석사학위 논문

클래스 불균형을 극복한 향상된 BERT 기반 특허의 극한 다중 레이블 분류

Enhanced BERT-based Extreme Multi-label Classification of Patents Overcome Class Imbalance

2020년 12월

숭실대학교 대학원

소프트웨어학과

정 구 익

석사학위 논문

클래스 불균형을 극복한 향상된 BERT 기반 특허의 극한 다중 레이블 분류

Enhanced BERT-based Extreme Multi-label Classification of Patents Overcome Class Imbalance

2020년 12월

숭실대학교 대학원

소프트웨어학과

정 구 익

석사학위 논문

클래스 불균형을 극복한 향상된 BERT 기반 특허의 극한 다중 레이블 분류

지도교수 이 상 준

이 논문을 석사학위 논문으로 제출함

2020년 12월

숭실대학교 대학원

소프트웨어학과

정 구 익

정 구 익의 석사학위 논문을 인준함

심 사 위 원 장 최 종 선 인

심 사 위 원 신 정 훈 인

심 사 위 원 이 상 준 인

2020년 12월

숭실대학교 대학원

목 차

표 목 차

그 림 목 차

국문초록

클래스 불균형을 극복한 향상된 BERT 기반 특허의 극한 다중 레이블 분류

정구익

소프트웨어학과

숭실대학교 대학원

지식 정보 사회에 접어들면서 지식재산권의 대표적인 형태인 특허의 중요성을 날로 증가하고 있으며 매년 출원되는 특허의 수는 또한 빠르게 증가하고 있다. 이처럼 빠르게 증가하는 특허 데이터를 효율적으로 처리하고 정보 사용자가 효과적으로 데이터를 이용할 수 있도록 하는 특허 분류는 필수적인 업무이다. 현재 대부분의 특허 분류 작업은 수작업으로 수행되고 있으며 빠르게 증가하는 특허 데이터를 수동으로 분류하는 것은 매우 비효율적이다. 이러한 비효율성을 해결하기 위해 특허의 자동 분류를 위한 다양한 딥러닝 기반의 방법들이 제안되었다. 최근 대용량 말뭉치를 이용하여 사전 학습한 언어 모델인 BERT를 Fine-Tuning 하여 기존의 방법들보다 우수한 분류 성능을 보였다. 하지만 이 또한 특허

데이터의 특성을 고려하지 않으며 만족할만한 성능을 보이지 못하였다. 이로 인해 여전히 특허 분류는 대부분 수작업으로 수행되고 있다.

특허 데이터는 수백 수천 개의 클래스로 분류되어야 한다. 또한, 하나의 문서가 여러 클래스를 가질 수 있는 다중 레이블 분류 문제이며 클래스에 따른 데이터 분포가 매우 불균형하다. 본 연구에서는 위와 같은 특허 데이터의 특성을 고려한 분류 모델 생성을 위해 BERT 기반의 향상된 극한 다중 레이블 분류 모델을 제안하였다. 제안 모델에서는 대부분의 자연어처리 분야에서 우수한 성능을 보이는 BERT를 워드 임베딩을 위한 인코더로 활용하였다. 또한, 기존의 BERT 기반 분류 모델에서 모든 층을 거치며 전체 문장을 결합한 의미를 갖는 [CLS] 토큰만을 사용하는 것과는 달리 제안 모델에서는 BERT의 모든 final hidden state를 분류에 활용하였다. 이후 다중 레이블 특성을 고려한 문서 표현과 레이블 상관관계를 반영한 문서 표현을 얻기 위해 Multi-label Attention Mechanism과 Interaction Attention Mechanism 기반의 레이어를 BERT 위에 추가하였다. 본 연구에[서는 이렇게 획득한 두 문서 표현의 장점을 보존하며 최종 레이블 예측하기 위해 새로운 Fusion Strategy를 제안하였다. 추가적으로 특허 문서의 클래스 불균형 문제를 해결하기 위해 이진 교차 엔트로피 손실 함수를 대신하여 Normalized Focal Loss를 이용하여 모델을 학습하였다.

이에 따라 본 연구에서는 제안 모델의 우수성을 보이기 위해 미국, 유럽, 일본에서 출원된 약 백만 개의 영문 특허를 이용한 실험을 진행하였다. 먼저 제안 모델의 Fusion Strategy의 우수성을 보이기 위한 비교 실험과 손실함수에 따른 성능 변화를 확인하기 위한 실험을 진행하였다. 또한, 특허 분류와 극한 다중 레이블 분류에서 우수한 성능을 보인 베이스라인 모델들과의 비교 실험을 진행하였다. 실험 결과 제안하는 Fusion Strategy를 사용하였을 때 가장 높은 분류 정확도를 보였으며,

Normalized Focal Loss를 통해 모델을 학습시켰을 때 클래스 불균형을 극복할 뿐만 아니라 전체적인 성능이 향상됨을 보였다. 또한, 베이스라인 모델들과의 비교 실험에서 BERT Fine-Tuning 모델 대비 $P@1$에서 1.61%의 성능 향상을 보였으며 $P@3$, $P@5$ 각각에서 1.627%, 1.135%의 큰 성능 향상을 보였다. 본 연구를 통해 제안 모델이 기존의 특허 분류와 극한 다중 레이블 분류에서 우수한 성능을 보인 모델들과 달리 특허 데이터의 특성을 고려하고 문맥 정보를 반영하였기 때문에 특허 분류에 우수한 성능을 보인다는 것을 확인하였다.

ABSTRACT

Enhanced BERT-based Extreme Multi-label Classification of Patents Overcome Class Imbalance

GuIk Jung

Department of Software

Graduate School of Soongsil University

As the knowledge information society enters, the importance of patents, which are representative forms of intellectual property rights, is increasing day by day, and the number of patents filed every year is also increasing rapidly. Patent classification is essential to efficiently process rapidly increasing patent data and enable information users to use data effectively. Currently, most patent classification is performed manually, and it is very inefficient to manually classify rapidly increasing patent data. To solve this inefficiency, various deep learning based methods for the automatic classification of patents have been proposed. Recently, fine-tuning BERT, a pre-learned language model using a large corpus, showed better classification performance than existing methods. However, this

also did not consider the characteristics of the patent data and did not show satisfactory performance. For this reason, most of the patent classification is still carried out manually. Patent data should be classified into hundreds or thousands of classes. Also, it is a multi-label classification problem in which one document can have several classes, and the data distribution according to the classes is very unbalanced. In this study, a BERT based enhanced extreme multi-label classification model was proposed to generate a classification model that considers the characteristics of the above patent data. In the proposed model, BERT, which has an excellent performance in most natural language processing fields, was used as an encoder for word embedding. In addition, unlike existing BERT based classification models that use only [CLS] tokens with combined meanings of the entire sentence, the proposed model used all final hidden states of BERT for classification. After that, layers based on Multi-label Attention Mechanism and Interaction Attention Mechanism were added on the BERT to obtain the document representation considering the multi-label characteristics and the document expression reflecting the label correlation. In this study, a new Fusion Strategy was proposed to predict the final label while preserving the merits of the two document expressions thus obtained. Also, to solve the class imbalance problem in the patent document, the model was trained using Normalized Focal Loss instead of the binary cross entropy loss function. Accordingly, in this study, several experiments were conducted using about 1 million English patents filed in the US,

Europe, and Japan to show the superiority of the proposed model. First, a comparative experiment to show the superiority of the fusion strategy of the proposed model and an experiment to confirm the performance change according to the loss function were conducted. Also, a comparative experiment was conducted with baseline models that showed higher performance in patent classification and extreme multi-label text classification. The results of the experiment showed that the proposed Fusion Strategy showed the highest classification accuracy, and the Normalized Focal Loss showed that the model was not only overcome the class imbalance but also improved overall performance. In addition, in comparison with the baseline models, A showed 1.61% improvement in performance $P@1$ compared to BERT Fine-Tuning models, and a significant 1.627% and 1.135% improvement in $P@3$ and $P@5$ respectively. Through this study, it was confirmed that the proposed model shows excellent performance in patent classification because it considers the characteristics of patent data and reflects context information, unlike models that have shown excellent performance in conventional patent classification and extreme multi-label classification.

제 1 장 서 론

1.1 연구 배경

특허는 지식재산권의 대표적인 형태로 지식과 정보를 관리하기 위한 전략적 자원으로 여겨지며 그 중요성은 날로 증가하고 있다. 특허 문서는 최신의 기술과 지식을 내포하기 때문에 관련 특허의 조사 및 분석을 통해 연구에 소모되는 시간과 비용을 줄일 수 있어 신제품 개발과 같은 다양한 연구 개발 활동에 활용된다 [1]. 하지만 전문적인 내용을 다루며 수년간 축적된 방대한 특허 데이터 속에서 필요로 하는 특허데이터를 선별, 분석 및 처리하는 과정에서 많은 어려움을 겪는다. 이러한 어려움을 해결하기 위해 특허를 효율적으로 처리하고 이를 효과적으로 이용할 수 있도록 하는 많은 대안이 등장하였다 [2]. 특허 분류는 이러한 대안 중 하나로 기술 분야에 따라 특허를 구분하는 것을 말한다.

매년 출원되는 특허의 수는 모든 분야에 걸쳐 빠르게 증가하고 있다. World Intellectual Property Organization(WIPO)의 통계에 의하면 2019년 가장 많은 특허를 출원한 중국의 경우 출원된 특허 수가 그 전해에 보다 11.6% 증가하였다. 또한, 특허 출원 상위 10개국 중 8개의 국가의 특허 출원 수가 전년 대비 증가하였다. [표 1-1]은 세계 특허 출원 상위 8개국의 특허 출원 통계를 나타낸다 [3]. 이처럼 빠르게 생성되는 특허 문서를 사람에 의해 일일이 수동으로 분류하는 것은 매우 비효율적이다.

현재 대부분의 특허 분류 작업은 특허 전문가에 의해 수동으로 이루어지고 있으며, 빠르게 생성되는 특허 문서를 사람이 일일이 분석하고 이를 분류하는 것은 매우 비효율적이다 [4].

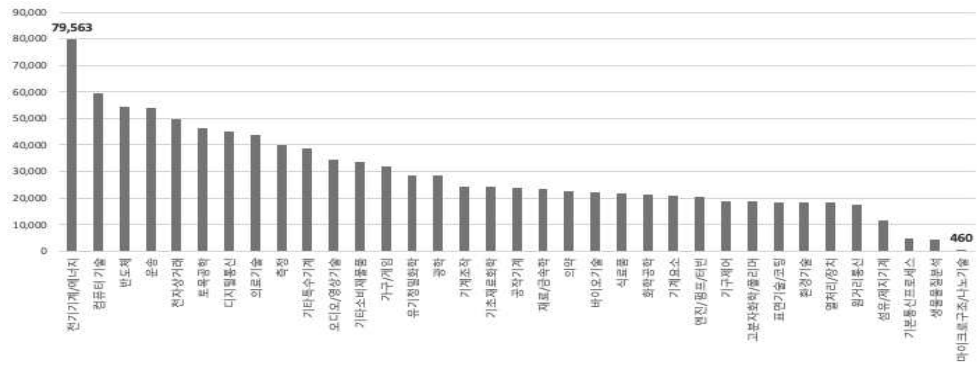
[표 1-1] 세계 특허 출원 상위 8 개국의 특허 출원 통계 [3]

출원국	중국	미국	일본	독일	한국	프랑스	영국	스위스
2017	48,900	56,680	48,206	18,948	15,752	8,013	5,569	4,485
2018	53,349	56,252	49,706	19,742	16,917	7,918	5,634	4,576
2019	58,990	57,840	52,660	19,353	19,085	7,934	5,786	4,610
증가율	20.6%	2.0%	9.2%	2.1%	21.2%	-1.0%	3.9%	2.8%

위와 같은 비효율성을 해결하기 위해 인공지능을 기반으로 특허를 자동으로 분류하기 위한 연구가 활발히 진행되고 있다 [5,6]. 대표적으로 머신러닝을 이용한 방법들과 Recurrent Neural Networks(RNN), Bidirectional Encoder Representations from Transformers(BERT)와 같은 딥러닝 기법을 이용한 방법들이 제안되었다. 하지만 이러한 방법들은 만족할만한 성능을 보이지 못하였으며 특허 분류는 여전히 수동으로 진행되는 경우가 많다.

1.2 연구 목적

인공지능 기반의 특허 분류는 다음과 같은 어려움을 갖는다. 첫 번째, 특허 문서는 다양한 분야에 해당하는 문서들의 집합이다. 따라서 특허 분류를 위한 클래스는 모든 기술 및 산업 분야를 포함하며 이를 세분화해야 하므로 그 수가 매우 많다 [7]. 두 번째, 특허 데이터는 문서 하나에 하나 이상의 클래스에 포함될 수 있는 다중 레이블 데이터이다. 또한, 하나의 문서에 포함될 수 있는 클래스의 수가 정해져 있지 않다. 이는 확장성 문제를 일으키며 분류 작업을 계산적으로 어렵게 만든다. 세 번째, 특허 데이터는 클래스 불균형한 데이터이며 Tail label을 갖는다. [그림 1-1]은 2015년도부터 2019년도까지 한국의 WIPO 기술 분류 기준 기술별 출원 특허 수를 나타낸다. 전기기계 및 에너지 관련 특허는 5년간 79,563개가 출원된 것에 반해 마이크로 구조 및 나노기술 관련 출원 특허는 460개에 불과하다 [8]. 이처럼 클래스에 따른 데이터 분포가 불균형한 것을 클래스 불균형이라고 부르며 데이터가 적어 학습에 어려움이 있는 클래스를 Tail label이라 부른다. 이러한 클래스 불균형과 Tail label은 딥러닝 기반의 분류 문제에서 발생하는 고질적인 문제로 여러 문제의 원인이 된다 [9].



[그림 1-1] 2015~2019 년도 한국의 기술 분야별 출원 특허 수 [8]

본 논문은 다음 두 가지를 목표로 한다. 첫 번째, 기존의 딥러닝 기반의 다중 레이블 텍스트 분류 모델보다 높은 분류 정확도의 극한 다중 레이블 특허 분류 모델을 제시한다. 두 번째, 손실 함수를 통해 클래스 불균형으로 인해 발생하는 문제를 극복한다. 우리가 아는 한 본 연구는 특허 분류에 있어 클래스 불균형 문제를 고려하고 이를 극복한 첫 번째 연구이다.

높은 성능의 분류 모델을 위해 BERT를 활용하여 문장을 표현하고, 이를 통해 토큰에 대한 문맥적 표현(Contextual Representation)이 가능하도록 하였다. 이때 본 연구에서는 BERT의 첫 번째 토큰인 [CLS] 만으로는 전체 시퀀스 정보를 충분히 표현하는 데 부족함이 있다고 생각하여 전체 시퀀스 벡터를 이용하였다. 이후, 전체 시퀀스에서 각 레이블에 대한 단어의 가중치를 찾기 위한 층, 단어와 레이블 사이의 의미적 연관성을 결정하는 층과 두 층의 출력을 적절히 조합하여 최종 레이블들을 분류하기 위한 층을 추가하여 분류 성능을 높인다. 또한, 다중 레이블 분류를 위해 일반적으로 사용되는 이진 교차 엔트로피 손실 함수를 대신하여 클래스 불균형을 극복할 수 있는 Normalized Focal Loss를 통해 학습을 진행하여 클래스 불균형으로 인해 발생하는 문제를 다소 극복하였다.

본 연구에서는 제안하는 모델의 우수성을 검증하기 위해 기존 특허 IPC 분류에서 우수한 성능을 보인 BERT Fine-Tuning 모델과 극한 다중 레이블 분류에서 SOTA를 달성한 모델 두 가지를 베이스라인 모델로 채택하여 성능을 비교분석 하였다 [10]. 추가로 클래스 불균형을 극복하였음을 보이기 위해 손실함수에 따른 클래스별 분류 정확도를 비교 분석하였다.

1.3 논문 구성

본 논문의 구성은 다음과 같다. 1장에서는 논문의 연구 배경과 목적에 대해 설명하고, 논문 구성을 소개한다. 2장은 관련 연구로 본 연구와 관련된 텍스트 분류와 기존의 특허 분류 연구를 소개하고, 제안하는 모델의 우수성을 보이기 위해 선정한 베이스라인 모델들에 관하여 설명한다. 3장 제안 방법에서는 본 논문에서 제안하는 분류 모델과 모델을 구성하는 각 층에 대해 설명하고 클래스 불균형을 극복하기 위해 사용한 손실 함수와 모델 개발 절차에 대해 설명한다. 4장에서는 본 연구의 실험에 쓰이는 특허 데이터와 평가 지표를 설명하고, 베이스라인 모델과 제안 모델의 성능을 비교 분석하고 손실 함수 변경에 따른 실험 결과를 해석한다. 마지막 5장에서 본 연구의 결론과 향후 연구 방향에 대해 고찰해 본다.

제 2 장 관련 연구 및 배경

제2장 관련 연구 및 배경에서는 첫 번째로, 본 연구와 관련된 텍스트 분류에 대해 설명하며 두 번째로, 딥러닝 기반으로 특허를 자동 분류하기 위한 기존 연구들에 대해 설명한다. 세 번째와 네 번째는 제안하는 모델의 우수성을 보이기 위해 선정한 베이스라인 모델들에 대한 설명으로 각각 입력 텍스트에 대해 수백, 수천 개의 후보 클래스 중 관련성 있는 클래스들을 찾는 Extreme Multi-label Text Classification(XMTC) 모델과 BERT 모델에 대해 설명한다.

2.1 텍스트 분류

텍스트 분류(Text Classification)는 입력받은 텍스트에 대해 후보 클래스 중 가장 관련성 있는 클래스로 구분하는 작업을 의미하며, 자연어처리의 대표적인 주제 중 하나이다 [11]. 인공지능 기술이 빠르게 발전하면서 텍스트 분류에 대한 연구가 활발히 진행되고 있으며, 그중 딥러닝 기반의 방법이 가장 주목을 받고 있다. 딥러닝 기반의 텍스트 분류는 일반적으로 크게 3단계로 구성된다. 첫 번째는, 텍스트 데이터 셋을 정제하여 노이즈로 작용할만한 단어들을 제거하기 위한 전처리 과정이다. 두 번째는, 인간의 언어를 컴퓨터가 이해할 수 있는 적절한 형태인 벡터로 변환하기 위한 Word Representation 과정이다. Word Representation은 단어를 학습하는 방식과 표현을 위한 차원에 따라 다양하며 자연어 처리에 있어 많은 성능 차이를 가져온다 [12]. 대표적인 단어 학습 방법으로는 주변 단어를 통해 중심 단어를 예측하는 방식의 Continuous Bag-of-Words(CBoW), 중심 단어를 통해 주변 단어를 예측하는 방식의

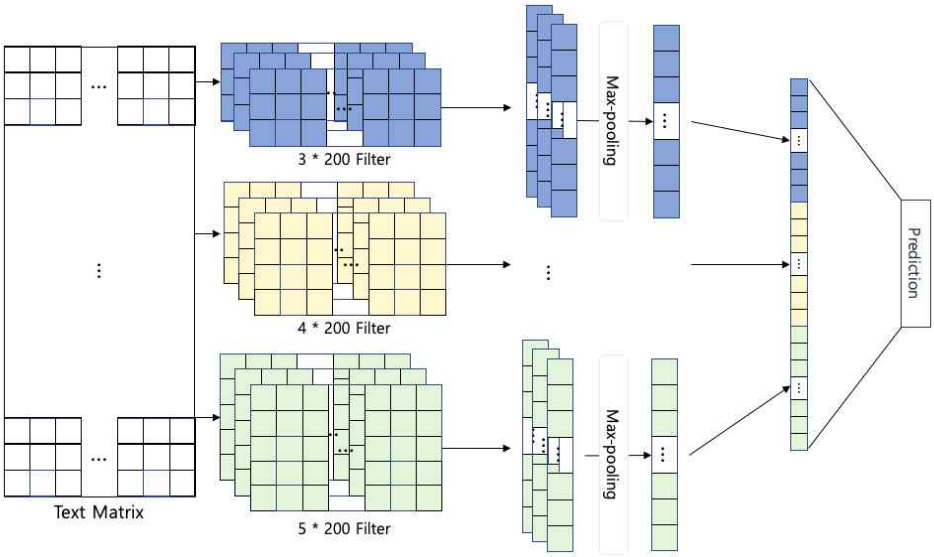
Continuous Skip-gram과 단어의 동시 등장 확률을 고려한 Global Vectors for Word Representation(GloVe)이 있다 [13,14]. 마지막은 텍스트 분류를 위한 단계 중 가장 중요한 분류 단계로 사용하는 분류 모델에 따라 많은 성능 차이를 보이며 이에 대한 지속적인 연구가 진행되고 있다.

Xiang et al.[15]은 Convolutional Neural Network(CNN) 기반 문장 분류 모델로 Word2Vec 워드 임베딩 모델을 통해 단어 벡터를 얻고 이후 Multi-channel CNN 모델을 통해 문장을 분류하는 방법을 제안하였다 [16]. 이는 CNN 모델은 컴퓨터 비전 분야의 문제를 해결하기 위해 고안되었지만, 자연어 처리에 대해서도 효과적임을 보여주었다. 이후 언어의 순차적인 정보를 반영하기 위해 Long Short-Term Memory(LSTM) 기반의 문장 분류 모델 Cheng et al.[17]이 제안되었으며 순차적인 정보를 반영하는 LSTM과 주변 단어들의 조합 패턴을 반영할 수 있는 CNN의 장점을 모두 채택하기 위해 LSTM과 CNN을 동시에 활용한 분류 방법론인 Zhou et al.[18]이 제안되었다. LSTM 기반의 모델은 입력 길이가 길어짐에 따라 Vanishing Gradient 문제가 발생하며, 하나의 고정된 크기의 벡터에 모든 입력 정보를 압축하기 때문에 정보가 손실되는 문제가 존재한다. 이 문제를 해결하기 위해 각 입력 스텝의 중요도를 반영하는 Attention Mechanism이 등장하였다 [19]. Yang et al.[20]에서는 이를 추가한 실험을 통해 Attention이 성능 향상에 도움이 됨을 보였다. 이후 RNN을 사용하지 않고 Attention만으로 구현된 Transformer가 제안되고, 이를 기반으로 하는 BERT 등의 다양한 언어 모델들이 등장하여 대부분의 자연어처리 분야에서 SOTA를 달성하였다.

2.2 특허 분류를 위한 연구

이전부터 특허 문서를 자동적으로 분류하기 위해 머신러닝과 딥러닝을 이용한 다양한 방법들이 제안되어 왔다. 머신러닝을 이용한 특허 분류의 대표적인 방법으로는 통계 기반 방법인 Naive Bayes(NB) 및 Support Vector Machine(SVM)와 예제 기반 범주화 방법인 K-Nearest Neighbor(KNN) 등이 있다 [21]. 딥러닝 기술의 발전과 함께 여러 분야에서 딥러닝 모델이 머신러닝 모델보다 높은 성능을 보이며 딥러닝을 이용한 특허 분류 방법에 대한 연구가 활발히 이루어지고 있다. Mattyws et al.[22]은 2017년에 발표된 논문으로 Word2Vec 워드 임베딩 모델과 LSTM 기반의 모델을 통해 50개의 IPC subclass를 분류하고자 하였다. 사전 학습된 워드 임베딩 모델을 사용하지 않고 학습을 통해 직접 모델을 생성하였으며, 임베딩 벡터의 크기에 따른 성능을 비교하였다. 결과적으로 임베딩 벡터 크기를 클 때 더 높은 성능을 보이며 63%의 분류 정확도를 기록하였다. Jie et al.[23]은 특허의 멀티 레이블 분류를 위해 CNN을 기반의 N-Gram Feature Extraction 후 Bi-LSTM을 통해 분류를 진행하는 hierarchical feature extraction model(HFEM)을 제안하였다. 이들은 CNN을 통해 Feature Extraction을 진행하는 것이 워드 임베딩만 진행하였을 경우보다 높은 성능을 보인다고 주장하였다. Shaobo et al.[24]은 CNN과 워드 임베딩 기반의 W2V-CNN 특허 분류 모델을 이용하여 637개의 특허 클래스를 분류하고자 하였다. 이들은 특허 문서의 title과 abstract를 추출하고 이를 통해 skip-gram 방식으로 학습한 임베딩 모델을 생성하였다. 이후, 여러 크기의 필터를 갖는 CNN 네트워크와 Max-pooling 층을 통해 local feature를 추출하고 그 결과를 결합함으로써 Feature 계층을 학습하고자 하였으며, 마지막으로 Fully connected 층

을 거쳐 특허 클래스를 예측하였다. [24]에서 제안하는 W2V-CNN 모델은 [그림 2-1]과 같은 네트워크 구조를 갖는다.



[그림 2-1] W2V-CNN 네트워크 구조

Lee et al.[10]에서는 BERT의 Fine-Tuning을 통해 특허 문서를 분류하고자 하였다. 이들은 BERT에 [CLS] 토큰만을 입력으로 하는 간단한 Classifier 층을 추가하고 Fine-Tuning 하는 것을 통해 656개의 CPC 코드에 대해 84.26%의 분류 정확도를 보였다. 이는 특허 분류 모델에서 가장 높은 분류 정확도를 기록하였다.

2.3 Extreme Multi-label Text Classification

XMTC는 자연어처리 Task의 일종으로 수백, 수천 개의 무수히 많은 후보 클래스 중에서 입력 텍스트와 관련성 높은 클래스들을 찾아내는 것을 목표로 하며 item categorization, web page tagging, news annotation 등에 적용되어 유용하게 사용되고 있다 [25-28]. 이전에 XMTC 방법은 크게 임베딩 기반의 방법과 트리 기반의 방법으로 구분되었지만 최근 딥러닝 기술이 발전하면서 딥러닝 기반의 방법이 XMTC를 위한 효과적인 방법으로 떠오르고 있다.

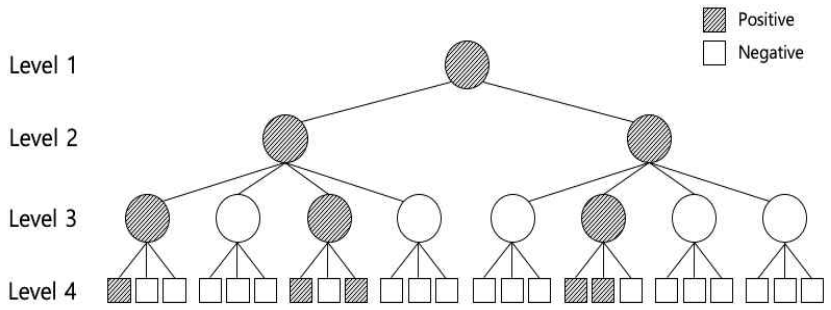
임베딩 기반 방법은 학습 레이블이 low-rank라고 가정하고 레이블의 상관관계는 최대한 보존하면서 고차원 레이블 벡터를 낮은 차원의 공간으로 축소하는 것을 목표로 한다. 가장 대표적인 임베딩 기반 방법은 Sparse Local Embeddings for Extreme Classification(SLEEC) 방법으로 기존의 low-rank 가정으로부터 벗어나서 가장 가까운 레이블 벡터 사이의 쌍방향 거리를 보존하는 임베딩을 학습하여 분류 정확도를 높였다 [25]. 트리 기반 방법은 의사 결정 트리를 기초하여 제안된 방법으로 각 비단말 노드에서 문서를 재귀적으로 분할하여 각 단말 노드의 문서는 유사한 레이블 분포를 공유한다. 이 방법은 임베딩 기반의 방법보다 상당히 높은 분류 정확도를 보이며 대표적인 분류 방법으로 FastXML이 있다 [26]. 이 방법은 label space 대신 feature space를 분할하여 기존의 트리 기반의 방법들보다 향상된 분류 성능을 보였다.

기존의 임베딩 기반의 방법과 트리 기반의 방법은 단어 사이의 의미적 의존성을 포착하기 어려운 반면 딥러닝 기반의 방법은 입력 텍스트의 의미적 의존 포착이 가능하다. 이러한 이점을 기반으로 XMTC를 위해 CNN, RNN, Attention Mechanism 등을 이용한 방법들이 제안되었으며 가장

높은 분류 정확도를 보인 분류 모델은 Bi-LSTM과 Attention Mechanism을 이용한 AttentionXML과 Label-Aware document representation model via a Hybrid Attention neural network (LAHA)이다 [27,28].

2.3.1 AttentionXML

AttentionXML은 2018년 발표되어 XMTC에서 SOTA를 달성한 레이블 트리 기반의 딥러닝 모델이다. 이들은 기존의 딥러닝 모델이 각 레이블에 대한 중요한 하위 텍스트를 포착하지 못하며, 수많은 레이블에 대한 확장성이 부족하기 때문에 이를 극복한 모델을 제안하고자 하였다 [27]. AttentionXML은 두 가지의 특징을 가지고 있다. 첫 번째는 각 레이블에 대해 텍스트의 가장 관련성이 높은 부분을 포착할 수 있는 Multi-label Attention이며, 두 번째는 수백만 개의 레이블을 처리할 수 있는 얕고 넓은 probabilistic label tree(PLT)이다 [29]. 이 때 PLT의 단말 노드 하나는 레이블 하나를 나타낸다.



[그림 2-2] Probabilistic Label Tree

AttentionXML은 워드 임베딩을 위해 사전 학습된 300차원의 GloVe 모델을 사용하였으며, 양방향 문맥을 반영하기 위해 Bi-LSTM 층을 이용하였다. 이들은 실험을 통해 모든 XMTC를 위한 데이터 셋에서 기존의 분류 방법들보다 AttentionXML이 높은 분류 정확도 보이며 Multi-label Attention이 효과적임을 증명하였다.

2.3.2 LAHA

LAHA는 2019년 발표된 딥러닝 기반의 XMTC를 위한 모델로 레이블 간의 상관관계를 표현할 수 있으면서 텍스트 간의 종속성을 충분히 포착할 수 있는 분류 방법을 제안하고자 하였다 [28]. 기존의 트리 기반의 방법과 임베딩 기반의 방법은 레이블의 계층구조를 구축하거나 저차원의 잠재공간을 학습하여 레이블 상관관계를 찾는 것이 가능하였다. 이에 반해, 기존의 딥러닝 기반의 방법은 텍스트 간의 종속성을 충분히 포착할 수 있도록 문서를 잘 표현하지만, 레이블 상관관계를 반영하지 않는다. 딥러닝 기반의 성공적인 모델인 AttentionXML은 각 레이블에 대한 중요한 하위 텍스트를 포착하였지만, 레이블 사이의 상관관계와 구조를 고려하지 않는다.

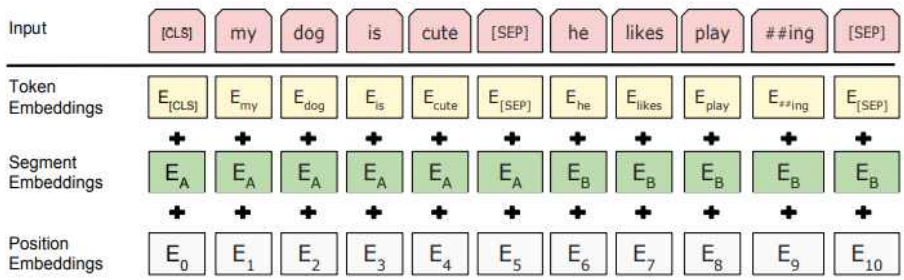
LAHA는 AttentionXML과 마찬가지로 Bi-LSTM과 Attention Mechanism을 기반으로 구축된 딥러닝 네트워크이다. 레이블의 동시 등장 그래프를 기반으로 하여 레이블 구조를 충분히 표현할 수 있는 레이블 임베딩을 생성하고 이를 이용하여 레이블 상관관계를 반영하고자 하였다. LAHA는 크게 세 부분으로 구분된다. 첫 번째, 문서를 구성하는 단어들이 각 레이블에 미치는 영향에 대한 가중치를 얻기 위한 Self-Attention. 두 번째, 잠재된 공간에서 단어와 레이블 사이의 의미적 연관성을 결정하기

위한 Interaction-Attention. 세 번째, 앞의 두 부분에서 얻은 정보를 적절히 조합하여 최종적인 예측을 얻기 위한 Attention Fusion. 이러한 구조를 갖는 LAHA는 최종적으로 모든 데이터 셋에서 AttentionXML보다 높은 분류 성능을 보이며 레이블 상관관계 반영의 중요성을 증명하였다.

2.4 BERT

BERT는 Transformers 구조를 활용하고 대용량의 말뭉치를 이용하여 사전 학습한 언어 모델이다 [30]. 사전 학습은 unlabeled data를 이용한 Masked Language Model Task와 Next Sentence Prediction Task를 통해 진행된다. 이후, 특정 task를 갖는 labeled data를 이용하여 transfer leaning 시킨다. BERT는 네트워크의 사이즈에 따라 여러 모델이 존재하며 가장 많이 이용되는 BERT-base 모델은 트랜스포머 블록 12개와 12개의 self-attention head로 구성되며 768 차원의 hidden 사이즈를 갖는 인코더를 포함하고 있다. 입력으로 받을 수 있는 문장의 최대 길이는 512로 제한된다.

Word2Vec과 같은 기존 워드 임베딩 모델은 문맥을 고려하지 못하며 동음이의어 문제를 처리할 수 없다. 하지만 BERT는 문맥을 고려하여 단어의 위치와 형태에 따라 같은 단어라도 다른 벡터값을 갖는다 [30]. 예를 들어 'constitution of the society'에서 constitution은 조직을 나타내지만 'nervous constitution'은 성격을 나타낸다. 이 때 기존의 워드 임베딩 모델은 constitution을 같은 벡터값으로 표현하지만, BERT는 두 단어를 다른 벡터값으로 표현한다.



[그림 2-3] BERT input representation [30]

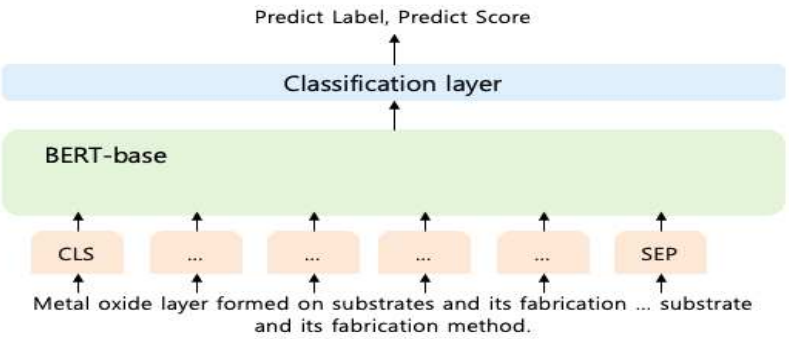
BERT의 입력은 세 가지 임베딩 값의 합으로 이루어지며 [그림 2-3]에서 볼 수 있듯이, BERT는 Transformer에서 사용한 Position embedding을 사용하며 이를 통해 단어의 순차 정보를 반영할 수 있다. BERT에는 두 개의 특별한 토큰이 존재한다. 먼저 첫 번째는 문장을 구분하고 문장의 끝을 의미하는 [SEP] 토큰이다. 두 번째는, 항상 문장의 첫 번째 토큰으로 등장하는 [CLS] 토큰으로 모든 층을 거치고 나면 전체 시퀀스를 결합한 의미를 갖는다. 여기에 간단한 classifier를 올리면 단일 문장, 또는 연속된 문장의 분류를 쉽게 할 수 있다.

사전 학습된 언어 모델을 사용하는 방식에는 크게 두 가지가 있다. 하나는 특정 작업을 수행하는 네트워크에 사전 학습된 언어 모델을 추가하여 사용하는 방식인 feature-based approach 방식이며, 또 다른 하나는 Fine-Tuning approach 방식이다. Fine-Tuning은 사전 학습된 모델로부터 downstream 작업에 맞게 학습할 수 있도록 하는 효과적인 모델링 전략으로 기존 BERT 모델의 내부 구조를 변경하지 않고 출력 벡터들을 입력으로 하는 몇 개의 얕은 계층만을 추가하는 것만으로 특정 작업을 수행하는 모델이 된다.

제 3 장 제안 방법

기존의 사전 학습된 BERT를 통한 분류는 마지막 층에 풀링 레이어와 출력 레이어로만 구성된 간단한 Classifier를 추가하여 Fine-Tuning 하는 것만으로도 높은 성능의 분류 정확도를 얻을 수 있다고 주장하였다. 이때 BERT 분류 모델에서는 [CLS] 토큰만을 이용한다. 하지만 [CLS] 토큰만으로는 전체 시퀀스 정보를 충분히 표현할 수 없으며, 해당 모델은 레이블 구조와 레이블 상관관계를 전혀 고려하지 못한다.

본 연구에서는 [그림 3-1]과 같이 기존의 BERT 분류 모델의 개선을 통해 특허의 다중 레이블을 분류 성능을 향상시키는 것을 첫 번째 목표로 한다. 이를 위해 본 논문에서는 기존의 BERT를 이용한 다중 레이블 분류 방법의 단점을 개선하기 위해 전체 시퀀스 정보를 이용하고 레이블 사이의 구조와 상관관계를 고려한 방법을 제안한다. 또한, 분류 시 클래스 불균형으로 인해 발생하는 문제를 손실 함수를 통해 극복하는 것을 두 번째 목표로 한다. 본 섹션에서는 본 논문에서 제안하는 극한 다중 레이블 분류를 위한 모델과 모델을 구성하는 각 층에 대해 설명하고, 클래스 불균형을 해결하기 위해 채택한 손실 함수에 대해 기술한다.



[그림 3-1] BERT 기반의 다중 레이블 분류 모델

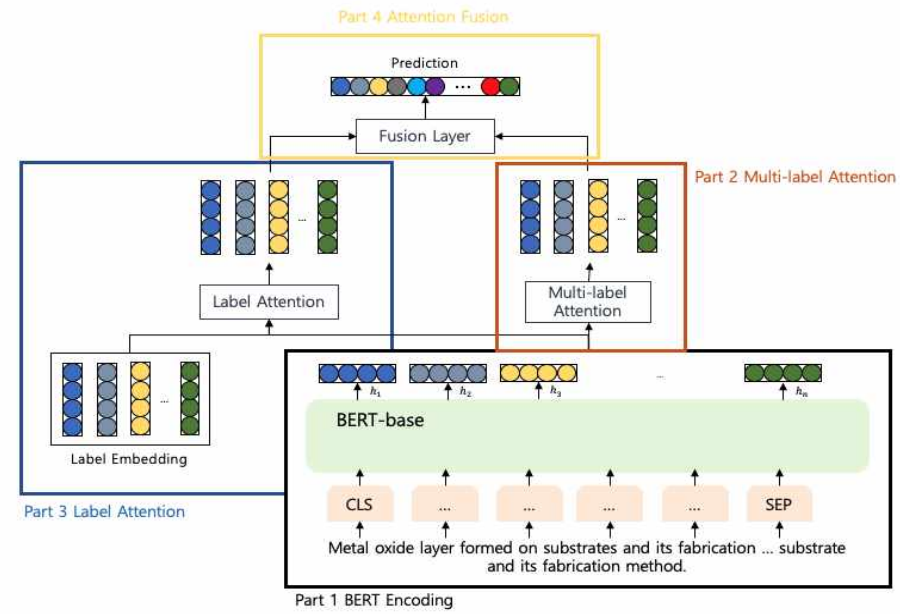
3.1 제안 모델

3.1.1 개요

다중 레이블 텍스트 분류 모델은 입력 텍스트에 대해 관련성이 높은 레이블들로 분류하는 것을 말한다. 본 논문에서는 Devlin et al.의 연구 BERT와 Huang et al.의 연구 LAHA를 바탕으로 특허 문서의 다중 레이블 분류를 위한 딥러닝 모델을 제안하고자 한다 [28,30].

[그림 3-2]은 본 연구에서 제안하는 다중 레이블 분류 모델의 전반적인 구조로 크게 네 개의 부분으로 구성된다. 첫 번째는 입력 시퀀스를 적절한 벡터로 변환하기 위한 문장 인코딩 단계로 문맥적 표현이 가능하며 자연어 처리의 대부분 작업에서 강력함을 보이는 BERT를 사용하였다. 두 번째는 각 레이블에 대해 토큰의 중요성을 결정하기 위한 Multi-label Attention Layer이며, 세 번째는 레이블 임베딩과 시퀀스 벡터를 이용하여 단어와 레이블 사이의 연관성과 레이블에 대한 단어의 가중치를 결정하기 위한 Label Attention Layer이다. 마지막으로, 앞의 두 부분을 통해 얻은 정보를 적절히 통합하여 최종적으로 레이블을 예측하기 위한 Fusion Layer 부분으로 구성되어 있다. 또한, 제안 모델에서는 특허 데이터의 클래스 불균형 문제를 극복하기 위해 일반적으로 사용되는 이진 교차 엔트로피 손실 함수를 대신하여 Computer Vision 분야에서 클래스 불균형을 극복하기 위해 주로 사용되는 Normalized Focal Loss 손실 함수를 사용하여 학습하였다.

이를 통해 레이블 구조, 레이블 상관관계, 각 레이블에 따른 토큰의 가중치 모두를 충분히 고려하면서 클래스 불균형으로 인해 발생하는 문제를 극복한 모델 생성이 가능하다.



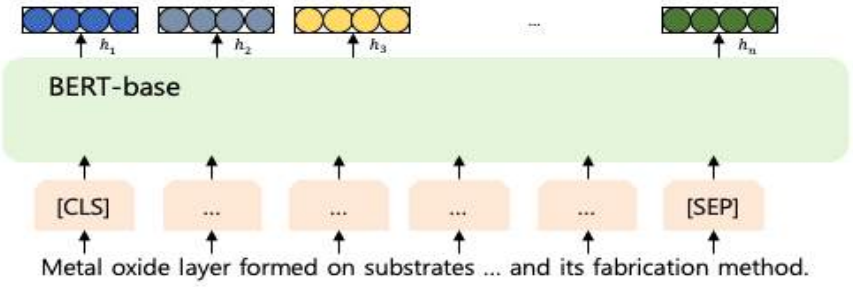
[그림 3-2] 제안 모델의 네트워크 구조

3.1.2 Feature Embedding

본 연구에서는 원시 데이터를 저차원의 벡터로 표현하기 위해 두 개의 인코더를 사용한다. 첫 번째는 문장을 구성하는 토큰들을 적절한 벡터로 표현하기 위한 문장 인코더이다. 문장 인코딩 과정은 N개의 원본 학습 데이터 $D=(x_1, y_1), \cdots, (x_N, y_N)$가 주어졌을 때 각 데이터 x_i를 구성하는 n개의 토큰을 각각 d차원 벡터 $v_n \in R^d$으로 변환하는 역할을 한다.

본 연구에서는 BERT 모델의 최종 출력을 단어의 벡터 표현으로 사용하고, 전체 시퀀스 정보를 충분히 반영한 텍스트 분류를 위해 출력된 토큰 벡터 모두를 다중 레이블 예측에 사용할 것을 제안한다. 적절한 전처리와 토큰화 과정을 거친 토큰들은 [그림 3-3]과 같이 BERT를 통해 문맥을 잘 표현할 수 있는 저차원의 벡터들로 변환된다. 본 연구에서는 BERT base

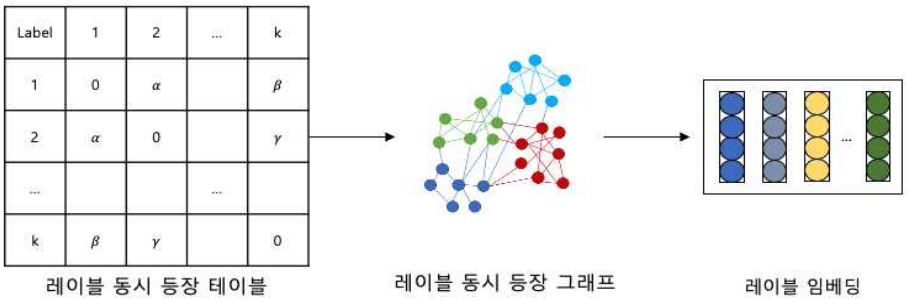
모델을 사용하였기 때문에 각 토큰은 768차원의 벡터로 변환되며, BERT는 다중 레이블 분류를 위해 추가된 층들과 함께 Fine-Tuning 된다.



[그림 3-3] BERT를 이용한 문장 인코딩 [30]

두 번째는 레이블의 동시 등장 그래프를 기반으로 테이블 구조와 상관관계를 포착하고 레이블을 저차원의 벡터 표현으로 변환하기 위한 레이블 임베딩이다. 본 연구에서는 레이블 상관관계를 포착하고 이를 잘 표현할 수 있는 임베딩을 얻기 위해 LAHA에서 제안한 방식을 사용하였으며, 그 과정은 [그림 3-4]와 같다. 가장 먼저 학습 데이터를 기반으로 하여 레이블의 동시 등장 테이블을 만든다. 이후 레이블 구조와 상관관계를 추출하기 위해 각 레이블이 노드로 표현되는 동시 등장 그래프를 생성한다 [31]. 생성된 그래프는 두 레이블이 한 번 이상 동시에 등장한다면 두 노드는 엣지로 연결한다. 레이블의 동시 등장 그래프를 기반으로 레이블을 저차원의 공간으로 표현하기 위해 강력하면서 일반적으로 사용되는 node2vec을 사용하였다 [32]. node2vec은 Breadth-first sampling과 Depth-first sampling을 활용하여 그래프의 노드를 학습함으로써 노드들의 관계를 더욱 잘 포착할 수 있다. 이를 통해 저차원 공간에서 인접한 레이블이 유사한 표현을 갖도록 하여 레이블 구조를 최대한 보존한 레이

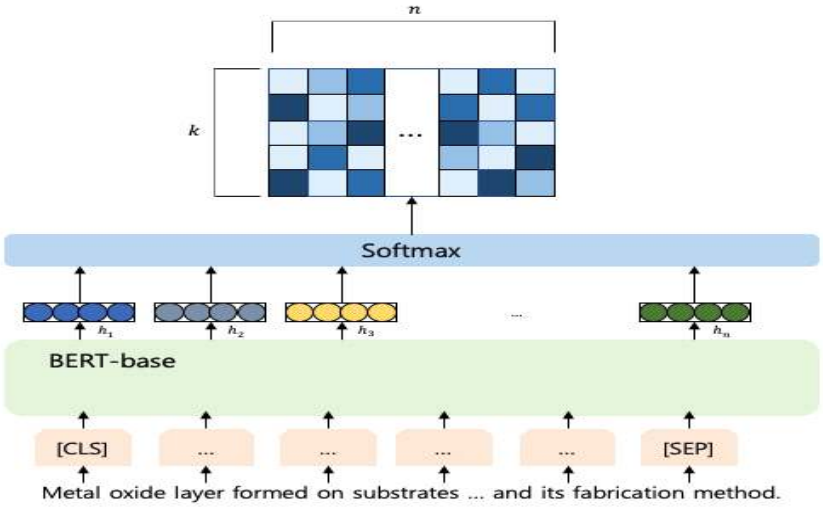
블 임베딩을 얻을 수 있다. 결과적으로 각 레이블 $l_i \in R^d$로 표현되며 전체 레이블 임베딩 $L = (l_1, l_2, ..., l_k) \in R^{k \times d}$으로 표현된다.



[그림 3-4] 레이블 임베딩 생성 과정

3.1.3 Multi-label Attention Layer

Attention Mechanism은 딥러닝 기반의 번역, 관계 추출 및 음성 인식과 같은 자연어 처리 분야뿐만 아니라 다양한 분야에 적용되어 성공적으로 사용되고 있다. 일반적으로 사용되는 Attention은 해당 문서를 구성하는 단어들이 특정 작업에 대해 얼마나 많은 가중치를 갖는지만을 평가한다. 즉, 문서 전체에서 어떤 문맥이 중요한가만을 평가한다. 하지만 클래스가 여러 개일 경우 클래스마다 중요한 문맥과 단어가 다를 수 있다.



[그림 3-5] Multi-label Attention Layer

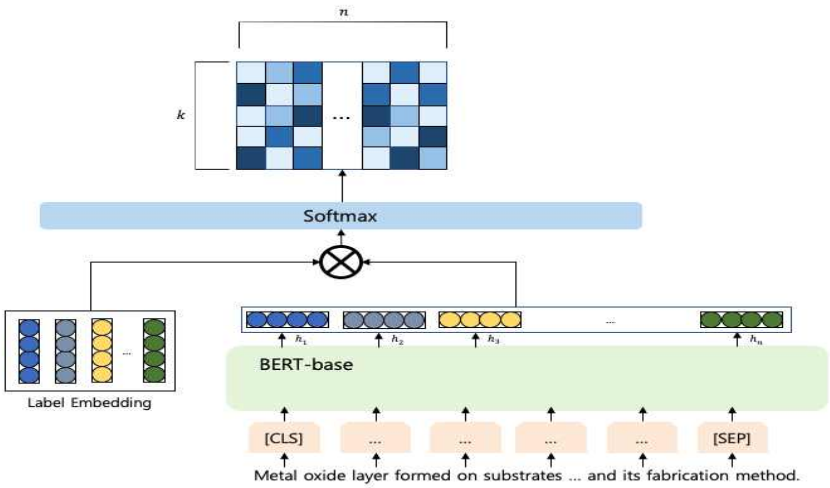
본 연구에서는 이러한 특수성을 고려하기 위해 [27]에서 제안한 Multi-label Attention Mechanism을 활용한 Multi-label Attention Layer를 사용하였다. [그림 3-5]는 Multi-label Attention Layer에서 Multi-label Attention에 의해 레이블에 따른 문맥의 가중치 $A \in R^{k \times n}$를 구하는 과정을 나타낸다. 그림과 같이 softmax 층을 지나 각 레이블에 따라 다른 가중치를 갖는 A를 얻을 수 있다. 이렇게 얻은 가중치 A와 BERT의 최종 출력 $H \in R^{d \times n}$를 결합하여 다중 레이블 특성을 고려한 문서 표현 $C_s \in R^{k \times d}$를 얻게 된다. 가중치 A와 문서 표현 C_s를 얻기 위한 연산은 아래와 같다.

$$A = softmax(W_{s1}H) \ , \ C_s = AH$$

이때 $W_{s1} \in R^{d \times k}$는 파라미터로 Fine-Tuning시에 학습된다.

3.1.4 Label Attention Layer

Label Attention Layer는 LAHA의 Interaction Attention을 기반으로 하여 단어와 레이블의 의미적 연관성을 찾고 이를 반영한 문서 표현을 얻기 위한 층으로 [그림 3-6]과 같은 과정을 통해 연관성을 결정한다[28]. 해당 층에서는 BERT의 출력값 H와 레이블 동시 등장 그래프 기반의 레이블 임베딩 $L \in R^{k \times d}$를 활용하며, H와 L은 반드시 같은 차원의 벡터로 표현되어야 한다.



[그림 3-6] Label Attention Layer

Attention Query $Q = W_q L$와 Attention Key $K = W_k H$의 Interaction mechanism을 통해 단어와 레이블의 의미적 연관성을 찾게 된다. 이후, softmax를 통한 정규화 과정을 거쳐 가중치 $\hat{A} \in R^{n \times k}$와 이를 반영한 문서 표현 $C_l \in R^{k \times d}$를 얻게 된다.

$$K = W_{l1}H \ , \ Q = W_{l2}L, \ \hat{A} = softmax(Q,K^T), \ C_l = \hat{A}H$$

Multi-label Attention Layer에서와 마찬가지로 $W_q \in R^{d \times d}$와 $W_k \in R^{d \times d}$는 파라미터로 Fine-Tuning시에 학습된다.

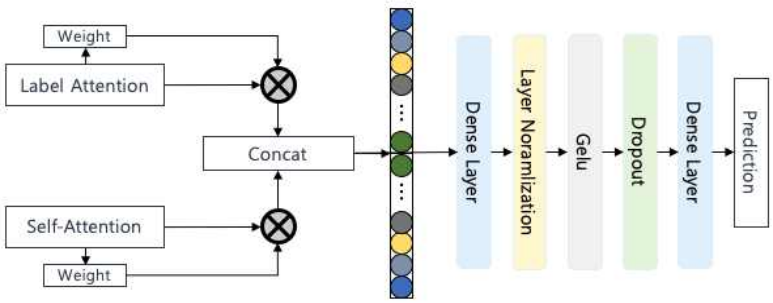
3.1.5 Fusion Layer

Multi-label Attention Layer와 Label Attention Layer를 통해 레이블을 고려한 문서 표현 C_s와 C_l를 얻었다. C_s는 문서 내용을 잘 반영하였으며 C_l는 레이블 구조를 잘 반영하고 있다. Fusion Layer는 서로 다른 장점을 가진 C_s와 C_l를 적절히 조합하고 두 문서 표현의 장점을 모두 반영한 최종 레이블 예측을 위한 층이다. 본 연구에서 제안하는 모델에서는 [28]의 Attention Fusion에서 영감을 받아 새로운 Fusion Strategy를 제안한다. Attention Fusion은 C_s와 C_l 각각을 입력으로 하는 fully connected layer와 sigmoid 함수를 통해 가중치 α와 β를 구한다. 이후, 정규화 과정 $\alpha_j = \alpha_j / \alpha_j + \beta_j$ 와 $\beta_j = 1 - \alpha_j$를 거쳐 각 문서 표현이 레이블 예측에 차지하는 비중을 계산한다. 이렇게 계산된 비중을 각각 반영한 두 문서 표현의 합을 최종 레이블 예측을 위한 문서 표현으로 사용한다.

$$\alpha = \sigma(C_s W_{f1}) \ , \ \beta = \sigma(C_l W_{f2}) \ , \ C = \alpha \times C_s \ + \ \beta \times C_l$$

각 문서 표현의 비중을 반영한 문서 표현의 합은 정보의 손실이 발생하여 두 문서 표현의 장점을 온전히 보존하는 데 부족함이 있다고 생각하였다. 본 연구에서는 두 문서 표현의 장점을 더 잘 추출할 수 있는 새로운

attention fusion strategy를 제안하고자 한다. 본 연구에서 제안하는 방법은 [그림 3-7]과 같은 네트워크 구조를 가지며 Attention Fusion에서와 마찬가지로 가중치 α와 β를 구하고, 정규화 과정을 거친다. 이후 정규화된 가중치를 반영한 문서 표현을 연결하여 최종 표현 $C \in R^{k \times 2d}$를 얻는다. 단순히 두 문서 표현을 합하는 방식에서 연결하는 방식으로 변경함으로써 정보의 손실을 방지하고 두 문서 표현의 장점을 더욱 잘 보존할 수 있다고 생각하였다. 이후의 최종 예측 $\hat{y}$를 얻기 위해 fully connected와 activation function으로 구성된 간단한 Classifier층을 거친다. 최종 예측 $\hat{y} = \sigma(W_o(g(LN(W_c C))))$와 같으며 $W_c \in R^{r \times 2r}$, $W_o \in R^{1 \times r}$, LN은 Layer Normalization, g는 activation function GeLU, σ은 최종 예측값을 확률로 변환해주기 위한 sigmoid function을 의미한다.



[그림 3-7] 제안 모델의 Fusion Layer 네트워크 구조

3.2 손실 함수

손실 함수는 딥러닝 모델의 예측값과 실제 정답 값의 손실, 오차를 이용하여 신경망이 학습할 수 있도록 해주는 지표이다. 학습의 최종적인 목표는 이 손실 함수가 최소화되도록 하는 Weight와 Bias를 찾는 것이다. 일반적인 다중 레이블 분류를 위해서는 이진 교차 엔트로피(Binary Cross Entropy) 함수를 주로 사용하며 수식은 다음과 같다.

$$BCE(x) = \frac{1}{k}\sum_{i=1}^{k} -(y_i \log(\hat{y_i}) + (1-y_i)\log(1-\hat{y_i}))$$

이진 교차 엔트로피 손실 함수를 이용하여 클래스 불균형이 심한 데이터를 학습할 때 자주 등장하는 클래스 위주로 학습이 진행되게 된다 [9]. 결과적으로 데이터가 많은 클래스에서는 높은 정확도를 보이지만 데이터가 적은 클래스는 낮은 정확도를 보이며 클래스별 분류 정확도 평균을 비롯하여 전체적인 성능이 감소한다.

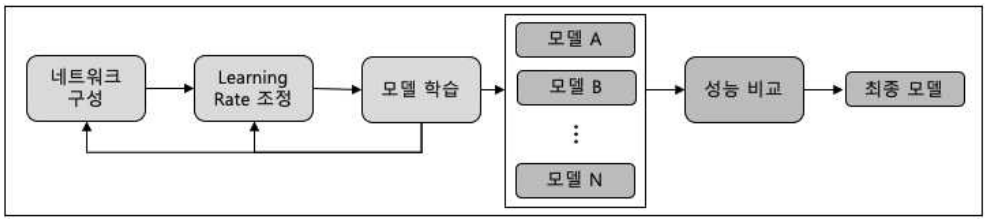
본 연구에서는 클래스 불균형 문제를 극복하고자 하였으며 Computer Vision 분야에서 주로 활용되는 Normalized Focal Loss를 사용하였다. Focal Loss는 Object Detection에서 오브젝트와 배경 사이에서 발생하는 클래스 불균형을 극복하기 위해 등장한 손실 함수이다 [33]. 자주 등장하지 않는(Hard Positives) 클래스에 높은 가중치를 부여하고 자주 등장하는(Easy Negatives) 클래스에는 작은 가중치를 할당하여 학습을 진행한다. 이를 통해 Hard Positive 클래스 위주로 모델이 학습되는 것을 방지할 수 있다. Normalized Focal Loss는 학습이 진행되며 정확도가 향상될수록 손실 함수의 총합이 감소하여 손실 함수의 효과가 사라지는 문제를

해결하기 위해 등장한 손실 함수로 아래와 같은 수식을 갖는다 [34].

$$NFL(i,\hat{y}) = -\frac{1}{P(\hat{y})}(1-\hat{y}_i)^{\gamma}\log\hat{y}_i \qquad P(\hat{y}) = \frac{1}{k}\sum_{1}^{k}(1-y_i)^{\gamma}$$

3.3 모델 개발 절차

[그림 3-8]은 본 연구에서 진행한 모델 개발 절차이며 네트워크 구성 단계에서는 Bias와 활성화 함수 유무에 따른 여러 모델을 구성하고 이를 학습하였다. 다음으로 Learning Rate를 달리하며 모델을 학습하여 여러 모델을 만들고 학습된 모델의 성능을 비교, 평가하여 최종적으로 가장 높은 정확도를 보이는 모델을 선정하였다.



[그림 3-8] 모델 개발 절차

제 4 장 실험 및 분석

4.1 데이터 셋

이 섹션에서는 본 연구에서 사용한 데이터 셋에 대해 설명한다. 본 연구에서는 미국, 유럽, 일본에서 출원된 영문 특허 989,955개의 원본 특허 데이터를 사용하였다. 분류하고자 하는 클래스는 모든 기술 및 산업 분야를 포함하도록 하는 1,383개로 구성되어있다. 특허 데이터는 고도로 구조화된 문서로 다양한 필드들로 구성되며 본 연구에서 IPC, 특허 제목, 요약, 청구항 필드를 분류를 위한 텍스트 데이터로 사용하였다.

본 연구에서 활용한 특허 데이터 셋은 가장 데이터가 적은 클래스의 경우 오직 8개의 데이터만 존재하며 가장 많은 데이터를 갖는 클래스는 3만개 이상의 데이터를 포함하고 있다. [표 4-1]는 각 데이터 셋의 데이터의 속성을 나타낸다.

[표 4-1] 실험을 위한 특허 데이터셋의 통계 정보

N	V	M	D	L	$\hat{L}$	$\check{L}$	T
791,839	99,058	99,058	768,151	1383	1.44	1.43	512

N은 학습 데이터 수, V는 검증 데이터 수, M은 실험 데이터 수, D는 학습 데이터에 포함된 총 단어의 수, L은 분류하고자 하는 레이블 수, $\hat{L}$은 학습 데이터의 문서당 평균 레이블 수, $\check{L}$은 테스트 데이터의 문서당 평균 레이블 수, T는 입력 데이터의 길이를 나타낸다.

4.1.1 텍스트 전처리

자연어 처리(NLP)에서 전처리는 필수적인 과정이며 모델의 복잡도를 감소시키고 모델 성능에 여러 방면으로 영향을 미친다. 본 논문에서는 Non-ASCII Removal, Tag Removal, Contractions Fix, Stop-words Removal, Lemmatization 5가지의 주요 표준 전처리 단계를 적용하였다. Non-ASCII Removal 과정은 특수 문자와 같이 텍스트 분류에 있어 의미를 갖지 않는 문자들을 제거하기 위한 과정이다. 본 연구에서는 Elasticsearch를 통해 데이터를 수집하였으며 수집된 데이터에는 태그가 포함되어 있다 [35]. Tag Removal은 이렇게 포함된 태그를 제거하기 위한 과정이다. 영어 데이터의 경우 축약형 표현이 존재하며 딥러닝 모델은 이러한 축약형 표현을 인식할 수 없어 모델의 혼란이 가중된다. Contractions Fix는 축약형 표현을 일반적인 표현으로 풀어주는 역할을 한다. Stop-word는 전치사, 한정사, 관사 등과 같이 모든 문서에서 등장 빈도가 높아 추가적인 정보를 갖고 있지 않은 단어들을 의미하며 이를 제거하기 위해 파이썬의 NLTK(Natural Language Toolkit) 패키지의 'English' stopword 리스트를 이용하였다. 이후 단어의 원형을 찾기 위한 Lemmatization 과정을 위해 NLTK 패키지의 WordNetLemmatizer 함수를 이용하였다. 각 전처리 과정에 따른 텍스트 데이터의 변화는 [표 4-2]와 같다.

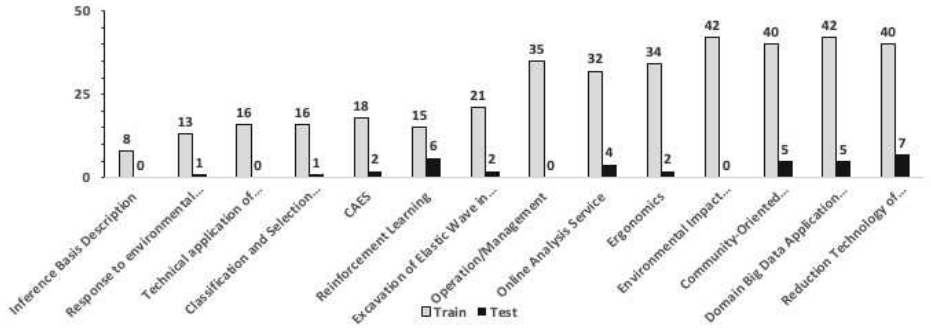
[표 4-2] 전처리에 따른 텍스트 데이터 변화 예시 [36]

Preprocessing	Before	After
Contractions Fix	It‘s an object to provide a flexible light-emitting device with high reliability in a simple way.	It is an object to provide a flexible light-emitting device with high reliability in a simple way.
Stop-words Removal	The present invention directed to a new semiconductor file comprising of metal oxide grown on a substrate and its fabrication method.	The present invention directed new semiconductor file comprising metal oxide grown substrate fabrication method.
Lemmatization	The present invention directed new semiconductor file comprising metal oxide grown substrate fabrication method.	The present invention direct new semiconductor file comprise metal oxide grown substrate fabrication method.

4.1.2 데이터 분할

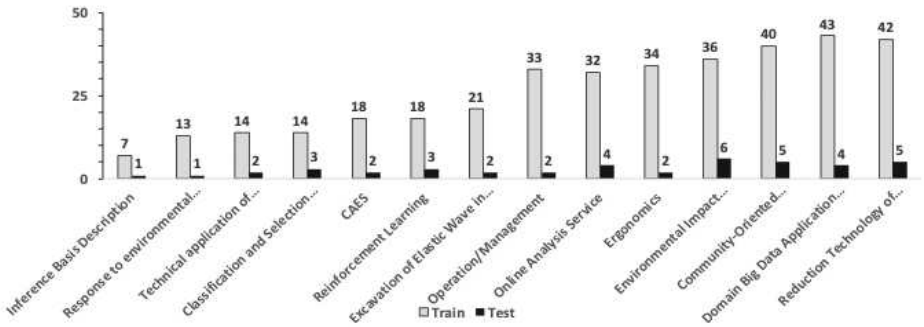
실험에 앞서 학습을 위해 데이터 셋을 학습 데이터, 검증 데이터, 실험 데이터로 분할할 필요가 있다. 일반적인 데이터 분할에서는 계층적 데이터 샘플링을 사용한다 [37]. 하지만 다중 레이블 데이터를 일반적인 계층적 데이터 샘플링을 통해 분할을 진행할 경우 문제가 발생할 수 있다. [그림 4-1]은 일반적인 계층적 데이터 샘플링 방법을 통해 다중 레이블 데이터를 분할하였을 때 희귀 클래스의 분할 예시로 비율이 작은 실험 데이터에 희귀 클래스에 해당하는 데이터가 포함되지 않는 경우가 발생한다 [38]. 이와 같은 데이터 샘플링은 실험의 진행에 문제가 되지 않지

만, 실험 결과에 대한 신뢰성에 문제가 될 수 있다. 비정상적으로 분할된 데이터 셋을 이용한 실험은 희귀 클래스가 포함되지 않을 수 있으므로 실험에 대한 신뢰성을 보장하지 못한다.



[그림 4-1] 다중 레이블을 고려하지 않은 샘플링의 데이터 분포

본 연구에서는 Sechidis et al.의 연구를 파이썬을 이용해 구현한 iterative-stratification 패키지의 Multi-Label Stratified Kfold를 통해 데이터 샘플링을 진행하였다[38]. [그림 4-2]은 다중 레이블 특성을 고려한 데이터 샘플링 시 희귀 클래스 분포를 나타내며, 모든 희귀 클래스에 대해 하나 이상의 데이터가 실험 데이터 셋에 포함된 것을 알 수 있다.



[그림 4-2] 다중 레이블을 고려한 샘플링의 데이터 분포

4.1.3 데이터 분포

[그림 4-3]은 본 연구에서 사용한 특허 데이터의 클래스에 따른 데이터의 분포를 나타내는 그림이다. 데이터가 많은 상위 300개 클래스에 해당하는 데이터가 전체 문서의 63.44%를 차지하는 반면 하위 300개 클래스에는 전체의 2.57%만의 데이터가 포함되어 있다. 가장 많은 데이터를 포함하고 있는 클래스는 31,402개의 데이터가 포함되어 있지만 가장 적은 데이터를 포함하는 클래스는 8개의 데이터만이 포함된다. 이를 통해 본 연구에서 사용하는 특허 데이터는 클래스에 따른 데이터 분포가 매우 불균형하다는 것을 알 수 있다.

[그림 4-3] 실험 데이터의 클래스별 데이터 분포

4.2 실험 및 평가

본 연구에서는 제안하는 모델의 특허 분류 성능이 우수하다는 것을 보이기 위해 특허 데이터를 이용한 다양한 실험을 진행하였다. 먼저 제안하는 attention fusion strategy의 평가를 위해 다른 fusion strategy와의 비교 실험을 진행하였으며 손실 함수에 따른 실험을 진행하여 성능 변화 및 클래스 불균형 극복을 확인하였다. 마지막으로 본 논문에서 제안하는 모델의 우수성을 보이기 위해 다른 베이스라인 모델들과 제안 모델의 특허 분류 성능을 비교하고 이를 분석하였다. [표 4-3]은 제안 모델 및 베이스라인 모델들의 실험 환경을 나타낸다.

[표 4-3] 실험 환경

OS	Ubuntu 16.04 LTS
CPU	Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz
Disk	SSD 2TB
RAM	192GB
GPU	Tesla V100 32GB

4.2.1 평가 지표

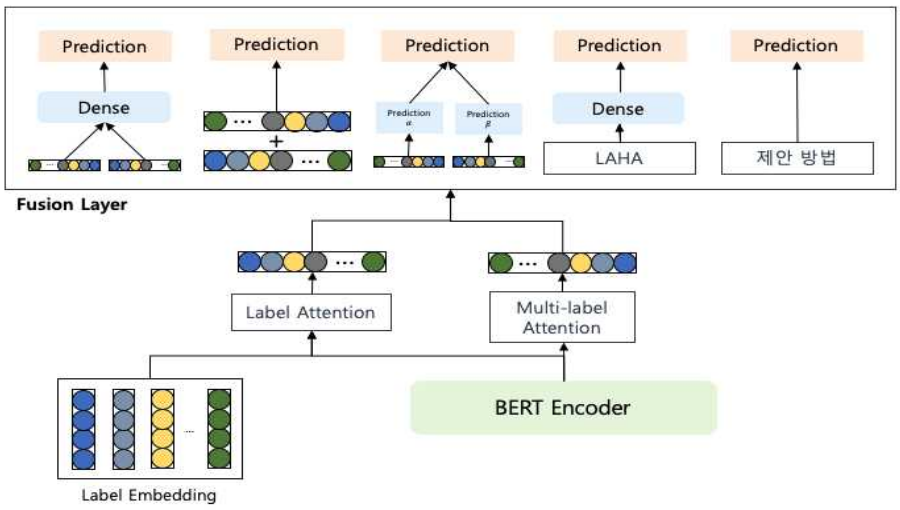
특허 분류는 하나의 데이터가 여러 개의 레이블을 가질 수 있는 다중 레이블 분류 문제이다. 이러한 다중 레이블 분류 문제에서는 랭크 기반의 평가 지표인 Precision at n(P@n)과 normalized Discounted Cumulative Gain at n(nDCG@n)이 모델 성능을 평가하는 데 주로 사용되며 각 평가 지표는 다음과 같이 정의된다.

$$P@n = \frac{1}{n} \sum_{l \in r_n(\hat{y})} y_l \qquad nDCG@n = \sum_{l=1}^{\min(n, 1^T y)} \frac{1}{\log(l+1)} \times \sum_{l=1}^{n} \frac{y_l}{\log(l+1)}$$

$P@n$에서 $y \in \{0, 1\}^k$는 각 데이터에 대한 정답 레이블, $\hat{y}$은 예측 레이블, $r_n(\hat{y})$은 예측 점수가 가장 높은 n개를 추출하였을 때의 레이블들을 의미한다. 본 논문에서는 특허의 다중 레이블 분류의 성능을 평가하기 위해 위의 두 평가 지표를 사용한다.

4.2.2 Fusion Strategy 비교

본 논문에서 제안하는 새로운 attention fusion strategy가 앞의 두 개의 층에서 출력되는 문서 표현의 장점들을 보존하며 우수한 분류 성능을 가져온다는 것을 보이기 위해 일곱 가지 상황을 비교하였다. 첫 번째와 두 번째 상황은 Fusion Layer 없이 각 층만을 이용하여 최종 레이블을 예측하는 상황이다. 세 번째와 네 번째 상황은 두 문서 표현을 단순히 연결하거나 더한 후 이를 최종 예측에 사용한 상황이며 다섯 번째 상황은 Late Fusion으로 두 출력을 통해 각각의 예측값을 구한 다음 두 예측값을 통해 최종 레이블을 예측하는 상황이다. 여섯 번째 상황은 LAHA에서 제안한 attention fusion을 그대로 사용한 상황이며, 마지막 일곱 번째 상황은 본 논문에서 제안하는 새로운 attention fusion strategy를 적용한 상황이다. [그림 4-4]는 세 번째부터 일곱 번째까지 상황에서 Fusion Layer를 나타낸다.



[그림 4-4] Fusion Layer 상황에 따른 네트워크 구조

[표 4-4] 제안 모델의 Fusion Layer 상황별 성능 비교표

Model	Evaluation Method	n = 1	n = 3	n = 5
Self Attention Layer	$P@n$	74.318%	36.658%	23.979%
	$nDCG@n$	-	78.256%	80.194%
Label Attention Layer	$P@n$	74.545%	36.731%	23.992%
	$nDCG@n$	-	78.460%	80.361%
Concatenate	$P@n$	74.223%	36.742%	24.022%
	$nDCG@n$	-	78.367%	80.281%
Sum	$P@n$	73.131%	36.491%	23.987%
	$nDCG@n$	-	77.600%	79.631%
Late Fusion	$P@n$	74.546%	36.579%	23.891%
	$nDCG@n$	-	78.216%	80.088%
Attention Fusion	$P@n$	74.626%	36.774%	24.043%
	$nDCG@n$	-	78.567%	80.483%
제안 모델	$P@n$	**75.143%**	**36.858%**	**24.078%**
	$nDCG@n$	-	**78.851%**	**80.727%**

[표 4-5]는 각 상황에 따른 성능을 나타낸 표이며 각 성능 지표별로 가장 높은 성능은 굵은 글씨로 두 번째로 높은 성능은 밑줄로 표시하였다. 실험 결과 본 연구에서 제안하는 attention fusion strategy를 사용하였을 때 모든 성능 지표에서 가장 높은 성능을 보였다. 이를 통해 본 논문에서 제안하는 attention fusion strategy를 사용하였을 때 LAHA의 Attention Fusion 보다 두 문서 표현의 장점을 잘 반영한 다중 레이블 예측이 가능하며 그 밖의 다른 fusion strategy들보다 우수한 분류 성능을 보인다는 것을 알 수 있다.

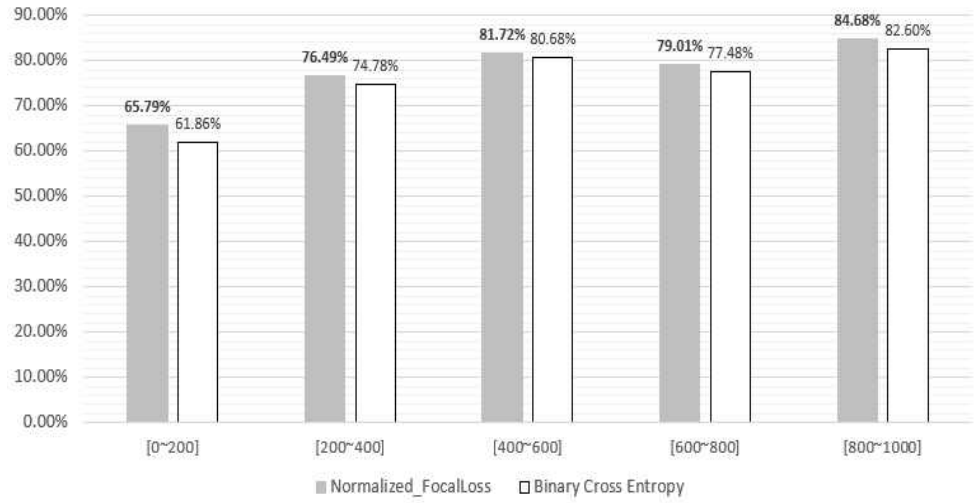
4.2.3 손실 함수 성능 비교

본 논문에서는 클래스 불균형으로 인해 발생하는 문제를 해결하고 이를 통해 성능이 향상된 분류 모델을 만들기 위해 Normalized Focal Loss 손실 함수의 사용을 제안하였다. 제안 방법을 통한 성능 변화와 클래스 불균형 문제의 극복을 확인하기 위해 일반적으로 사용되는 이진 교차 엔트로피 손실 함수와 Normalized Focal Loss 손실 함수를 이용한 실험을 진행하였다. 학습에 사용되는 손실 함수에 따른 모델의 분류 성능 결과는 [표 4-5]와 같으며 각 성능 지표별로 가장 높은 성능을 굵은 글씨로 표시하였다.

[표 4-5] 학습을 위한 손실 함수에 따른 제안 모델의 성능 비교표

Model	Evaluation Method	n = 1	n = 3	n = 5
Binary Cross Entropy	$P@n$	75.143%	36.858%	24.078%
	$nDCG@n$	-	78.851%	80.727%
Normalized Focal Loss	$P@n$	**75.538%**	**37.391%**	**24.421%**
	$nDCG@n$	-	**79.718%**	**81.605%**

실험 결과 Normalized Focal Loss를 이용하여 학습하였을 때 모든 지표에서 더 높은 성능을 보였다. 특히, $P@3$ (+0.533%), $P@5$ (+0.343%)에서 성능이 크게 향상된 것을 볼 수 있다. 이를 통해 Normalized Focal Loss 손실 함수를 이용하여 모델을 학습하는 것이 클래스 불균형만 극복하는 것뿐만 아니라 전반적인 성능 향상에도 도움이 된다는 것을 알 수 있다.



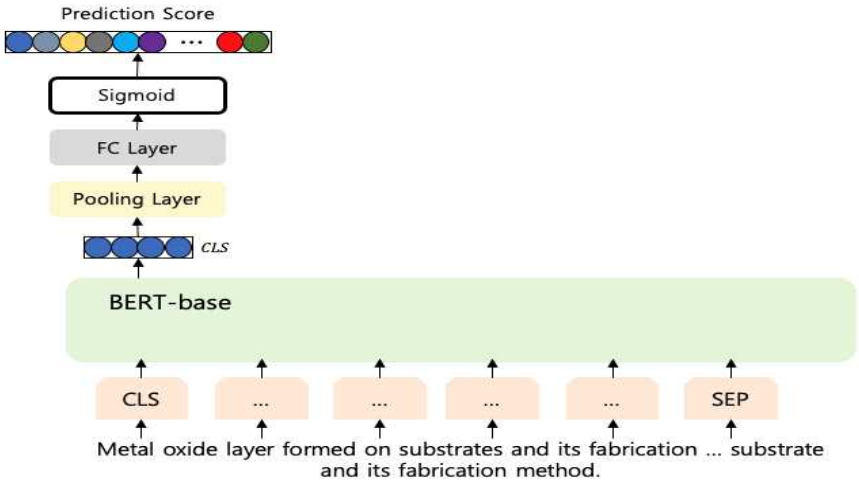
[그림 4-5] 데이터가 1,000개 미만인 클래스의 평균 분류 정확도

[그림 4-5]는 학습 데이터가 1,000개 미만인 클래스의 평균 분류 정확도를 손실 함수별로 나타낸 그림이다. $x-axis$은 학습 데이터에 포함된 데이터의 범위를 나타내며 $y-axis$은 해당 범위에 포함되는 클래스들의 평균 분류 정확도를 나타낸다. 학습 데이터가 1,000개 미만으로 상대적으로 적은 데이터를 갖는 클래스에 대한 분류 정확도가 전반적으로 향상된 것을 볼 수 있다. 이를 통해 Normalized Focal Loss를 통해 모델을 학습함으로써 클래스 불균형을 다소 극복할 수 있음을 알 수 있다.

4.2.4 비교 모델과의 성능 비교

본 논문에서 제안하는 극한 다중 레이블 분류 모델은 딥러닝 기반의 방법으로 제안 모델의 성능 비교를 위해 딥러닝 기반의 텍스트 극한 다중 레이블 분류 선행 연구인 Ronghui et al.(2019)의 AttentionXML 모델, Huang et al.(2019)의 LAHA 모델을 워드 임베딩 모델에 따라 성능을

측정한다. 또한, 대부분의 NLP Tasks에서 SOTA를 달성하였으며 최근 특허 분류 모델에서 가장 높은 성능을 보인 BERT Fine-Tuning 다중 레이블 예측 모델을 베이스라인 모델로 선정하였다. [그림 4-6]는 베이스라인 모델로 선정한 BERT Fine-Tuning 다중 레이블 예측 모델의 구조를 나타낸다.



[그림 4-6] BERT Fine-Tuning 다중 레이블 예측 모델

[표 4-6]는 베이스 라인 모델과 제안 모델의 학습 시에 사용한 최적화 함수, Learning Rate와 Learning Rate Scheduler를 나타낸다. AttentionXML과 LAHA 분류 모델의 학습을 위해서는 AttentionXML에서 사용된 최적화 함수인 DenseSparseAdam을 그대로 사용하였으며 Learning Rate Scheduler 없이 학습을 진행하였다. BERT Multi-label Classification과 제안 모델의 학습을 위해 AdamW 최적화 함수를 사용하였으며 반복 실험을 통해 가장 높은 분류 정확도를 보이는 Learning Rate를 채택하였다.

[표 4-6] 모델별 옵티마이저, 하이퍼파라미터, 스케쥴러

Model	Optimizer	Learning Rate	Scheduler
AttentionXML	$DenseSparseAdam$	$1\times e^{-3}$	-
LAHA			
BERT Classification	$AdamW$	$5\times e^{-5}$	Linear
제안 모델		$7\times e^{-5}$	

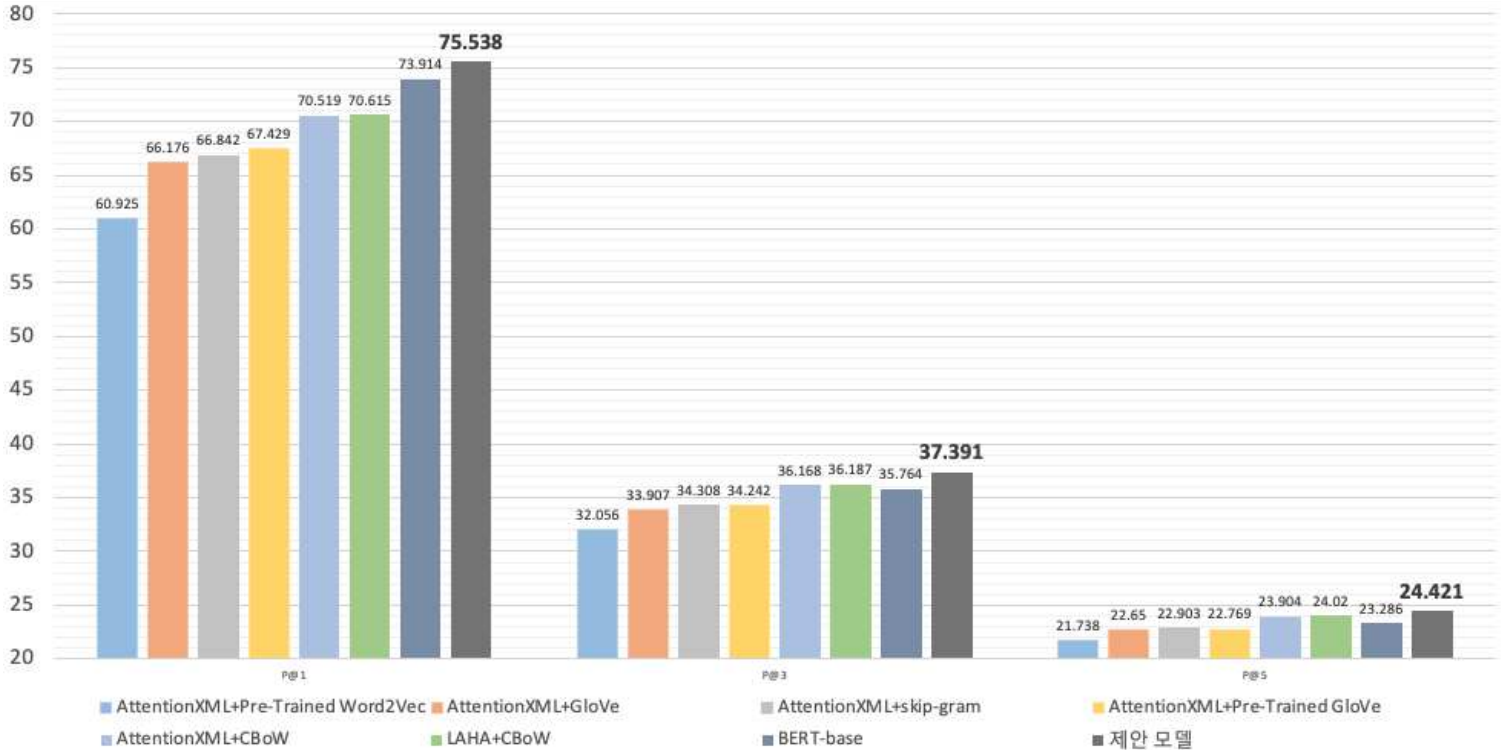
[표 4-7]와 [그림 4-7]은 각각 워드 임베딩 모델에 따른 AttentionXML 모델, AttentionXML에서 가장 높은 성능을 보인 워드 임베딩 모델을 사용한 LAHA 모델, BERT Fine-Tuning for Multi-label Classification, 제안 모델의 분류 성능을 비교한 표와 그래프이다. 각 평가 지표에서 가장 높은 성능의 값은 굵은 글씨로 표시하였으며 두 번째로 높은 성능의 값은 밑줄로 표시하였다.

실험 결과 본 논문에서 제안하는 모델이 모든 성능 지표에서 가장 높은 분류 성능을 보였다. 기존의 특허 분류에서 가장 높은 성능을 보였던 BERT Fine-Tuning 모델과 비교해 보았을 때 $P@1$에서는 1.61%의 성능 향상을 보였으며 $P@3$, $P@5$ 각각에서 1.627%, 1.135%의 성능 향상을 보였다. 또한, 기존의 다중 레이블 분류에서 SOTA를 달성한 LAHA와 비교하여 $P@n$ 성능 지표에서 각각 4.923%, 1.204%, 0.401%의 성능 향상을 보였다.

베이스라인 모델에서 BERT Fine-Tuning 모델이 하나의 레이블을 예측하는 강점을 보이지만 모델 다중 레이블 예측에서는 LAHA가 강점을 보인다. 이처럼 기존 베이스라인 모델들은 단일 레이블 예측과 다중 레이블 예측 둘 중 하나에만 강점을 보이는 반면, 본 논문에서 제안하는 모델은 단일 레이블 예측과 다중 레이블 예측 모두에서 우수함을 보이며 기존의 모델들보다 분류 정확도가 크게 향상된 것을 알 수 있다.

[표 4-7] 베이스라인 모델과 제안 모델의 성능 비교표

Model	Word Embedding	$P@n$			$nDCG@n$	
		n = 1	n = 3	n = 5	n = 3	n = 5
AttentionXML	pre-trained Word2Vec	60.925%	32.056%	21.738%	66.769%	69.721%
	GloVe	66.176%	33.907%	22.650%	71.319%	73.878%
	Skip-Gram	66.842%	34.308%	22.903%	72.193%	74.744%
	pre-trained GloVe	67.429%	34.242%	22.769%	72.293%	74.708%
	CBoW	70.519%	36.168%	23.994%	73.878%	78.652%
LAHA	CBoW	70.615%	<u>36.187%</u>	<u>24.020%</u>	76.276%	<u>78.724%</u>
BERT Fine-Tuning		<u>73.928%</u>	35.764%	23.286%	<u>76.896%</u>	78.624%
제안 모델		**75.538%**	**37.391%**	**24.421%**	**79.718%**	**81.605%**



[그림 4-7] 베이스라인 모델과 제안 모델의 성능 그래프

과적합은 모델이 학습 데이터를 과하게 학습하여 학습 데이터에 대해서는 높은 성능을 보이지만 새로운 데이터 적용 시에는 성능이 떨어지는 것을 말한다. 본 연구에서 진행한 실험에서는 과적합 되지 않은 모델을 얻기 위해 Early Stopping과 snapshot 방법을 사용하였다. [표 4-8]는 Tesla V100 GPU 1대로 모델을 학습하였을 때 모델별 학습에 걸리는 시간을 나타낸다.

[표 4-8] 모델별 학습 소요 시간 (Tesla V100)

Model	Learning Time
AttentionXML	15h 45m
LAHA	25h 37m
BERT Fine-Tuning	54h 36m
제안 모델	22h 35m

본 논문에서 제안하는 모델은 베이스라인인 다른 분류 모델들보다 높은 성능을 보일 뿐만 아니라 AttentionXML을 제외한 나머지 베이스라인 모델들보다 학습에 걸리는 시간이 크게 감소하였다. 특히 BERT Fine-Tuning 모델과 비교하였을 때 학습 시간이 절반 이상 단축되었음을 알 수 있다. 이는 1회 학습 시 소요되는 학습 시간은 비슷하지만 Best Model에 도달하기 위한 반복 횟수가 줄어든 결과이다. 이를 통해 본 논문에서 제안하는 모델은 기존의 모델들보다 높은 분류 정확도를 보일 뿐만 아니라 적은 반복 횟수로 Best Model에 도달할 수 있다는 것을 알 수 있다.

제 5 장 결 론

5.1 결론

특허 출원 수는 빠르게 증가함에 따라 특허 분류를 자동화하기 위해 많은 연구가 진행되었지만 만족할만한 성능을 보이지 못하며 특허 분류는 여전히 대부분 수작업으로 이루어지고 있다. 본 연구에서는 특허 분류를 위한 BERT 기반의 향상된 극한 다중 레이블 분류 모델을 제안하였으며, 클래스 불균형 문제를 극복하기 위해 이진 교차 엔트로피 손실 함수를 대신하여 Normalized Focal Loss 손실 함수를 통해 모델을 학습하였다. 이후, 제안 모델의 우수성을 보이기 위해 유럽, 일본에서 출원된 영문 특허 약 백만 건을 통해 데이터 셋을 구축하고 이를 이용하여 세 가지 실험을 진행하였다.

첫째, 본 논문에서 제안하는 Fusion Strategy의 우수성을 보이기 위한 실험을 진행하였다. 이를 위해 Fusion Layer를 적용하지 않은 두 가지 경우와 제안 방법을 포함한 다섯 가지 Fusion Strategy에 대한 실험을 진행하였다. 실험 결과 제안된 Fusion Strategy를 사용하였을 때 모든 성능 지표에서 가장 높은 성능을 보였다. 이를 통해 본 논문에서 제안하는 방법이 기존 방법들 보다 문서 표현의 장점을 잘 보존할 수 있으며 이를 반영한 다중 레이블 예측이 가능하다는 것을 볼 수 있다.

둘째, Normalized Focal Loss를 이용한 클래스 불균형 극복을 보이기 위한 실험을 진행하였다. 실험을 위해서 이진 교차 엔트로피 손실함수를 통해 학습한 모델과 Normalized Focal Loss를 통해 학습한 모델을 만들고 이 둘의 성능을 비교 분석하였다. 실험 결과 Normalized Focal Loss를 사용함으로서 클래스 불균형이 극복될 뿐만 아니라 전체적인 분류 정

확도가 향상된 것을 확인하였다.

셋째, 제안 모델과 베이스라인 모델들과의 비교 실험을 진행하였다. 본 연구에서는 제안 모델과 성능을 비교하기 위해 최근 특허 분류에서 우수한 성능을 보인 BERT Fine-Tuning 모델과 극한 다중 레이블 텍스트 분류에서 SOTA를 달성한 AttentionXML, LAHA를 베이스라인 모델로 채택하였다. 실험 결과 제안 모델이 단일 레이블 분류와 다중 레이블 분류 모두에서 강력함을 보이며 기존 모델들보다 높은 성능을 보이는 것을 확인하였다.

본 연구를 통하여, BERT의 모든 final hidden state를 활용하였을 때 [CLS] 토큰만을 활용하였을 때보다 시퀀스 정보를 더 잘 표현할 수 있다는 것을 확인하였으며, 특허 데이터의 특성을 반영한 문서 표현을 획득하고 이를 통한 다중 레이블 분류가 효과적임을 확인하였다. 또한, 제안한 Fusion Strategy가 기존의 방법들보다 서로 다른 장점이 있는 두 문서 표현의 장점을 보존하며 레이블을 분류할 수 있음을 확인하였으며, Normalized Focal Loss를 통해 모델을 학습함으로써 클래스 불균형 문제를 극복할 뿐만 아니라 전체적인 분류 정확도가 향상됨을 확인하였다. 마지막으로 베이스라인 모델들과의 성능 비교를 통해 본 논문에서 제안하는 모델이 특허 분류에 있어 기존 연구들보다 우수하다는 것을 보였다.

본 연구에서는 영문 특허만을 이용하여 제안 모델에 대한 실험을 진행하였다. 제안 모델이 영문 특허에서는 높은 성능을 보였지만 한글 문서에서는 그렇지 못할 가능성이 있다. 따라서 한글 특허 문서를 이용한 실험을 추가로 진행할 필요가 있다. 또한, 본 연구는 사전 학습된 BERT 기반의 모델을 Fine-Tuning 하여 특허 분류모델을 개선하였다. 사전 학습된 BERT를 Fine-Tuning 할 경우 특정 도메인에 취약하다는 단점이 있

다. 특허 분류에도 이러한 단점이 영향을 미치는지 확인하고, 이를 해결하기 위해 대용량의 특허 데이터를 기반으로 Further Training 시켜 직접 사전학습 모델을 구축하고, 이를 Feature Extraction으로 활용한 다중 레이블 특허 분류를 해보고자 한다.

참고문헌

[1] Li, Q., Maggitti, P. G., Smith, K. G., Tesluk, P. E., and Katila, R. (2013) “Top management attention to innovation: The role of search selection and intensity in new product introductions.” Academy of Management Journal, 56(3), 893 - 916.

[2] Heeyong N, Yeongran J, Sungjoo L, (2015). “Keyword selection and processing strategy for applying text mining to patent analysis”. Expert Systems with Application, 42(9), 4348-4360

[3] WIPO (2020) “World intellectual property indicators 2019”. Geneva: World Intellectual Property Orgamization

[4] 특허청 특허심사기획과, 한국특허정보원 특허정보진흥센터 분류사업팀 (2015) 팀, “IPC 가이드 & 가이드라인 해설서”

[5] Shaobo. Li, Jie. Hu, Yuxin. Cui, and Jianjun. Hu. (2018) “DeepPatent: patent classification with convolutional neural networks and word embedding”. SCIENTOMETRICS. vol 117(2), 721-744

[6] Jie Hu , Shaobo Li, Jianjun Hu, and Guanci Yang. (2018) A Hierarchical Feature Extraction Model for Multi-Label Mechanical Patent Classification. Sustainability, MDPI, Open Access Journal, vol. 10(1), 1-22

[7] WIPO (2020) “International Patent Classification: Guide”, Geneva : World Intellectual Property Organization, 2020.

[8] 통계청 국가통계포털 KOSIS (2020) 기술분야별 특허 출원 (WIPO 기술분류 기준)

[9] Fadi Thabtah, Suhel Hammound, Firuz Kamalov and Amanda Gonsalves. (2020) “Data imbalance in classification: Experimental evalution”. Information Sciences. vol 513. 429-441

[10] Jieh-Sheng Lee, and Jieh Hsiang. (2019) “PatentBERT: Patent

Classification with Fine-Tuning a pre-trained BERT Model". arXiv:1906.02124

[11] Aggarwal C.C., Zhai C. (2012) "A Survey of Text Classification Algorithms". In: Aggarwal C., Zhai C. (eds) Mining Text Data. Springer
[12] Huda Abdulrahman Almuzaini and Aqil M. Azmi. (2020)" Impact of Stemming and Word Embedding on Deep Learining-Based Arabic Text Categoritzation". IEEE Access. vol 8. 127913-127928

[13] Mikolov T, Chen K, Corrado G, Dean J. (2013). "Efficient estimation of word representations in vector space". arXiv preprint arXiv:1301.3781. 2013a

[14] Jeffrey P, Richard S and Christopher D., (2014). "GloVe: Global vectors for word representation". Processings of the 2014 Conferrence on Empirical Methods in Natural Language Processing. 1532-1543

[15] Xiang Zhang, Junbo Zhao and Yann LeCun (2015) "Character-level Convolutional Networks for Text Classification" Advances in Neural Information Processing Systems 28 (NIPS 2015)

[16] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner (1998) "Gradient-based learning applied to document recognition", Proceedings of the IEEE, 86, pp. 2278 - 2324

[17] J. Cheng, L. Dong, and M. Lapata, (2016) "Long short-term memory-networks for machine reading," arXiv preprint arXiv:1601.06733
[18] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau (2015) "A C-LSTM neural network for text classification", CoRR, abs/1511.08630

[19] D. Bahdanau, K. Cho, and Y. Bengio (2014) "Neural machine translation by jointly learning to align and translate", CoRR, abs/1409.0473.

[20] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy (2016) "Hierarchical attention networks for document classification", in HLT-NAACL

[21] C. J. Fall, A. Törcsvári, K. Benzineb, and G. Karetka (2003) “Automated categorization in the international patent classification”. SIGIR Forum 37, 1, 10 - 25.

[22] Mattyws F., Caudia A., Andreia G., (2017). “Automated Patent Classification Using Word Embedding”. 2017 16th IEEE International Conference on Machine Learing and Application(ICMLA), 408-411

[23] Jie Hu , Shaobo Li, Jianjun Hu, and Guanci Yang. (2018) “A Hierarchical Feature Extraction Model for Multi-Label Mechanical Patent Classification”. Sustainability, MDPI, Open Access Journal, vol. 10(1), 1-22

[24] Shaobo. Li, Jie. Hu, Yuxin. Cui, and Jianjun. Hu. (2018) “DeepPatent: patent classification with convolutional neural networks and word embedding”. SCIENTOMETRICS. vol 117(2), 721-744

[25] Bhatia K, Jain H, Kar P, Varma M and Jain P (2015) “Sparse local embeddings for extreme multi-label classification”. In: Proc. of NIPS. 730-738.

[26] Prabhu Y and Varma M (2014) “Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning”. In: Proc. of ACM SIGKDD, 263-272.

[27] Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamisuka and Shanfeng Zhu (2019) “AttentionXML: Label Tree-based Attention-Aware Deep Model for High-Performance Extreme Multi-Label Text Classification”. arXiv:1811.01727

[28] Xin Huang, Boli Chen, Lin Xiao and Liping Jing (2019) “Label-aware Document Representation via Hybrid Attention for Exreme Multi-Label Text Classification”. arXiv:1905.10070

[29] B. Liu, F. Sadeghi, M. Tappen, O. Shamir, and C. Liu (2013) “Probabilistic label trees for efficient large scale image classification” In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pages 843 - 850

[30] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova (2018) “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. arXiv:1810.04805

[31] Zhang W, Yan J, Wang X and Zha H (2018) “Deep extreme multi-label learning”. In: Proc. of ACM ICMR, 100-107.

[32] Grover A and Leskovec J (2016) “node2vec: Scalable feature learning for networks”. In: Proc. of ACM SIGKDD, 855-864.

[33] T. Lin, P. Goyal, R. Girshick, K. He and P. Dollar, (2017) “Focal Loss for Dense Object Detection,” in 2017 IEEE Int. Conf. on Comput. Vision (ICCV), Venice, Italy

[34] Sofiiuk, K., Barinova, O. and Konushin, A (2019) Adaptis: Adaptive instance selection network. In: Proc. IEEE Int. Conf. Comp. Vis.

[35] Oleksii Kononenko, Olga Baysal, Reid Holmes, and Michael W. Godfrey (2014) “Mining modern repositories with elasticsearch”. In Proceedings of the 11th Working Conference on Mining Software Repositories (MSR 2014). Association for Computing Mac9hinery, New York, NY, USA, 328 – 331.

[36] Yamazaki, S., Hatano, K., Katayama, M., Eguchi, S., Oikawa, Y., & Nakamura, A. (2014). U.S. Patent No. 8,766,269. Washington, DC: U.S. Patent and Trademark Office.

[37] Kohavi, R. (1995) “A study of cross-validation and bootstrap for accuracy estimation and model selection”. In: IJCAI, pp. 1137 – 1145

[38] Sechidis K., Tsoumakas G., Vlahavas I. (2011) “On the Stratification of Multi-Label Data”. In: Gunopulos D., Hofmann T., Malerba D., Vazirgiannis M. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2011. Lecture Notes in Computer Science, vol 6913. Springer, Berlin, Heidelberg.