Journal of the Korean Institute of Industrial Engineers
Vol. 51, No. 1, pp. 61-72, February 2025.
ISSN 1225-0988 | EISSN 2234-6457
https://doi.org/10.7232/JKIIE.2025.51.1.061

Bigbird-Pegasus 기반의 청구범위 생성요약을 통한 특허분류 방법론

이영재 · 김지호 · 이홍철†

고려대학교 산업경영공학과

A Methodology for Patent Classification through Bigbird-Pegasus Based Claim Abstractive Summarization

Youngjae Lee · Jiho Kim · Hongchul Lee

Department of Industrial and Management Engineering, Korea University

Patent classification is a crucial process in the examination procedure, matching the invention technology of the application with technical classification codes, and manually classifying is significant time and cost. To automate this, various machine learning-based AI methods have been researched, and recently, Transformer-based patent classification models have shown excellent performance. However, Transformer models are limited to a maximum of 512 tokens for input, there is a possibility of information loss. This study proposes a method to improve performance by using Bigbird-Pegasus and PatentSBERTa to summarize the entire text data of the claims into a fixed size before inputting it into the classification model. Experimental results show that the F1 score achieved up to 67.554% in a small-scale patent data environment, representing a 4% point performance improvement over existing methods. Additionally, this study suggests an effective patent automatic classification method through the optimal combination of summarized text and other patent items.

Keywords: Patent Classification, Patent Analysis, Deep Learning, Natural Language Processing, Abstractive Summarization

1. 서 론

최근 현대사회는 시장의 빠른 변화와 기술혁신을 위한 무한한 경쟁으로 제품의 수명주기가 짧아지는 경향을 보인다(Lee *et al.*, 2009). 이러한 상황에서 기업은 연구개발을 통한 경쟁우위 및 새로운 기술을 확보하는 것에 역량을 다하고 있다(Wang *et al.*, 2020). 국가는 특허 출원제도를 통해 기술에 대한 독점 사용권을 출원인에게 부여하여 경제적 권리를 보장하고, 발명원리를 공개하여 기술 경쟁을 통해 산업 발전을 도모하고 있다(Kasravi *et al.*, 2007). 특허는 발명기술에 대한 최신 정보를 제공하므로 데이터 분석을 통한 기술개발 추적 및 기술영역 탐색 등 다방면으로 활용되어 기술혁신을 유도한다(Lee *et al.*, 2012; Jang *et al.*, 2017). 이렇게 높아지는 특허의 중요성으로 전 세계 출원율은 계속 증가하는 경향을 보인다(WIPO, 2023). 각국의 특허청은 지식재산권에 관한 총체적 업무를 담당하며 특허의 성립 요건을 검토하는 기본 업무 과정에서 특허분류 작업을 수행한다(Korde *et al.*, 2012).

특허분류란, 출원된 특허를 구성하는 발명기술이 어느 기술에 속하는지를 미리 정의된 클래스인 기술분류체계에 할당하는 작업을 의미한다(Fall *et al.*, 2003). 이는 서로 다른 발명기술 영역을 구체적으로 식별하고 특허검색 및 동향을 분석하도록 활용될 수 있다(Kim and Bae, 2017). 현재 세계적으로 통용되

본 논문은 교육부 및 한국연구재단의 4단계 BK21 사업으로 지원된 연구임.
† 연락저자 : 이홍철 교수, 02841 서울특별시 성북구 안암로 145 고려대학교 산업경영공학부, Tel : 02-3290-3389, Fax : 02-929-5388, E-mail : hclee@korea.ac.kr
2024년 7월 4일 접수; 2024년 9월 1일 수정본 접수; 2024년 9월 30일 게재 확정.

는 대표적인 특허 기술분류체계는 '국제특허분류(IPC)', '선진특허분류(CPC)'이며 모두 최상위 단계인 Section부터 Class, Sub-class, Group, 최하위 단계인 Sub-group까지 5단계로 구성된 계층적 구조를 갖는다. Sub-group에서 IPC는 약 7만 개, CPC는 약 27만 개의 분류코드를 보유하여 보다 세분된 CPC가 산업에서의 전체 기술을 다루는 데 활용 범위가 넓은 것으로 증명되었다(Degroote *et al.*, 2018).

그러나 높아지는 특허 출원율에도 불구하고 현재 대다수의 특허분류 작업은 여전히 전담 인원이 직접 읽는 등의 수작업 방식으로 진행되고 있다(Li *et al.*, 2018). 2018년 한해에는 미국에서 등록특허가 308,853개에 달하여 미국 역사상 2번째로 많은 수치를 기록하였는데 이는 방대한 텍스트로 구성된 특허문서를 효율적으로 분류하는 과정이 매우 어려운 문제인 것으로 평가되었다(Risch and Krestel, 2019). 또한 IPC, CPC와 같은 기술분류체계는 정기적으로 개정되고 있어 그 이후에는 기존 특허문서의 일부를 다시 분류하는 작업이 뒤따르며 이 또한 수작업으로 진행된다(Held *et al.*, 2011; D'hondt *et al.*, 2013). 이는 향후 담당자의 업무 과부하로 이어져 출원 심사가 지연되거나 오분류가 발생하는 원인으로 이어질 수 있다. 따라서 이러한 문제를 방지하기 위해 자동화 특허분류의 필요성이 증대되었고 그동안 머신러닝 기반의 AI 기술을 접목한 특허분류 연구가 활발하게 수행되었다. 최근에는 자연어처리(NLP)가 트랜스포머를 바탕으로 고도화되어 특허문서에 적용한 연구가 나타나고 있다(Jang *et al.*, 2023).

대표적으로 각 트랜스포머 파생모델인 BERT, SBERT를 활용한 텍스트 입력 기반의 특허분류 연구가 연달아 높은 성능을 보이는 등의 유용성을 증명하였다(Lee and Jieh Hsiang, 2020; Bekamiri *et al.*, 2024). 그러나 다음과 같은 한계점 역시 존재하는데, 먼저 트랜스포머의 최대 입력이 512 토큰이라는 제한된 길이를 갖기에 특허의 모든 텍스트를 활용할 수 없다는 점에 있다. 이는 입력 데이터 부족으로 인한 정보 손실을 초래할 수 있으며, 특히 청구범위는 독립항과 종속항으로 구성되어 발명기술을 설명하므로 일부 내용만 입력할 경우 항목간 의미적 연관성을 고려할 수 없다(Lee *et al.*, 2013). 더하여, Lee(2020), Bekamiri(2024)와 같은 기존의 선행연구는 대규모 특허 데이터를 학습하여 높은 분류성능을 증명하였으나 반대로 소규모 데이터에 대한 workflow는 확립되지 않았다는 점에 있다. 이는 일반적으로 분류모델에 있어 소규모 데이터로 야기되는 불균형한 클래스 분포가 성능에 악영향을 끼치는 것에 의한 조치로 해석된다(Hu *et al.*, 2015). 그러나 상기 특허분류 자동화의 필요성을 기반하여 분류모델은 높은 성능과 더불어 소규모 입력 데이터를 실시간으로 처리할 수 있어야 한다.

본 연구에서 활용할 입력 데이터는 특허 청구범위로 정하였다. 이는, 청구범위가 발명기술의 범위와 경계를 정의하는 만큼 특허를 출원할 때 실무자에 있어 가장 중요하게 다뤄지는 항목이기 때문이다. 또한, 특허를 구성하는 여러 항목은 이러한 청구범위에서 확장되거나 파생된 내용으로 되어 있기에 그 중요성을 방증한다(Lee and Jieh Hsiang, 2020). 그리고 상술한 출원율이 계속 올라가는 만큼 특허분류를 포함한 효율적인 특허분석의 필요성이 더욱 요구된다. 이를 수행할 보조도구에는 특허요약이 포함되며 이는 그동안 실무자와의 협력을 통해 중요한 작업으로 확인되었다(Brügmann *et al.*, 2015; Kim and Yoon, 2022).

따라서 본 연구가 제안하는 자동화 특허분류 방법론은 다음과 같다. 우선, 딥러닝 기반의 NLP 모델로 전체 청구범위를 512 토큰 이내로 생성요약하여 도출된 결과를 다시 특허분류 모델에 입력하는 방법을 제안한다. 또한, 이렇게 요약된 청구범위를 요약항(Abstract)과 같은 기존의 다른 특허항목과의 조합으로 가장 높은 성능을 갖는 최적의 입력을 찾도록 한다. 이를 통해 소규모 데이터의 입력으로 야기되는 낮은 분류성능을 개선하는 CPC 특허 다중분류 방법론을 제안한다. 상술한 것처럼 특허 청구범위는 독립항과 종속항이 의미적 연관관계를 지닌다(Lee *et al.*, 2013). 특히 청구범위가 다른 문장보다 더욱 길고 법률 및 기술용어가 혼합되어 복잡한 구문 구조를 갖기에 해석하기 어렵다는 특징을 갖는다(Verberne *et al.*, 2010). 그러므로 생성요약을 활용하여 토큰 수에 제한받지 않음과 동시에 청구범위를 분류모델이 이해하기 수월한 문장으로 변환하고자 한다.

본 논문의 구성은 다음과 같다. 제2장에서는 전반적인 특허분류 선행연구를 정리하였다. 제3장에서는 전체 청구범위에 대한 생성요약 모델선정 기준을 포함하여 본 연구의 전반적인 프로세스를 작성하였다. 제4장에서는 생성요약 및 특허분류에 대한 실험절차 및 분석결과를 제시하며 제5장에서는 본 연구의 결론을 포함한 의의 및 한계점에 대해 논의하였다.

2. 특허분류 선행연구

2.1 머신러닝 기반의 특허분류 선행연구

그동안 자동화 특허분류 연구는 머신러닝에 기반하여 다양하게 수행되어 왔다. Chen(2012)은 SVM, K-means, KNN과 같은 기존 머신러닝 알고리즘을 포함한 3단계의 계층적 특허분류 방법론을 제안하였다. TF-IDF를 활용하여 판별력 있는 용어를 선택하고, 이를 통해 21,104개의 특허문서를 각 IPC 단계에서 분류하였다. IPC의 Class, Sub-class에서 각 65.75%, 53.25%인 최상의 정확도를 달성하였다(Chen and Yuan, 2012). 그러나 TF-IDF는 문서를 고차원의 희소 벡터로 표현하여 메모리와 같은 계산자원을 많이 소모하고 단어 간의 의미적 관계를 반영하지 못한다는 단점이 존재한다. 이와 달리 Grawe(2017)는 분산 표상 기반 모델인 Word2Vec으로 단어 임베딩을 학습하였고 여기에 LSTM을 활용하여 IPC Sub-class에서 F1-score가 최대 62%에 도달하였다(Grawe *et al.*, 2017). 그러나 LSTM은 RNN과 마찬가지로 입력 데이터를 순차적으로 처리하여 속도가 느리고 장기 의존성 문제를 완전히 해결하지는 못하였기에 특허와 같은 긴 문서에 적용하기에는 무리가 있다.

Li(2018) 또한 마찬가지로 단어 임베딩을 학습하였는데 여기에 CNN을 결합하여 특허분류 알고리즘인 DeepPatent를 제안하였다. 약 2백만 개의 특허로 구성된 특허분류 벤치마크 데이터셋 USPTO-2M으로 분석한 결과 IPC Sub-class 기준 73.88%의 정밀도를 달성하였다. DeepPatent는 Random Forest와 같은 기존 머신러닝 알고리즘보다 뛰어난 분류성능을 가지며 무엇보다, 대규모 특허분류 과업에 딥러닝 모델을 개발하고 적용한 최초의 연구라는 점을 기여로 소개하였다(Li *et al.*, 2018). 그러나 최근에는 트랜스포머의 모델성능이 CNN, RNN을 능가하는 경향을 보이며 특히, 텍스트를 다루는 것에 있어 트랜스포머 기반의 파생모델을 활용하는 것이 기존 머신러닝 접근 방법에 비해 더 나은 결과를 가져다준다는 연구가 제시되었다(Taneja and Vashishtha, 2022; Rahali and Akhloufi, 2023). 트랜스포머는 Self 어텐션을 통해 입력 데이터를 병렬로 계산하여 기존 RNN 계열 모델보다 처리 속도가 빠르다. 또한, 입력 시퀀스의 각 위치 간 관계를 모두 학습하여 문맥에 대한 높은 이해력을 지닌다는 장점이 존재한다(Vaswani *et al.*, 2017). 이에 다음 절인 2-2에 트랜스포머 기반의 특허분류 선행연구에 대해 작성하였다.

2.2 트랜스포머 기반의 특허분류 선행연구

최근에는 트랜스포머에서 파생된 모델로 특허분류 과업을 수행한 연구가 하나둘씩 나타나고 있다. Lee(2020)는 사전학습된 BERT-base에 특허 데이터를 학습한 분류모델인 PatentBERT를 제안하였다. 특히, 선행연구인 DeepPatent를 벤치마크 하여 분류성능과 분석 기법에서의 우위를 다음과 같이 2가지 실험을 통해 증명하였다. 첫 번째로, Lee(2020)는 DeepPatent와 같은 특허 데이터인 USPTO-2M의 '발명의 명칭(Title)' 및 '요약항(Abstract)'을 활용하여 비교실험을 진행하였다. 결과는 전반적인 분류성능에서 우위를 보였으며, 특히 정밀도 최고성능 값이 81.75%를 달성하여 73.88%인 DeepPatent에 비해 7.87% point의 큰 차이를 보였다. 두 번째로, Lee(2020)는 CPC Sub-class에서의 다중분류를 수행하기 위하여 '청구범위(Claim)'를 단독으로 PatentBERT에 입력하는 것만으로 최상의 결과를 달성할 수 있음을 증명하였다(Lee and Jieh Hsiang, 2020).

그리고 Bekamiri(2024)는 앞선 DeepPatent, PatentBERT를 벤치마크 하여 특허분류 모델인 PatentSBERTa를 제안하였다. Bekamiri(2024)는 상기 두 선행연구가 높은 분류성능 값을 갖으나 특허분류 과업에서 문서 간의 기술적 유사도를 측정하는 workflow가 진행되지 않은 점을 언급하였다. PatentSBERTa는 Lee(2020)와 마찬가지로 입력 데이터를 청구범위로 구성하였으며 512 토큰을 초과하지 않도록 오직 1항(대표청구범위)으로 고정하여 학습시켰다. 또한, Augmented SBERT를 활용하여 데이터 간의 의미적인 유사도를 측정한 다음 높은 결괏값을 보인 특허그룹을 도출하는 방식으로 차등을 두었으며 BERT와 같이 트랜스포머 파생모델의 높은 학습시간 등의 비용적 측면을 크게 개선하였다고 설명하였다. 유사도 측정 이후에는 KNN을 사용하여 CPC의 Class, Sub-class를 직관적으로 분류할 수 있는 hybrid 특허분류 모델을 제안하였다(Bekamiri *et al.*, 2024).

이처럼 문서 간 의미적인 유사도를 활용한 방법은 이후 새로운 유망영역을 식별하는 방법으로도 제시되었는데 예를 들어, Jeon(2023)은 텍스트 기반의 특허 간 발명기술 유사도를 측정하는 PatentSBERTa의 유용성을 언급하며 디지털 치료제와 같은 새로운 기술영역을 탐색하는 데 활용한 사례가 존재한다(Jeon *et al.*, 2023). 그러므로 본 연구에서 수행할 특허 다중분류 모델을 PatentSBERTa로 정하였다. 분류할 CPC 단계는 Section을 제외한 Class와 Sub-class로 설정하였는데 이는 Section이 분류 난이도에 있어서 상대적으로 간단하나 유용성 관점에서 그리 효과적이지 못하기 때문이다(Krestel *et al.*, 2021).

3. 프레임워크

3.1 연구 프로세스

본 연구의 전체 프로세스는 <Figure 1>에 작성하였다. Step 1을 보면 본 연구에서 활용할 특허 데이터는 선행연구인 PatentBERT, PatentSBERTa와 동일하게 PatentsView로 선정하였다(https://console.cloud.google.com/marketplace/product/google_patents_public_datasets/patentsview?project=civic-matrix-383404). Google BigQuery에서 SQL로 추출한 11,110개의 특허는 모두 미국 특허청(USPTO) 등록특허로 구성되며 변수는 'Id(특허번호)', 'Date(특허 출원일)', 'Abstract(요약항)', 'Claim(전체 청구범위)', 'CPC code'로 총 5개이다. 4-1절에 활용 데이터에 대한 상세한 설명을 작성하였다. 다음으로 Step 2를 보면 PatentsView의 Claim을 생성요약하기 위해 Bigbird(Zaheer *et al.*, 2020)의 large 버전인 NLP 모델 Bigbird-Pegasus를 선정하였다. 이때 Claim의 입력 길이 차이에 의한 요약성능과 그에 따른 분류성능을 확인하기 위해 512 토큰의 2, 4, 8배에 달하는 각 1,024, 2,048, 4,096 토큰을 최대 입력 길이로 선정하여 생성요약을 진행하였다. 여기서 Claim을 생성요약해서 도출한 결과를 'Sm_claim'이라 칭하였으며 해당 모델 선정과정을 바로 다음 절인 3-2에 상세히 작성하였다. Step 3에서는 상기 3개의 입력토큰을 구성해서 도출된 Sm_claim에 대한 요약성능을 측정하였다. Sm_claim과 비교할 참조요약을 Abstract로 설정하여 평가지표를 ROUGE로 정하였다. 여기서 Sm_claim, Abstract의 토큰 수와 생성요약 결과에 대한 상세한 설명을 4-2, 4-3에 작성하였다. 마지막으로 Step 4를 보면 특허분류 모델인 PatentSBERTa에 입력할 수 있는 모든 항목을 활용하여 가장 높은 분류성능을 갖는 항목을 찾도록 구성하였다. 여기에는 각 항목을 단독으로 입력한 것과 2개의 항목을 조합해서 입력한 것을 모두 포함한다. 특히 Sm_claim을 활용한 입력 데이터에 유의하여 특허분류에 있어 생성요약의 유용성을 확인하도록 한다. 본 연구에서는 이러한 분류결과의 상세한 내용을 4-4에 작성하였으며 여기에 제안한 특허분류 방법론이 대표청구범위만을 모델에

활용한 Lee(2020), Bekamiri(2024)와 어떠한 성능 차이를 보이는지 포함하여 서술하였다.

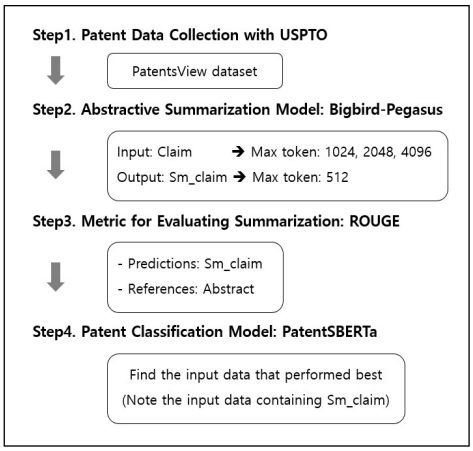

Figure 1. All Processes of Research Concept and Framework

3.2 생성요약 모델선정

앞서 트랜스포머 파생모델에 512 토큰보다 긴 텍스트를 입력하는 것은 물리적으로 불가능하기에 전체 청구범위에 적용할 요약모델을 관련 선행연구를 통해 최종 선정하도록 한다. 문서요약에는 크게 추출요약과 생성요약으로 나뉘며 특히 딥러닝 기반의 생성요약 방법론이 성능 측면에서 크게 발전되어 트랜스포머 이후로 더욱 가속화되고 있다(See *et al.*, 2017). 실제로 Gehrmann(2018)은 트랜스포머를 활용하여 CNN/Daily Mail과 같은 요약벤치마크 데이터셋에서의 성능을 개선하였다(Gehrmann *et al.*, 2018). 그러나 Sharma(2019)는 요약벤치마크 데이터셋의 상당한 경우가 도메인이 주로 신문이나 뉴스에서 가져온 것임을 언급하며 이는 특허문서와의 명백한 차이가 있음을 주장하였다. 이는 특허가 주로 문서 앞단에 요약할 가치가 있다고 판단되는 내용이 들어있는 뉴스와 달리, 문서 전체에 중요 내용이 고르게 퍼져있기에 보다 풍부한 담론 구조를 갖는다고 설명하였다. 따라서 약 130만 개의 USPTO 특허로 구성된 특허문서요약을 위한 별도의 생성요약 벤치마크 데이터셋인 BIGPATENT를 제안하였다(Sharma *et al.*, 2019). BIGPATENT는 생성요약에 있어 원본 문서인 입력과 생성한 요약문인 출력을 각 특허의 '발명의 상세한 설명(Description)'과 '요약항(Abstract)'으로 구성된다. 여기서 Description은 특허명세서에 포함된 항목으로서 Claim을 보충하고 해석하는 자료로 활용되기에 높은 중요성을 지닌다(Shinmori *et al.*, 2004).

따라서 본 연구는 BIGPATENT에서의 높은 요약성능을 Claim에 대한 생성요약 모델 선정기준으로 삼았고 해당 성능 순위는 Papers with Code(PWC) 플랫폼의 리더보드를 참고하여 확인할 수 있다. PWC는 공개된 논문에 관한 구현 코드 및 데이터를 공유할 수 있는 website로 최근 한 연구에서는 AI 연구 활동에 미치는 전반적인 영향력을 조사한 사례가 있을 정도로 활용성이 증명되었다(Kang *et al.*, 2023). 이에 높은 ROUGE 성능을 증명한 모델은 LongT5(Guo *et al.*, 2021)와 BigBird-Pegasus(Zaheer *et al.*, 2020)인 것으로 나타났으며 Hugging Face를 추가로 참고하여 최종 요약모델을 선정하도록 한다. 확인 결과, 상기 두 모델 모두 Google에서 개발하였기에 모델배포자 또한 Google이어야 하며, BIGPATENT 데이터셋으로 최종 학습된 조건을 모두 갖춘 Hugging Face 모델은 'google/bigbird-pegasus-large-bigpatent'로 나타났다(https://huggingface.co/google/bigbird-pegasus-large-bigpatent). 따라서 Claim에 대한 생성요약 모델을 Bigbird-Pegasus로 정하였고 원리에 대한 설명은 다음 절인 3-3에 작성하였다.

3.3 생성요약 모델: Bigbird-Pegasus

<Figure 2>에 생성요약 모델 Bigbird-Pegasus의 개형을 작성하였다. Bigbird는 Self 어텐션 기반의 트랜스포머와 달리 인코더를 Sparse 어텐션 기반으로 설계하여 기존 입력 길이를 트랜스포머보다 8배 긴 최대 4,096 토큰까지 확장하였다. Self 어텐션은 문장의 모든 토큰을 참조하며 이에 따른 계산 복잡도는 $O(n^2)$으로 표현된다. 이때 n은 문장길이로 n이 증가할수록 계산비용이 2차적으로 증가하여 막대한 컴퓨팅 예산 및 bottleneck이 발생할 수 있다. Zaheer(2020)는 Self 어텐션과 달리 Sparse 어텐션의 계산비용을 $O(n)$인 즉, 선형성을 갖도록 하여 완화시켰다. 따라서 Bigbird는 트랜스포머에 비해 적은 Inner-product를 갖기에 입력 문장을 더욱 길게 처리할 수 있다. Sparse 어텐션은 다음과 같은 3가지의 세부 어텐션이 서로 결합하여 작동한다. <Figure 2>에 작성된 예시문 "I always study hard English everyday"을 통해 설명하였다.

- Global 어텐션: 'I'가 Global 토큰으로 지정되면 문장의 처음과 끝에 있는 'everyday'를 포함해 모든 토큰과의 어텐션을 계산하여 전체 문장과 상호작용한다. Global 토큰은 주로 문서의 구조적 중요성에 따라 선택되며, 주로 문장의 양끝에 위치하거나 특정 부분의 제목 또는 핵심 단어가 선택된다. 이는 long range dependency를 완화하고, 문맥을 설정하여 중요한 정보를 제공하는 데 필수적인 역할을 한다.
- Window 어텐션: 'hard'가 Window 토큰으로 지정되면 'hard'의 양옆에 놓인 'study'와 'English'와의 어텐션을 계산하는데, 이는 인접한 단어 간의 의미적인 상관성이 높다는 점을 활용한 것이다. 이를 통해 문맥적으로 가까운 토큰 간의 정보 교환이 효율적으로 이루어지며, 이는 긴 문장에서 특정 부분을 이해하는 데 유용하다. 또한, 언어의 locality를 반영하여 문장의 의미를 정확히 파악할 수 있도록 돕는다.
- Random 어텐션: Random 토큰은 무작위로 다른 토큰들과의 어텐션을 계산한다. 이 방식은 인접 토큰에 의존하는 Window 어텐션과는 상반된다. 이러한 Random 어텐션은 예기치 못한 토큰 간의 관계를 학습하여 정보의 전역적이고 다방면적인 흐름을 포착하게 한다. 이를 통해 모델의 표현력을 높이는 역할을 한다.

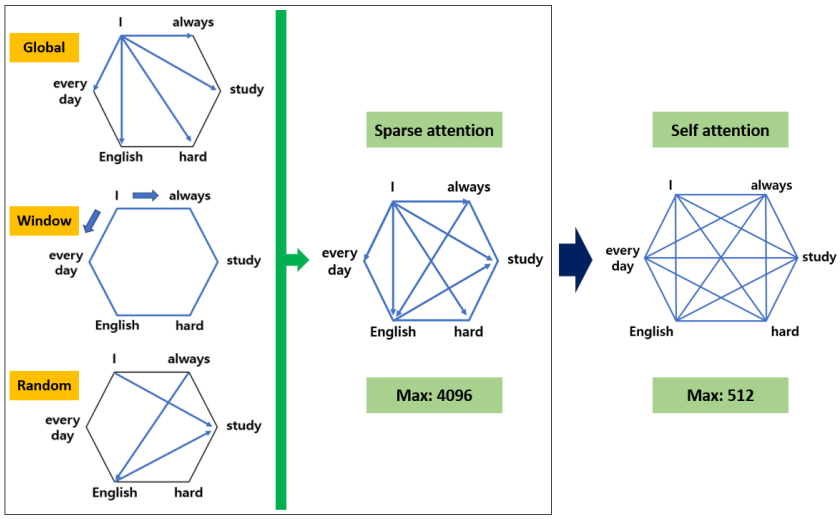

Figure 2. Abstractive Summarization Model: Bigbird-Pegasus

Bigbird는 Longformer(Beltagy *et al.*, 2020)와 마찬가지로 Warm-Starting을 활용하여 RoBERTa 모델의 가중치를 포함한 체크포인트로부터 학습된 모델이다. 이는 학습 과정에서 발생하는 시간 및 비용을 줄이기 위함이며 사전학습 역시 RoBERTa의 MLM(Masked Language Model)을 기반으로 설계되었다. 모델의 인코더에는 Sparse 어텐션을 사용하여 입력 문장길이를 최대 4,096 토큰까지 확장했지만, 디코더에는 Self 어텐션을 사용하였다. 이는 실제 요약작업에서 일반적으로 생성하는 문장길이가 입력에 비해 짧기 때문이라고 저자는 설명하였다. Bigbird는 모델 크기에 따라 base인 Bigbird-RoBERTa, large인 Bigbird-Pegasus로 구분되며 각 RoBERTa와 Pegasus (Zhang *et al.*, 2020)로부터 Warm-Starting이 이뤄진 모델이다. Pegasus는 사전학습에서 MLM이 아닌 GSG(Gap Sentences Generation)를 사용하여 요약작업에 최적화되도록 설계되었기에 Bigbird-Pegasus가 BIGPATENT와 같은 긴 문서에서 높은 요약성능을 증명할 수 있었다. MLM이 입력 텍스트에서 임의의 토큰을 가리고 해당 부분을 예측하도록 훈련되어 모델이 문맥을 이해하도록 설계되었다면, GSG는 문장에서 특정 부분을 제거하고 모델이 누락된 텍스트를 다시 생성하도록 하는 것에 차이가 있다. 따라서 GSG는 보다 긴 문장의 생성능력을 통해 문서요약과 같은 작업에 더욱 적합한 것이다. Zaheer (2020)는 입력 길이를 길게 만들수록 downstream task에 사용되는 정보가 많아지기에 특허를 포함한 긴 문서의 요약실험을 진행하여 입력 단서 증가에 따른 높은 성능을 증명하였다.

3.4 특허분류 모델: PatentSBERTa

PatentSBERTa는 Augmented SBERT로 특허 간의 기술거리를 측정하고 이후 KNN을 활용하여 CPC code를 할당한 hybrid 특허분류 모델이다. 이에 대한 개형 및 원리를 <Figure 3>에 작성하였다. Augmented SBERT는 BERT 기반의 문장 임베딩 모델

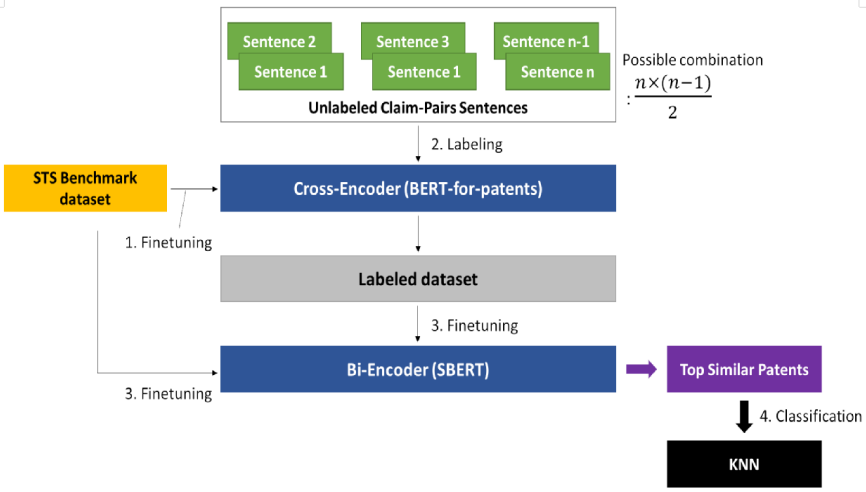

Figure 3. Patent Classification Model: PatentSBERTa

인 Sentence-BERT를 확장한 것으로, 문장 간의 유사성을 더 정확하고 효율적으로 계산하기 위해 데이터 증강기법을 활용한 모델이다(Thakur *et al.*, 2020). 이는 특허와 같은 특정 도메인에서도 높은 성능을 나타낸다. PatentSBERTa에서 Augmented SBERT와 KNN이 어떻게 작동하는지 다음과 같이 번호를 매겨 서술하였다. 해당 번호는 <Figure 3>에 작성된 번호 순서를 따른다.

1. Google에 의해 1억 개가 넘는 특허로 학습된 BERT-Large 기반 모델인 BERT-for-patents에 STS(Semantic Textual Similarity) 벤치마크 데이터셋을 미세조정 시킨다. 이리하여 문장의 의미적 유사도를 평가할 수 있다(https://github.com/google/patents-public-data/blob/master/models/BERT%20for%20Patents.md).
2. 1,143개의 청구범위 문장에서 가능한 모든 쌍을 도출하면 총 652,653개의 청구범위 조합이 계산된다. 이때, 3,343개의 문장 쌍을 sampling 한 다음 BERT-for-patents에 크로스 인코더 방식으로 labeling 수행한 것을 Labeled 데이터셋이라 지칭한다. 크로스 인코더는 각 쌍을 개별적으로 평가하여 높은 계산비용과 정확도를 갖기에 특허문서와 같이 어려운 문장에 labeling 처리할 때 효과적이다.
3. Labeled 데이터와 STS 벤치마크 데이터셋을 병합하여 SBERT에 바이 인코더 방식으로 미세조정을 수행한다. 바이 인코더는 높은 계산 효율성으로 대규모 데이터셋 처리에 적합하다. 이후 높은 의미적 유사도 값을 지닌 특허들을 도출한다.
4. 유사도 값을 기준으로 상위 특허그룹에 대해 KNN을 수행하여 k 값을 1에서 20까지 설정해 CPC의 Class, Sub-class 단계에서 다중분류를 수행한다.

4. 실험 방법 및 결과

4.1 특허 데이터 추출

본 연구에 사용한 11,110개의 PatentsView 데이터의 행렬을 <Figure 4>에 작성하였다. 행을 구성하는 각 특허는 'Patent id (특허번호)', 'Date(특허 출원일)', 'Abstract(요약항)', 'Claim (특허 청구범위)', 'CPC code'로 되어있다. 여기서 Claim은 전체 청구범위이며 추가로 본 연구는 분류모델에 청구범위 1항만 입력한 Lee(2020), Bekamiri(2024)와 비교하기 위해 SQL 문

	Claim	Patent id	Date	Abstract	CPC
1	1. A potato seed planting apparatus comprising: a frame configured to be advanced over a subjacent field to which potato seed is to be planted; at least one container on the frame for a supply of potato seed; and a plurality of laterally spaced planting units on the frame,...	10165724	2019-01-01	A potato seed planting apparatus has: a frame configured to be advanced over a subjacent field into which potato seed is planted; at least one container on the frame for a supply of potato seed; and a plurality of laterally spaced planting units on ...	A01C
2	1. A mobile machine comprising: map generator logic configured to receive a first data set and a second data set, wherein the first and second data sets comprise indications of a soil parameter of a worksite, wherein the soil parameter comprises soil temperature, and wherein the...	10165725	2019-01-01	An agricultural machine has a communication component configured to receive a first data set and a second data set. The first and second data sets comprise indications of a soil parameter of a worksite...	A01C,B64C,G05D
. . .	. . .	. . .	. . .	. . .	. . .
11110	.	.	2019-01-15	.	.

Figure 4. The Shape of Patent Data

Table 1. The Distribution of CPC Code

Section definition		Section data	Class	Sub-class
A	HUMAN NECESSITIES	2,190	15	64
B	PERFORMING OPERATIONS, TRANSPORTING	1,874	35	112
C	CHEMISTRY, METALLURGY	1,312	17	56
D	TEXTILES, PAPER	56	9	20
E	FIXED CONSTRUCTIONS	269	7	28
F	MECHANICAL ENGINEERING, LIGHTING, HEATING, WEAPONS; BLASTING ENGINES OR PUMPS	1,041	17	69
G	PHYSICS	4,226	13	57
H	ELECTRICITY	4,588	5	47
Y	NEW TECHNOLOGY DEVELOPMENTS	200	3	7
Sum	**NUMBER OF ITEMS TO CATEGORIZE**	**15,756**	**121**	**460**

을 일부 수정하여 별도로 청구범위 1항만으로 구성된 11,110개의 특허를 추출하였다. 해당 대표청구범위를 'Claim1'이라 칭하였으며 그 외 나머지 항목 및 구성은 모두 같다. 그리고 전체 Claim에 대한 생성요약 결과를 'Sm_claim'이라 표현하였으며 각 특허의 CPC code는 Sub-class 및 Class 단계에서 하나 혹은 둘 이상으로 존재하기에 본 연구는 특허 다중분류이다. 또한, <Table 1>에는 목적 데이터인 CPC code의 분포를 작성하였다. <Table 1>의 'Section data'는 총 11,110개의 각 특허에서 나타난 CPC code를 Section 기준에서 중복 여부 고려하지 않고 계산한 값이다. 그리고 <Table 1>의 'Class'와 'Sub-class'는 각 Section 기준에서 나타난 CPC code 종류의 개수를 나타낸다. 따라서 본 연구에서 분류하고자 하는 CPC code 종류의 총 개수는 Sub-class 460개, Class 121개이다. 전체적으로 특정 CPC code가 지나치게 많거나 적게 나오는 등의 모습을 보이기에 분류하고자 하는 목적 데이터가 매우 불균형한 것을 알 수 있다. 마지막으로 Date는 시작일이 2019/01/01일, 마지막 일이 2019/01/15일로 차이가 14일이라는 짧은 기간으로 확인되었다. 이는, 본 연구의 특허분류 모델 학습에 있어 여타 다른 선행연구와 달리 Date가 train, test 데이터를 구성하는 것에 영향을 주기 어렵다고 판단하였다. 예를 들어, Haghighian(2022)이 train 데이터를 2006~2014년에 출원된 특허로 분류모델을 학습시키고 test 데이터를 2015년에 출원된 특허로 평가한 것을 하나의 사례로 들 수 있다(Haghighian *et al.*, 2022).

4.2 특허 데이터의 토큰 수 비교

본 연구는 각 Abstract와 Claim1, Claim, 그리고 생성요약한 Claim인 Sm_claim의 토큰 수를 모두 포함하여 비교한 결과를 <Table 2>에 정리하였다. 요약모델에 입력할 Claim의 최대 토큰을 각 1,024, 2,048, 4,096으로 설정하였을 때의 출력물인 Sm_claim 토큰 수는 서로 차이가 적은 것으로 나타났다. 이는 향후 특허분류 모델인 PatentSBERTa에 입력하기 위해 최대 출력 토큰 수를 512로 고정한 것에 반해 모델이 출력할 최소 토큰 수를 따로 정해주지 않았기 때문으로 해석된다. <Table 2>에는 가장 긴 토큰 수를 'Top 1'로 지칭하였고 'Top 10,000'은 상위 10,000개의 토큰 수 평균이며 'ALL'은 전체 11,110개 데이터의 토큰 수 평균으로 표현하였다. 토큰 수 관점에서 Abstract의 경우 최대 787을 갖으나 대부분 512보다, 더욱 짧은 것으로 나타났으며 대표청구항인 Claim 1 또한 Abstract보다는 길지만 대부분 512보다 짧은 결과를 보인다. 이에 비해 Claim은 최대 11,497이며 데이터 전체의 평균이 1,123으로 나타나 트랜스포머 최대 입력 길이인 512의 2배가 넘는 길이를 가지는 것으로 나타났다. 생성요약에 대한 성능평가는 바로 다음 절인 4-3에 작성하였다.

4.3 청구범위 생성요약 성능평가

본 연구에서 사용할 생성요약 성능평가 지표는 Lin(2004)의 ROUGE(Recall-Oriented Understudy for Gisting Evaluation)로 정하였다. ROUGE는 언어모델이 생성한 요약문과 사람에 의해 작성된 정답 요약문을 비교하여 요약성능을 측정하는 방식으로 본 연구에서는 n-gram 기반으로 유사도를 구하는 ROUGE-N을 사용하였다. ROUGE-N은 다음에 정의한 식 (1)을 사용하여 계산된다.

$$\mathrm{ROUGE-N}=\frac{\sum_{S\in\{\mathrm{Reference\,Summaries}\}}\sum_{\mathrm{gram}_n\in S}\mathrm{Count}_{\mathrm{match}}(\mathrm{gram}_n)}{\sum_{S\in\{\mathrm{Reference\,Summaries}\}}\sum_{\mathrm{gram}_n\in S}\mathrm{Count}(\mathrm{gram}_n)} \quad (1)$$

ROUGE-N의 N은 n-gram의 길이를 의미하며 본 연구에서의 n-gram은 1, 2, L을 사용하였다. 여기서, ROUGE-1은 unigram(하나의 단어), ROUGE-2는 bigram(단어 두 개의 연속된 조합)으로 계산된다. 예를 들어, 'The cat sit on the bench'라는 문장이 있을 때, ROUGE-2의 경우 'The cat', 'sit on' 등이 모두 2-gram이 된다는 것으로 설명할 수 있다. ROUGE-L은 LCS(Longest Common Sequence)를 의미하는데 이는 최대 길이로 겹치는 열값을 계산하여 해당 문자열의 연속 여부와 관계없이 의미 있는 정보가 포함되었는지를 판단할 수 있게 한다.

먼저, 수식에 작성된 Reference Summaries는 사람에 의해 작성된 복수의 정답 요약문으로 본 연구에서는 특허의 요약항(Abstract)이 이에 해당하며 수식에서의 S는 Reference Summaries에 속한 하나의 정답 요약문을 의미한다. 그리고 언

Table 2. The Number of Token for Each Patent Data Item

		Average number of tokens					
		Top 1	Top 10	Top 100	Top 1,000	Top 10,000	ALL
Abstract		787	405	285	202	136	127
Claim 1		1,701	1,335	781	455	208	191
Claim		11,497	8,158	4,425	2,546	1,210	1,123
Max input token: 1,024 token	**Sm_claim**	512	461	314	230	112	104
Max input token: 2,048 token		512	465	330	234	112	104
Max input token: 4,096 token		512	444	317	224	111	103

Table 3. Configured a Abstractive Summarization Performance Results Comparison

Data	Model	R-1	R-2	R-L
BigPatent	TextRank(Mihalcea and Tarau, 2004)	35.99	11.14	29.60
	LexRank(Erkan and Radev, 2004)	35.57	10.47	29.03
	SumBasic(Nenkova and Vanderwende, 2005)	27.44	7.08	23.66
	RNN-ext RL(Chen and Bansal, 2018)	34.63	10.62	29.43
	LSTM seq2seq(Sutskever *et al.*, 2014) + attention	28.74	7.87	24.66
	Pointer-Generator(See *et al.*, 2017)	30.59	10.01	25.65
	Pointer-Generator + coverage(See *et al.*, 2017)	33.14	11.63	28.55
	SentRewriting(Chen and Bansal, 2018)	37.12	11.87	32.45
	TLM(Pilault *et al.*, 2020)	36.41	11.38	30.88
	TLM + Extracted sentences	38.65	12.31	34.09
	CTRL*sum*(He *et al.*, 2020)	45.80	18.68	39.06
	Pegasus *base*(Zhang *et al.*, 2020)(no pretraining)	42.98	20.51	31.87
	Pegasus *base*	43.55	20.43	31.80
	Pegasus *large*(C4)	53.63	33.16	42.25
	Pegasus *large*(HugeNews)	53.41	32.89	42.07
	BIGBIRD-RoBERTa(base, MLM)(Zaheer *et al.*, 2020)	55.69	37.27	45.56
	BIGBIRD-Pegasus(large, Pegasus pretrain)	60.64	42.46	50.01
Our data	**BIGBIRD-Pegasus(large, Pegasus pretrain)(only inference)**	51.46	34.18	41.43

어모델이 생성한 요약문은 생성요약된 청구범위(Sm_claim)로 정하였다. ROUGE 수식에서의 분모는 하나의 정답 요약문(S)에 등장하는 모든 n-gram의 빈도(Count)를 구하고 이를 모든 정답 요약문(Reference Summaries, 모든 S)에서 더한 값으로 구성된다. 즉, 분모는 모든 정답 요약문 내에서 등장한 모든 n-gram 빈도의 총합이다. 반면에 분자는 하나의 정답 요약문(S)과 언어모델이 생성한 요약문이 서로 일치하는 n-gram의 빈도(Count_match)를 구하고 이를 모든 정답 요약문(Reference Summaries, 모든 S)에서의 총합으로 구성된다. 이처럼 ROUGE는 두 텍스트 간의 n-gram 유사도를 측정하여 요약성능을 평가하는 대표적인 지표로 활용되며 n-gram 길이에 따라 성능 차이가 발생할 수 있다(Lin, 2004).

Casola(2022)는 다양한 선행연구에서의 BIGPATENT에 대한 요약결과를 ROUGE로 측정하여 표를 작성하였는데(Casola *et al.*, 2022) 본 연구는 해당 표 하단에, Claim에 대한 생성요약 결과를 추가하여 <Table 3>을 만들었다. 여기서 <Table 3>의 Our data는 11,110개 특허이며 본 연구의 생성요약 방법은 Our data에 추가학습을 통한 가중치 수정하지 않고 Inference 단계로 넘어가도록 하였다. 이는 학습비용문제로 인한 실험환경의 현실적 제약이 특허와 같은 긴 문서에 대한 요약작업에서 주로 나타나기 때문에 취한 조치이다(Huang *et al.*, 2021). Our data에 대한 ROUGE-1, 2, L의 측정값은 요약모델에 입력한 최대 토큰인 각 1,024, 2,048, 4,096일 때의 ROUGE 결과에 대한 평균값으로 51.65, 34.14, 41.52의 값을 가진다. 토큰이 1,024일 때는 51.35, 34.12, 41.35이며 2,048일 때는 51.38, 34.17, 41.41 그리고 4,096일 때는 각 51.64, 34.24, 41.52의 성능 값을 갖기에 서로의 성능 차이는 작다고 할 수 있다. 본 연구의 요약성능은 Casola(2022)가 작성한 17가지 case를 포함했을 때 각 ROUGE-1, 2, L이 5위, 3위, 5위를 차지하는 것으로 확인되었다. 이는 생성요약 결과가 이후 특허분류 모델에 입력할 수 있을 만한 성능이 나온 것으로 해석하였다. 특허분류에 대한 성능평가는 다음절인 4-4에 작성하였다.

4.4 특허분류 성능평가

특허분류 성능평가는 <Table 4>에 작성한 Confusion Matrix의 정밀도(Precision), 재현율(Recall), F1 점수(F1 score)를 참고하였으며 이들은 머신러닝 기반 AI 분류성능을 측정하는 대표적인 성능지표이다. 각 지표는 순서대로 다음에 정의한 식 (2) ~ (4)를 사용하여 계산된다.

Table 4. Confusion Matrix

Confusion Matrix		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$F1-Score=2\times\frac{Precision\times Recall}{Precision+Recall} \quad (4)$$

특허분류 모델은 PatentSBERTa로 선정하였으며 train : test 비율은 9:1로 설정하였는데 이는 Bekamiri(2024)가 test 비율을 전체 데이터의 8%로 설정한 것에 기인한다. 본 연구에서의 입력 데이터는 'Prior', 'Input data1', 'Input data2'인 총 3가지 유형으로 구분된다. 먼저, 'Prior'는 데이터를 대표청구범위인 Claim1로 구성하였으며 이는 선행연구 저자 Lee(2020), Bekamiri(2024)와 같은 방식을 채택하기 위함이다. 'Input data1'은 Abstract, Claim, Sm_Claim으로 구성되며 특히, 생성요약된 청구범위인 Sm_Claim이 Claim1과 비교해서 어느 정도의 효과를 보이는지 분석하기 위함이다. 마지막으로 'Input data2'는 'Input data1'의 각 항목 중 2개를 하나의 길이로 연결하여 구성하였다. 분류모델인 PatentSBERTa는 트랜스포머에 기반하므로 입력 제한 길이를 초과하면 내용이 잘려서 들어가기에 입력조합을 구성하는 것에 있어 긴 길이를 가진 Claim을 나중으로 연결하였다. 예를 들어, Abstract와 Claim을 조합하고자 할 때 'Abstract + Claim'은 가능하지만, 역순은 제외하였다는 것을 의미한다. 이는 최종 분류결과가 작성된 <Table 5>, <Table 6>에 그대로 반영하여 작성하였다. 추가로 과적합을 방지하기 위해 K-fold 교차검증을 사용하였고 k는 10으로 설정하였다. 이러한 과정을 통해 입력 유형인 Prior, Input data1, Input data2 간의 성능 차이를 확인하고 어느 입력 데이터가 가장 높은 분류성능을 갖는지 확인하도록 구성하였다.

<Table 5>는 특허분류 결과의 평균값으로 구성하였다. K-fold 교차검증을 적용하기 이전과 비교해보면 분류성능의 관점에서 전체적으로 소폭 증가하거나 감소하는 결과를 보이며 차이는 적은 것으로 나타났다. 이는 모델이 train 데이터에 대해서 과적합 되지 않고 잘 일반화되고 있는 것으로 판단된다. 우선 Prior와 Input data1을 보면, Abstract는 전체적으로 성능 측면에서 가장 낮으며 Claim1과 Sm_claim는 서로 간의 성능 차이가 작고 오히려 전체 청구범위인 Claim이 가장 높은 분류성능을 갖는 것으로 드러났다. 그러나, 두 항목을 연결하여 입력한 Input data2가 Prior와 Input data1를 상회하는 결과가 나타났는데 이는 입력토큰 수 관점에서 더 많은 정보량을 갖고 있기 때문이라 판단하였다. 추가로 Input data2에서 주목해야 할 점은 같은 항목 간의 데이터 조합이라도 병합 순서에 따라 분류성능 값이 달라질 수 있다는 점이다. 대표적으로 '(2048) Sm_claim + Abstract'와 'Abstract + (2048) Sm_claim'의 성능 차이를 예시로 들 수 있다. 이는 두 항목의 병합 순서에 따라 전체 문장의미 및 해석에서 차이가 발생한 경우 Augmented SBERT를 통과하여 도출되는 유사도 값 또한 서로 차이가 발생할 수 있는 것으로 해석하였다. 따라서 그다음 과정인 특허분류에서도 성능 차이가 발생했던 것으로 판단된다. 실험결과 Prior, Input data1, Input data2 중에서 가장 높은 성능 값을 갖는 최적의 입력조합은 '(1024) Sm_claim + Abstract', '(4096) Sm_claim + Abstract'로 나타났다. 이는 Sm_claim을 활용한 결과가 높은 분류성능을 갖기에 생성요약된 청구범위인 Sm_claim을 분류모델의 입력 데이터 항목으로서 고려할 가치를 증명한 결과라 해석하였다. 마찬가지로 특허분류 결과의 최댓값으로 구성한 <Table 6> 또한 K-fold 교차검증을 적용하기 이전과 비교하면 전체적으로 성능이 소폭 변동되었기에 <Table 5>와 비슷한 양상을 보인다. 여기서도 Input data2에서 높은 성능을 보이며 특히, Sm_claim을 활용한 입력조합인 '(2048) Sm_claim + Abstract', '(4096) Sm_claim +

Table 5. Patent Classification Results: Mean Value

Type	Mean	Class			Sub-class		
		F1	P	R	F1	P	R
Prior	Claim1	57.741	67.873	55.251	46.581	55.127	45.259
Input data1	Abstract	57.645	67.730	55.197	46.127	54.789	44.691
	(1024) Sm_claim	57.952	68.077	55.433	46.351	55.012	45.009
	(2048) Sm_claim	57.937	67.841	55.480	46.323	55.092	44.968
	(4096) Sm_claim	58.114	68.055	55.647	46.625	55.051	45.234
	Claim	60.220	70.597	57.649	49.348	58.508	47.777
Input data2	Abstract + Claim	60.539	70.796	58.029	49.594	58.778	48.067
	Abstract + (1024) Sm_claim	59.881	70.093	57.397	48.514	57.601	46.941
	Abstract + (2048) Sm_claim	59.535	69.659	57.085	48.362	57.404	46.806
	Abstract + (4096) Sm_claim	59.607	69.764	57.132	48.504	57.586	46.961
	(1024) Sm_claim + Abstract	**62.212**	**72.173**	**58.935**	50.746	59.814	49.272
	(2048) Sm_claim + Abstract	61.438	71.680	57.950	50.669	59.761	49.121
	(4096) Sm_claim + Abstract	61.491	70.923	58.020	**51.624**	**60.562**	**49.671**
	(1024) Sm_claim + Claim	61.263	71.645	58.684	50.379	59.358	48.798
	(2048) Sm_claim + Claim	61.320	71.558	58.780	50.519	59.471	49.290
	(4096) Sm_claim + Claim	61.307	71.571	58.761	50.472	59.425	48.896

Table 6. Patent Classification Results: Max Value

Type	Max	Class			Sub-class		
		F1	P	R	F1	P	R
Prior	Claim1	63.412	80.810	64.314	53.301	69.402	52.728
Input data1	Abstract	63.632	81.780	63.736	52.652	70.709	52.493
	(1024) Sm_claim	63.942	81.471	63.837	53.405	70.382	53.569
	(2048) Sm_claim	63.947	81.908	64.405	52.442	69.609	53.351
	(4096) Sm_claim	64.283	81.789	64.461	53.078	70.128	53.955
	Claim	66.197	84.316	66.118	55.283	72.801	56.086
Input data2	Abstract + Claim	66.717	83.584	67.147	56.384	73.817	56.387
	Abstract + (1024) Sm_claim	66.098	83.034	66.952	54.737	72.800	55.898
	Abstract + (2048) Sm_claim	66.118	83.005	66.614	54.716	72.209	55.071
	Abstract + (4096) Sm_claim	65.643	82.917	66.404	54.666	73.261	55.553
	(1024) Sm_claim + Abstract	66.237	83.607	67.163	55.798	73.797	56.861
	(2048) Sm_claim + Abstract	**67.554**	**84.918**	**67.706**	55.625	73.891	55.837
	(4096) Sm_claim + Abstract	66.500	83.390	66.531	**56.676**	**75.540**	**56.905**
	(1024) Sm_claim + Claim	66.098	84.164	66.607	55.878	73.062	56.281
	(2048) Sm_claim + Claim	66.504	83.601	66.951	55.283	72.801	56.086
	(4096) Sm_claim + Claim	66.750	83.661	66.742	55.863	74.299	56.007

Abstract'에서 최적의 값을 보였다. 전체 청구범위를 생성요약하고 이를 다시 분류모델에 입력한다는 본 연구의 유용성을 증명하는 결과라 해석하였다.

7. 결 론

본 연구는 Bigbird-Pegasus 모델을 통해 Claim을 생성요약 하여 도출된 결과를 PatentSBERTa 특허분류 모델에 입력하여 개선된 F1-score, Precision, Recall 성능 값을 도출하였다. 이는 트랜스포머 파생모델의 특성상 입력토큰이 최대 512로 제한되었기에 데이터 누락에 따른 정보손실을 방지하고 Claim 간의 의미적 관계를 고려해서 생성요약을 제안한 것이다. 다음으로 Claim1만으로 입력 데이터를 구성한 PatentSBERTa 저자와의 분류성능 비교를 수행하였으며, 특허항목 데이터 간의 다양한 조합으로 가장 높은 성능을 제시하는 최적의 입력 데이터 간의 조합을 확인하였다. 활용 데이터는 PatentsView 데이터셋으로 Google BigQuery에서 SQL 문을 사용하여 추출하였다.

본 연구는 기존 특허분류 선행연구와 비교했을 때 상대적으로 소규모인 11,110개의 데이터를 활용하였다. 여기서 일반적으로 머신러닝 기반의 분류과업에서 성능을 개선하기 위해 주로 쓰이는 방법인 일부 클래스를 제거하거나 데이터 증강하여 불균형한 클래스 분포를 균형 있게 맞추는 작업을 본 연구에서는 수행하지 아니하였다. 이는 PatentSBERTa 저자의 사례와 마찬가지로 본 특허 데이터의 CPC 분포가 매우 불균형하게 이루어져 있기 때문이며 실제 현장업무와의 괴리를 줄이기 위함이다. 본 연구는 이러한 소규모의 데이터와 불균형한 클래스로 야기되는 성능 하락과 같은 상황에서 생성요약이라는 방법을 분류과업에 추가함으로써 성능을 개선하는 방법론을 제안하였다. 이는 실제 특허분류 업무에 있어 향상된 성과를 기대할 수 있다고 판단하였다.

또한, 본 연구에서 사용한 Bigbird-Pegasus와 같은 생성형 AI 모델을 활용할 때 학습된 모델을 그대로 추론하여 활용하는 것만으로, 특허와 같은 복잡한 텍스트 데이터를 요약하는 데 좋은 성능을 나타낼 수 있었다. 이는 높은 학습비용을 줄일 수 있는 현실적 대안으로 제시될 수 있을 것으로 기대된다. 마지막으로 분류모델의 입력 데이터를 512 토큰 이내에서 Sm_claim과 결합할 수 있는 다른 특허항목과의 가능한 조합을 통해 해당 상황에서 가장 높은 성능을 보이는 최적의 입력 데이터를 탐색하였다. 이를 통해 특허분류 과업에 있어 생성요약 방법론의 활용성을 기대할 수 있다.

다만, 본 연구에는 다음과 같은 한계점이 존재한다. 첫째, 유한한 컴퓨팅 예산을 고려하여 오픈소스에서 학습된 모델을 그대로 추론하는 것과 모델 크기를 경량화해서 학습시켰을 경우와의 비교실험이 수행되지 아니하였다. 또한, 분석할 데이터 개수를 증가 혹은 감소시켰을 때의 분류성능 변화를 측정하지 않았다는 점이라 할 수 있다. 그리고 분류과업에 대한 본 연구의 접근이 특허 이외에 다양한 도메인에서 유용성을 증명하지 못한 점에 있다. 마지막으로 독립항과 종속항으로 구성된 청구범위의 계층적 구조를 살려 특허분류 모델에 활용하지 아니하였다. 이러한 사항들을 극복하려면 다양한 도메인 혹은 추가적인 특허 데이터를 활용한 비교실험을 통해 검증된 일반화 성능을 도출하고 정보 수집을 위해 다양한 모델에 의한 추가적인 방법론을 고안해야 한다. 또한, 대표청구범위 혹은 생성

요약된 청구범위에 종속항을 1개씩 추가하면서 분류성능을 측정하는 연구를 추후 고려해야 할 필요가 있다.

참고문헌

Bekamiri, H., Hain, D. S., and Jurowetzki, R. (2024), Patentsberta: A deep nlp based hybrid model for patent distance and classification using augmented sbert, *Technological Forecasting and Social Change*, **206**, 123536.

Beltagy, I., Peters, M. E., and Cohan, A. (2020), Longformer: The long-document transformer, arXiv preprint arXiv:2004.05150.

Brügmann, S., Bouayad-Agha, N., Burga, A., Carrascosa, S., Ciaramella, A., Ciaramella, M., Codina-Filba, J., Escorsa, E., Judea, A., Mille, S., Müller, A., Saggion, H., Ziering, P., Schütze, H., and Wanner, L. (2015), Towards content-oriented patent document processing: Intelligent patent analysis and summarization, *World Patent Information*, **40**, 30-42.

Casola, S. and Lavelli, A. (2022), Summarization, simplification, and generation: The case of patents, *Expert Systems with Applications*, *205*, 117627.

Chen, Y. L. and Chang, Y. C. (2012), A three-phase method for patent classification, *Information Processing & Management*, **48**(6), 1017-1030.

Degroote, B. and Held, P. (2018), Analysis of the patent documentation coverage of the CPC in comparison with the IPC with a focus on Asian documentation, *World Patent Information*, **54**, S78-S84.

D'hondt, E., Verberne, S., Koster, C., and Boves, L. (2013), Text representations for patent classification, *Computational Linguistics*, **39**(3), 755-775.

Fall, C. J., Törcsvári, A., Benzineb, K., and Karetka, G. (2003, April), Automated categorization in the international patent classification, In *Acm Sigir Forum* (Vol. 37, No. 1, pp. 10-25), New York, NY, USA: ACM.

Gehrmann, S., Deng, Y., and Rush, A. M. (2018), Bottom-up abstractive summarization, arXiv preprint arXiv:1808.10792.

Grawe, M. F., Martins, C. A., and Bonfante, A. G. (2017, December), Automated patent classification using word embedding, In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, pp. 408-411.

Guo, M., Ainslie, J., Uthus, D., Ontanon, S., Ni, J., Sung, Y. H., and Yang, Y. (2021), LongT5: Efficient text-to-text transformer for long sequences, arXiv preprint arXiv:2112.07916.

Haghighian Roudsari, A., Afshar, J., Lee, W., and Lee, S. (2022), PatentNet: Multi-label classification of patent documents using deep learning based language understanding, *Scientometrics*, **127**(1), 207-231.

Held, P., Schellner, I., and Ota, R. (2011, April), Understanding the world's major patent classification schemes, In *PIUG 2011 Annual Conference Workshop*, Vienna (Vol. 13).

Hu, Y., Guo, D., Fan, Z., Dong, C., Huang, Q., Xie, S., Liu, G., Tan, J., Li, B., and Xie, Q. (2015), An improved algorithm for imbalanced data and small sample size classification, *Journal of Data Analysis and Information Processing*, **3**(03), 27.

Huang, L., Cao, S., Parulian, N., Ji, H., and Wang, L. (2021), Efficient attentions for long document summarization, arXiv preprint arXiv:2104.02112.

Jang, H., Kim, S., and Yoon, B. (2023), An eXplainable AI (XAI) model for text-based patent novelty analysis, *Expert Systems with Applications*, **231**, 120839.

Jang, H., Roh, T., and Yoon, B. (2017), User needs-based technology opportunities in heterogeneous fields using opinion mining and patent analysis, *Journal of Korean Institute of Industrial Engineers*, **43**(1), 39-48.

Jeon, E., Yoon, N., and Sohn, S. Y. (2023), Exploring new digital therapeutics technologies for psychiatric disorders using BERTopic and PatentSBERTa, *Technological Forecasting and Social Change*, **186**, 122130.

Kang, D., Kang, T., and Jang, J. (2023), Papers with code or without code? Impact of GitHub repository usability on the diffusion of machine learning research, *Information Processing & Management*, **60**(6), 103477.

Kasravi, K. and Risov, M. (2007, January), Patent Mining-Discover y of Business Value from Patent Repositor ies, In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07),* IEEE, pp. 54-54.

Kim, G. and Bae, J. (2017), A novel approach to forecast promising technology through patent analysis, *Technological Forecasting and Social Change*, **117**, 228-237.

Kim, S. and Yoon, B. (2022), Multi-document summarization for patent documents based on generative adversarial network, *Expert Systems with Applications*, **207**, 117983.

Korde, V. and Mahender, C. N. (2012), Text classification and classifiers: A survey, *International Journal of Artificial Intelligence & Applications*, **3**(2), 85.

Krestel, R., Chikkamath, R., Hewel, C., and Risch, J. (2021), A survey on deep learning for patent analysis, *World Patent Information*, **65**, 102035.

Lee, C., Song, B., and Park, Y. (2013), How to assess patent infringement risks: A semantic patent claim analysis using dependency relationships, *Technology Analysis & Strategic Management*, **25**(1), 23-38.

Lee, J. S. and Hsiang, J. (2020), Patent classification by fine-tuning BERT language model, *World Patent Information*, **61**, 101965.

Lee, S., Lee, H. J., and Yoon, B. (2012), Modeling and analyzing technology innovation in the energy sector: Patent-based HMM approach, *Computers & Industrial Engineering*, **63**(3), 564-577.

Lee, S., Yoon, B., Lee, C., and Park, J. (2009), Business planning based on technological capabilities: Patent analysis for technology-driven roadmapping, *Technological Forecasting and Social Change*, **76**(6), 769-786.

Li, S., Hu, J., Cui, Y., and Hu, J. (2018), DeepPatent: Patent classification with convolutional neural networks and word embedding, *Scientometrics*, 117(2), 721-744.

Lin, C. Y. (2004, July), Rouge: A package for automatic evaluation of summaries, In *Text summarization branches out*, pp. 74-81.

Rahali, A. and Akhloufi, M. A. (2023), End-to-end transformer-based models in textual-based NLP, *AI*, **4**(1), 54-110.

Risch, J. and Krestel, R. (2019), Domain-specific word embeddings for patent classification, *Data Technologies and Applications*, **53**(1), 108-122.

See, A., Liu, P. J., and Manning, C. D. (2017), Get to the point: Summarization with pointer-generator networks, arXiv preprint arXiv:1704.04368.

Sharma, E., Li, C., and Wang, L. (2019), BIGPATENT: A large-scale dataset for abstractive and coherent summarization, arXiv preprint arXiv:1906.03741.

Shinmori, A., Okumura, M., and Marukawa, Y. (2004), Aligning Patent Claims with Detailed Descriptions for Readability, In NTCIR.

Taneja, K. and Vashishtha, J. (2022, March), Comparison of transfer learning and traditional machine learning approach for text classification, In *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, pp. 195-200.

Thakur, N., Reimers, N., Daxenberger, J., and Gurevych, I. (2020), Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks, arXiv preprint arXiv:2010.08240.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017), Attention is all you need, *Advances in Neural Information Processing Systems*, 30.

Verberne, S., D'hondt, E. K. L., Oostdijk, N. H. J., and Koster, C. H. (2010), Quantifying the challenges in parsing patent claims, *1st International Workshop on Advances in Patent Information Retrieval(AsPIRe'10)*, Association for Computing Machinery, NY, USA.

Wang, L., Luo, G. L., Sari, A., and Shao, X. F. (2020), What nurtures fourth industrial revolution? An investigation of economic and social determinants of technological innovation in advanced economies, *Technological Forecasting and Social Change*, **161**, 120305.

WIPO (2023), World intellectual property indicators 2023, World Intellectual Property Organization.

Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., and Ahmed, A. (2020), Big bird: Transformers for longer sequences, *Advances in Neural Information Processing Systems*, **33**, 17283- 17297.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. (2020, November), Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, In *International Conference on Machine Learning*, PMLR, pp. 11328-11339.

저자소개

이영재 : 청주대학교 공과대학 소프트웨어융합학부 빅데이터통계학과에서 2023년 학사학위를 취득하고 고려대학교 공과대학 산업경영공학과 석사과정에 재학 중이다. 연구분야는 AI, 자연어처리, 특허분석이다.

김지호 : 서울과학기술대학교 기술경영융합대학 글로벌융합산업공학과에서 2015년 학사학위를 취득하고 고려대학교 공과대학 산업경영공학과에서 2024년 박사학위를 취득하였다. 현재 고려대학교 BK21 산업경영공학교육연구단 박사후연구원으로 재직 중이며, 연구분야는 AI, 빅데이터 애널리틱스, 소셜 데이터 분석, 비즈니스 인텔리전스이다.

이홍철 : 고려대학교 공과대학 산업공학부에서 1983년 학사, University of Texas at Arlington 산업공학과에서 1988년 석사학위를 취득하고 Texas A&M 대학교에서 산업공학 박사학위를 취득하였다. 1996년부터 고려대학교 공과대학 산업경영공학부 교수로 재직하고 있다. 연구분야는 AI, 빅데이터, 정보시스템이다.