 1057-1079. https://doi.org/10.1108/JD-01-2022-0026

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. arXiv preprint. https://doi.org/10.48550/arXiv.1607.01759

Noruzi, A. (2024). The use of artificial intelligence in knowledge organization and subject indexing. Informology, 3(1), 1-8.

Peng, Z., Abdollahi, B., Xie, M., & Fang, Y. (2021). Multi-label classification of short texts with label correlated recurrent neural networks. Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR '21), 119-122. https://doi.org/10.1145/3471158.3472246

Prabhu, Y., Kag, A., Harsola, S., Agrawal, R., & Varma, M. (2018). Parabel: partitioned label trees for extreme classification with application to dynamic search advertising. Proceedings of the 2018 World Wide Web Conference, 993-1002. https://doi.org/10.1145/3178876.3185998

Sebastiani, F. (2002). Machine learning in automated text categorization. ACM Computing Surveys (CSUR), 34(1), 1-47. https://doi.org/10.1145/505282.505283

Suominen, O., Inkinen, J., & Lehtinen, M. (2022). Annif and finto AI: developing and implementing automated subject indexing. JLIS.It, 13(1). 265-282. https://doi.org/10.4403/jlis.it-12740

• 국문 참고자료의 영어 표기
(English translation / romanization of references originally written in Korean)

Chung, Young-Mee (2012). Research in Information Retrieval. Seoul: Yonsei University Press.

Gim, Young Bum & Chang, Woo Kwon (2023). Comparison of performance factors for automatic classification of records utilizing metadata. Journal of the Korean Society for Information Management, 40(3), 99-118. https://doi.org/10.3743/KOSIM.2023.40.3.099

Kim, In hu & Kim, Seong Hee (2022). Automatic classification of academic articles using BERT model based on deep learning. Journal of the Korean Society for Information Management, 39(3), 293-310. https://doi.org/10.3743/KOSIM.2022.39.3.293

Kim, Jeong-Hyen (2011). An analytical study on research trends of resource organization in Korea: 1970~2010. Journal of Korean Library and Information Science Society, 42(3), 149-164. https://doi.org/10.16981/kliss.42.3.201109.149

www.kci.go.kr

Kim, Pan Jun (2022). An experimental study on the automatic classification of Korean journal articles through feature selection. Journal of the Korean Society for Information Management, 39(1), 69-90. https://doi.org/10.3743/KOSIM.2022.39.1.069

Korea Culture Information Service Agency [n.d.]. BigData MarketC. Available: https://www.bigdata-culture.kr/

Kwak, Chul Wan (2014). Trends in the current library classification research in Korea: a review of the literature in the past 10 years. Journal of the Korean Biblia Society for Library and Information Science, 25(1), 173-191. http://doi.org/10.14699/kbiblia.2014.25.1.173

Lee, Yong-Gu (2020). A study on book categorization in social sciences using kNN classifiers and table of contents text. Journal of the Korean Society for Information Management, 37(1), 1-21. https://doi.org/10.3743/KOSIM.2020.37.1.001

Lee, Yong-Gu (2023a). Performance comparison of automatic classification using word embeddings of book titles. Journal of the Korean Society for Information Management, 40(4), 307-327. https://doi.org/10.3743/KOSIM.2023.40.4.307

Lee, Yong-Gu (2023b). A study on automatic classification of subject headings using BERT model. Journal of the Korean Society for Library and Information Science, 57(2), 435-452. https://doi.org/10.4275/KSLIS.2023.57.2.435

Ministry of Culture, Sports and Tourism [n.d.]. Library Bigdata. Available: https://www.data4library.kr/

Rho, Jee-Hyun, Lee, Mihwa, & Lee, Eun-Ju (2023). Cataloging Theory and Practice. Seoul: Korean Library Association.

www.kci.go.kr