Patent2NLP: Improve PatentSBERTa to using abstractive summarization and CPC description

강웅* 김하늘* 이영재* 한영주*
고려대학교 일반대학원 산업경영공학부

Abstract

특허분류(: Patent Classification)는 정부에서 특허 성립요건을 검토할 때 필수적으로 거치는 작업으로 출원된(: applicated) 특허의 발명기술이 어느 기술체계에 해당하는지 정하는 과업이다. 그동안 머신러닝(딥러닝)을 활용한 자동화 특허 분류모델이 광범위하게 연구되었으나 특허청 심사관의 수작업을 완전히 대체하기에는 한계점이 존재한다. 최근에는 Augmented SBERT로 특허 간 코사인 유사도를 활용하여 발명기술 간 거리를 측정하는 방식으로 분류성능을 향상시킨 선행연구 PatentSBERTa가 제안되었다. 그러나 여기에는 BERT 계열의 고질적 단점인 512 토큰으로 길이 제한된 입력 문장 그리고 모델에 입력할 두 문장을 추출하는 과정에서 높은 시간적 비용이 소모된다는 2가지 문제가 존재한다. 이에 본 연구는 Patent2NLP 모델을 제안하여 해당 문제점들을 해결하는 새로운 방법론을 적용하여 특허 분류성능을 더욱 향상시키기 위한 연구를 진행하였다. 입력 data는 특허를 구성하는 가장 중요한 텍스트인 '청구범위(: Claim)', 목적 data는 특허 기술분류체계인 CPC code(: Cooperative Patent Classification, 선진화 특허 분류)이다. Patent2NLP는 Baseline 모델 PatentSBERTa에 비해 2가지 기여점을 갖는다:
1) 입력 텍스트 data 생성요약으로 PatentSBERTa의 입력 토큰 수 제한 해결
2) Augmented SBERT의 입력 data 쌍 구성을 변화시켜 Cross-encoder 시간 감소

Github address: github link
본 연구에 사용된 dataset 및 소스코드는 모두 github_link에 수록하였다.
E-mail address: azxcv070707@naver.com

1 서론

국가 산업 생존에 있어 과학기술의 영향력은 절대적 위상을 차지한다. 이에 각 국은 특허(:Patent) 출원 제도를 통하여 발명기술에 대한 사용 독점권을 발명 주체에게 부여함과 동시에 해당 기술을 세간에 공개하여 기술경쟁을 통한 산업발전을 이루는 것을 궁극적인 목표로 한다. [1] 따라서 기업은 이러한 무한 경쟁 속에서 기술적 우위를 점하기 위한 연구 개발 (Research and Development, R&D)에 역량을 다하고 있다. [2] 기업 간 R&D 경쟁이 가속화되는 만큼 향후 발명 기술 사용권에 대한 높아지는 권리분쟁의 가능성을 고려해야 하기에 특허 출원을 통한 기술 권리화의 중요성이 이제는 기업에 있어 필수로 갖춰야 하는 역량이 되었다. 또한 기술 혁신의 관점에서 기업은 대량의 특허 정보를 활용하여 경쟁사의 선행 기술을 조사하여 이를 유도한다. [3]

이렇게 급속한 기술 발전과 높아지는 지식재산권의 중요성으로 전 세계 특허 출원율은 2009년을 제외하면 2003년부터 해마다 증가하고 있다. [4] 다만, 각 나라의 특허청이 처리해야 하는 특허 data가 늘어나는 만큼 심사관의 업무량이 가중되어 특허 출원 심사 기간이 늘어나는 것에 문제가 있다. 특허 출원 절차에서 기본적으로 진행되는 절차 중 하나가 특허분류로 특허청 심사관이 독점적으로 수행하는 작업이다. [5] 그동안 발명된 기술이 어느 기술 분야에 속하는지에 대한 자동화 분류 연구는 현재까지 활발히 수행되었으나 기본적으로 특허문서는 많은 양의 기술 및 법률 용어로 구성되어 있어 여전히 심사관의 수작업을 완전히 대체하기에는 힘든 것이 현실이다.

현재 특허 기술분류 체계는 IPC(International Patent Classification: 국제 특허 분류체계)와 CPC(Cooperative Patent Classification: 선진 국제 특허 분류체계)로 구성되어 모두 발명기술의 종류를 정할 수 있는 계층적 카테고리이다. 각 최하위 level의 분류체계까지 IPC는 약 6만개, CPC는 약 27만개로 구성되는데 이처럼 CPC가 IPC보다 더 세분화 되어있어 많은 국가들이 CPC 분류체계로 옮기는 추세를 보인다. 직관적 이해를 위하여 CPC 분류체계 구성을 Figure 1과 Table 1 에 작성하였다. [6] [7]

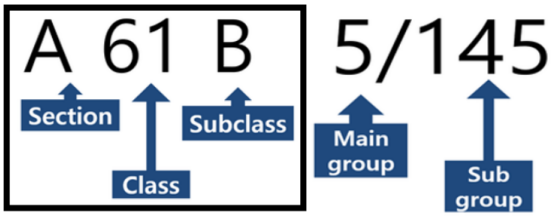

Figure 1: CPC 기술분류체계를 바탕으로 Section부터 Subgroup level까지의 작성법에 대한 예시이다. CPC는 계층적 구조로 9개의 Section부터 Subgroup은 27만개에 달한다. CPC는 정기적으로 특허청에 의해 갱신되고 있어 그 수가 계속 늘어나고 있다. 선행연구에 나타나 있는 CPC 특허 분류는 보통 663개의 SubClass level에서 수행되었다.

CPC Section	Section 설명	CPC Class	Class 설명
A	인간의 필수품	A01	농업;임업
B	처리 조작; 운수	A21	제빵;식용 반죽
C	화학	A22	도살;고기 처리
D	섬유	A23	식품 또는 식료품
E	고정 구조물	A24	담배
F	기계 공학	A41	의류
G	물리학	…	…
H	전기	…	…
Y	새로운 기술 발전의 일반적 구분표	…	…

Table 1: CPC 분류체계에 대한 직관적 이해를 위해 작성하였다. CPC는 최상위 level인 9개의 Section에서 최하위 level인 27만개의 Subgroup까지의 계층적 구조로 구성되며 CPC 자동화 분류는 선행연구인 DeepPatent, PatentBERT, PatentSBERTa의 경우처럼 663개의 CPC Subclass까지 수행되는 경우가 많다. 모든 CPC code에는 이를 설명하는 설명문이 첨부되어 있다. Table 1의 CPC Class는 CPC Section의 하위 level을 구성하며 Section A를 구성하는 level를 예시로 보여준다.

특허는 다양한 정보가 구성된 하나의 문서 형태로 저장한 형식으로 그 중에는 상기 IPC, CPC와 같은 기술분류체계, 출원인, 특허 만료일 등으로 특허에 관한 전반적인 정보를 알려주는 서지정보와 발명의 제목(title), 청구범위(claim), 요약항(abstract)과 같은 텍스트 정보 역시 특허문서에 포함된다. 이 중에서 심사관에 의하여 발명 기술의 권리범위를 결정하게 하는 특허성립에 있어 가장 중요한 항목은 청구범위로 이를 작성할 시 해당 발명 기술에 대한 구성요소를 간결하고 명확하게 작성해야 한다. 청구범위에 기재된 발명기술 구성요소를 기반으로 어느 기술에 해당되는지를 분류체계로 결정한다. 이는 전세계 모든 특허청에서의 특허성립 요건과 일치한다. [8] 자동화 특허 분류를 수행하는 한 가지 방법은 딥러닝과 같은 인공신경망을 자연어처리 task에서 활용할 수 있는데 청구범위와 같은 텍스트 data를 분류모델에 입력하여 해당 발명기술을 CPC 기술분류체계 카테고리로 분류하는 것이다. 이는 실제 특허청에서 가장 빈번하게 발생하는 특허 처리 업무 중 하나에 해당한다. [5] [9] 자연어 처리는 딥러닝 기법의 하위분야로 2017년 트랜스포머 이후 급격한 발전을 거듭하여 지난 몇 년 동안 KNN, K-means, SVM과 같은 머신러닝 분류기법 그리고 순환신경망 기반의 전통적 텍스트마이닝 기법을 대체할 정도로 놀라운 성능을 보인다. 이는 본 연구의 선행연구에서 대규모의 특허 dataset에서도 높은 정확도를 증명하였다. [10] [11] 특히, BERT 이후로 대규모 텍스트 data에 대한 비지도 사전학습 모델에 특정 작업에 대한 추가학습을 진행하는 미세조정으로 이어지는 패러다임이 감정분석, 질의응답 등의 여러 과제에서 탁월한 성과를 보였으며 이러한 사전학습 모델에는 대표적으로 ELMo, BERT, GPT Series 등이 포함된다. [12]–[14]

본 연구는 특허 CPC 카테고리 분류에 이러한 사전학습 모델을 사용함과 동시에 청구범위와 같은 강한 도메인 특성을 갖는 텍스트를 이해하기 위한 대표적 선행연구인 PatentSBERTa를 활용하고자 한다. 여기에는 Augmented SBERT가 활용되는데 기존 SBERT와 마찬가지로 두 특허(문장)간 시맨틱 기법에 의한 코사인 유사도를 활용하여 발명 기술 간의 유사도 거리 측정하여 분류 성능 향상시키는데 중요한 역할을 한다. 추가로 모델에 들어갈 두 문장을 추출하는 과정까지 관여하여 높은 유사도를 도출한다는 점에서 기존 SBERT와의 차이를 가진다. 특허의 강한 도메인 특성에 대한 예시를 Table 2 에서 직관적으로 보여준다. [11], [15], [16]

특허 번호/출원인	10-2017-0000235 / LG전자
발명의 명칭	분리형 스마트 워치
요약 항	본 발명은 스마트 워치 등의 단말기에 있어서 물속에서 비상 사태가 발생한 경우 이러한 위험을 감지하여 외부로 위험을 알리는 단말기를 제공하는 것을 그 목적으로 …
청구 범위	**청구항 1** 사용자 신체에 착용되는 제 1바디; 상기 제1 바디와 결합하고 제1 밀도 값 이하의 밀도를 갖는 제2 바디; 상기 제2 바디와 상기 제1 바디를 결합시키는 체결부; … **청구항 2** 제1 항에 있어서, 상기 제1 바디는, 상기 제2 바디가 안착되는 함몰부를 더 포함하고 상기 체결부는 …

Table 2: '스마트 워치'에 대한 LG 전자의 특허 중 일부를 나타내었다. 특허는 다양한 항목이 포함된 문서로 3개의 텍스트 정보인 '발명의 명칭(title)', '요약항(abstract)', '청구범위(claim)'를 일부 기재하였다. '요약항'은 사용자로 하여금 해당 특허를 쉽게 이해가능하도록 하고 '청구범위'는 기술적 (혹은 법률적) 용어가 특허에 맞게 다수 포함되어 강한 도메인 특성을 가짐을 알 수 있다.

SBERT는 BERT의 문장 임베딩 성능을 개선시킨 모델이라는 점에서 두 특허(문장: 청구범위)간 유사도 도출에 있어 장점을 갖는다. 여기에 Augmented SBERT는 기존 SBERT에 들어간 두 sentence를 STS benchmark dataset과 활용하고자 하는 특허 텍스트 dataset을 활용하여 추출함으로서 특허의 강한 도메인 문제를 완화시킬수 있다는 장점을 갖는다. 그러나 SBERT 모델의 단점은 입력 문장 최대 길이를 512토큰으로 한정하고 있다는 점이다. 만약 특허 청구범위에서 첫 번째 청구항이 평균 167토큰을 갖는다고 하면 10개의 청구범위는 대략 1670토큰을 갖기에 512 토큰을 넘어버려 입력 자체가 불가능하다는 문제가 발생한다. [17]

따라서 본 연구에서는 청구범위 생성요약을 수행하여 해당 문제를 해결할 수 있다. 텍스트 요약에는 추출요약, 생성요약이 있는데 긴 문서의 경우 중요한 정보가 고르게 분포가 되어있기 때문에 생성요약을 선택하였다. 이는 본 연구의 첫 번째 방법론에 해당하는 것으로 모든 특허 data의 청구범위를 510 토큰으로 생성요약 한 것이 첫 번째 청구범위를 사용한 벤치마킹한 PatentSBERTa에 비해서 특허 CPC 분류성능이 얼마나 높아졌는지에 대한 실험을 진행하도록 한다. 이는 PatentSBERTa가 특허분류 과업에서 SOTA 달성하였으며 하나의 청구범위를 padding, truncate하여 토큰을 510으로 고정하였기에 의미가 있을 것이라 판단하였다. 그리고 본 연구의 두 번째 방법론은 다음과 같은 문제점에 기인한다. PatentSBERTa는 상기 Augmented SBERT로 특허 간 거리를 측정하고 그 중 상위값을 갖는 특허값들에 대한 KNN을 수행하는 하이브리드 모델 구조를 갖는다. 여기서 Augmented SBERT는 기존 SBERT 사용 이전 단계에서 labeling 된 두 sentence(: 청구범위)를 추출하기 위해 사전학습된 RoBERTa에 STS benchmark dataset을 미세조정 시키고 labeling 되지 않은 특허 청구범위 쌍을 해당 RoBERTa에 미세조정하여 Cross-encoder로 labeling 시키는 방식으로 구성한다. [15]

$$\frac{n*(n-1)}{2} \tag{1}$$

그런데 Cross-encoder 부분에서 PatentSBERTa는 청구범위 쌍을 활용하는데 이는 n개의 청구범위가 있다면 될 수 있는 조합은 수식 (1)의 개수 만큼 발생하여 비용적 문제를 야기할 수 있다는 단점이 존재한다. PatentSBERTa는 바로 이전 연구인 PatentBERT의 비용적 문제를 언급하며 SBERT 모델계열 구조로 변경한 점을 기여점으로 언급하고 있으나 정작 SBERT에 들어갈 두 sentence를 추출하는 과정에서 비용적 문제가 생기는 것이다. [18] 이에 본 연구는 청구범위 쌍이 아닌 청구범위와 CPC 분류체계 설명문(CPC code description)으로 구성된 새로운 쌍을 제안한다. 여기서 CPC 설명문은 각 CPC 분류체계 코드에 대한 개념을 설명하는 문장으로 모든 각 CPC 분류체계(9개의 최상위 level부터 27만개의 최하위 level)를 설명한다. 분석을 수행하기 위한 전체 특허 dataset 행 값이 149만개이므로 쌍을 구성할 때 CPC 최하위 level의 설명문인 27만개를 사용해도 가능한 Cross-encoder 조합의 수가 줄어들기에 이는 비용감소라는 결과로 이어진다. 기존 (청구범위, 청구범위)를 (청구범위, CPC 설명문)으로 쌍을 변경하여 Augmented SBERT 활용한 코사인 유사도를 구하는 방법이 본 연구의 두 번째 방법론에 해당한다. 이는 선행연구인 PatentSBERTa와 비교실험을 수행하기 위한 목적이므로 사용되는 청구범위는 첫 번째 방법론에서의 요약된 전체 청구범위 그리고 첫 번째 청구범위를 각각 적용하도록 한다.

정리하자면 본 연구는 선행연구 PatentSBERTa의 상기 2가지 문제(입력 문장 토큰 수 한도, 청구범위 쌍에 대한 비용적 문제)를 포착하며 각 문제에 대한 제안 방법론을 적용하여 비교 실험을 수행하고 분류 성능 결과를 도출하도록 함에 있다. 이를 완성시킨 모델을 Patent2nlp라고 명명하였다.

2 관련 연구

최근에 BERT와 같이 범용적 언어 표현으로 이뤄진 대규모 텍스트 data를 비지도 사전학습하고 이를 특정 작업에 맞게 미세조정하는 패러다임의 유용성이 특히 자연어 이해 모델에서 가치를 증명하고 있다.

Hepburn et al. [19]은 트랜스포머 기반의 사전학습된 ULMFiT 모델을 특허분류 태스크에 사용하여 기존 머신러닝 모델에서 높은 분류 성능을 보인다고 알려진 SVM과 비교 실험을 진행하여 효과를 입증하였다. IPC 최상위 레벨 8개 분류 작업에서 F1 점수 78.4%를 얻었는데 이는 기존 SVM 보다 상대적으로 더 좋은 성능값을 가진다.

Li et al. [5]는 텍스트 data인 특허의 '발명의 제목(title)', '요약(abstract)' 항목을 스킵그램 기반의 단어 임베딩으로 변환후 합성곱 신경망을 활용하여 637개의 하위클래스 IPC 카테고리를 자동분류하는 DeepPatent를 제안

하였다. USPTO-2M dataset을 사용하였고 F1 점수 약 43%, 정밀도 약 74%를 얻었다. 대규모 특허 data에 대한 분류 작업을 해결하기 위해 심층 신경망을 적용한 최초의 연구라는 점에서 기여점을 갖는다.

추가로 '발명의 제목(title)', '요약(abstract)'은 '청구범위(claim)'와 달리 범용적 언어표현으로 구성되어 모델이 해당 특허를 이해하기에는 용이할 수 있으나 특허에 담긴 정보량과 가치에 있어 그 중요성은 '청구범위'에 미치지 못하며 이는 특허법으로도 청구범위의 중요성이 명시되어 있다. 또한 IPC는 특히 최하위 level에서 27만개에 달하는 분류체계를 가진 CPC에 비해 수가 한참 모자르기에 최근에는 CPC 기술분류체계로 특허분류를 수행하는 추세를 갖는다.

따라서 Lee et al. [10]는 상기 DeepPatent를 벤치마킹하여 PatentBERT를 제안하였는데 사전학습된 기본 (12층) BERT에 기반한 분류모델로 USPTO-3M dataset을 사용하였으며 DeepPatent와 다르게 입력 data를 '청구범위'로, 목적 data를 CPC 분류체계로 구성하였다. 결과는 F1점수가 66.83%, 정밀도를 84.26% 달성하여 DeepPatent를 대체할 수 있음을 증명하였다.

다음으로 Hamid et al. [11]는 상기 DeepPatent, PatentBERT를 벤치마킹하여 PatentSBERTa를 제안하였다. Hamid는 두 선행연구가 높은 분류 성능값을 갖지만 특허 간 기술 유사도를 측정하는 작업이 수행되지 않았음을 지적하며 이를 활용하여 더욱 향상된 분류기법을 제시하였다. 입력data와 목적 data는 각각 청구범위, CPC 분류체계로 PatentBERT와 동일하지만 BERT의 Cross-encoder 방식으로 인한 비용을 지적하며 이를 SBERT 계열로 대체한다. 특허 간 기술적 거리를 코사인 유사도로 측정할 수 있고 마지막에는 KNN을 사용하여 CPC의 663개의 하위클래스를 직관적으로 분류할 수 있도록 한다. 이처럼 트랜스포머 계열 모델인 상기 PatentBERT, PatentSBERTa와 같은 접근법으로 강한 도메인 특성이 도전과제였던 청구범위를 활용하여 더 세분화된 분류체계인 CPC를 예측하는데 높은 성능값을 증명하였다. [10] [11]

PatentSBERTa의 원리는 다음과 같다:
1) STSb benchmark dataset으로 텍스트 간 유사도를 학습한 모델(cross-encoder)로 특허 별 청구범위 1항(:1번째 특허 청구범위)간 유사도를 구하고 labeling 수행
2) SBERTa 모델을 labeling 된 청구범위 1항 쌍 data를 기반으로 문장 간 유사도 학습
3) 문장 쌍 별 코사인 유사도를 추론하고, 이를 활용하여 KNN 알고리즘을 통해 CPC code 기반 분류 과제 수행

따라서 본 연구에서는 직전 선행연구인 PatentSBERTa를 벤치마킹하여 보다 향상된 특허 분류모델 Patent2nlp를 제안한다.

3 연구 방법론

3.1 Data

본 연구에서 사용된 dataset은 구글 빅쿼리에서 USPTO의 PatentsView을 가져왔다. PatentSBERTa 또한 마찬가지로 PatentsView를 사용하였기에 이를 벤치마킹한다. SQL쿼리문은 PatentSBERTa에서 공개되어 있지는 않으나 PatentSBERTa가 바로 이전 연구인 PatentBERT를 벤치마킹하였고 PatentBERT에는 SQL쿼리문이 작성되어 있어 이를 기반으로 작성하였다. 우선 PatentSBERTa와 마찬가지로 특허 출원연도를 2013-2017, 전체 data 중 8%의 data를 test set 그리고 테이블 구성요소를 특허 청구범위(claim text), 특허번호(id), 특허 출원연도(date), 특허의 제목(title), 요약항(abstract), CPC 분류코드(CPC ids)로 동일하게 구성하도록 한다. 테이블 행의 수도 약 149만개로 거의 일치한다.

다만, 청구범위 같은 경우는 하나는 PatentBERT, PatentSBERTa와 동일하게 첫 번째 청구범위만으로 구성한 테이블, 다른 하나는 선행 연구와 달리 전체 청구범위를 가져온 테이블로 data를 구성한다. 이는 먼저, 첫 번째 방법론은 PatentSBERTa의 입력 문장 토큰 수 한계로 인한 전체 청구범위를 생성요약해서 첫 번째 청구범위만을 입력했을때의 분류성능을 비교하기 위함에 있다. 그리고 두 번째 방법론은 PatentSBERTa의 SBERT 부분에 입력할 두 문장을 추출하는 과정에서 Cross-encoder가 사용되는데 비용적 절감을 위한 data 쌍 변화를 수행한다. 따라서 첫 번째 청구범위만을 사용해서 기존 PatentSBERTa와 비교실험 수행함과 동시에 첫 번째 방법론에서 사용한 생성요약된 청구범위를 두 번째 방법론과 혼합하여 PatentSBERTa와 비교함으로서 방법론 간 교호작용을 확인하기 위함이다. 작성한 SQL 쿼리문은 PatentBERT 쿼리문과 함께 Appendix A에 기재하였다. 또한 Table 3에 특허의 청구범위, 특허번호, 특허 출원연도, 제목 등에 대한 예시를 나타내었다.

행	청구범위	특허번호	출원연도	발명의 제목	요약항	CPC코드
1	A safety vest assembly fitted for use by a wearer, the safety vest assembly comprising: an outer layer; a bullet protection layer . . .	8341762	2013-01-01	Safety vest assembly including a high reliability communication system	There is provided a safety vest assembly including a vest having a vest outer layer defining . . .	A41D,F41H
2	A reinforcing element for . . .	8341763	2013-01-01	Reinforcing element	The invention relates to a . . .	A41D,A63B

Table 3: 각 열에 해당하는 'Claim text', 'id', 'date', 'title', 'abstract', 'cpc ids'은 순서대로 특허 청구범위, 특허번호, 출원연도, 발명의 제목, 특허 요약항, CPC 분류 코드에 해당한다. 총 1492294개로 구성되며 행은 특허를 의미한다.

3.2 방법론1: Abstractive Summarization

Abstractive Summarization(: 생성요약)은 자연어 처리에서 텍스트의 핵심 내용을 추출함과 동시에 문장의 의도, 주제를 이해하고 원문의 내용을 조합하여 새로운 문장을 생성하고 요약한다. 이는 원문에서 단순히 문장을 추출하는 추출요약과의 다른 점이다. 본 연구는 LongT5 모델을 특허 청구범위 생성요약을 위한 모델로 사용한다. BigPatent 특허 benchmark dataset에서 해당 모델이 생성요약 과제에서 SOTA를 달성한 부분은 사용 근거에 해당한다. 추가로 특허는 그 자체로 강한 도메인 특성을 보유하고 있어 해당 자연어에 대한 기계의 이해가 반드시 필요하며, 이에 대한 생성요약을 수행하려면 트랜스포머의 인코더 혹은 디코더와 같은 특정 부분이 아닌 트랜스포머 자체를 쌓아 올린 사전학습 모델이 필요하다고 판단하였다. 이와 같은 모델은 T5, BART, BigBird 등이 있다. LongT5는 T5의 입력 문장 길이가 확장된 것으로 토큰 수 길이에 제한이 없다는 점이 또한 장점으로 작용된다. [20] BigPatent는 특허의 '발명의 상세한 설명(: Patent description)'을 입력으로 '요약(: abstract)'을 타겟 data로 구성하여 생성요약 과제에서 모델에 쌍으로 입력된다. 특허법상 '발명의 상세한 설명'은 우리가 요약하고자 하는 특허 '청구범위(: claim)'를 보정하기 위한 항목이며 마찬가지로 '청구범위' 또한 간결하고 명확하게 작성해야 하는 동시에 '발명의 상세한 설명'으로부터 보정받아야 '청구범위'로 인정받을 수 있다. 즉, 청구범위가 발명기술을 간결하고 명확하게 작성된 강한 도메인을 갖는다면 '발명의 상세한 설명'은 청구범위를 보정하는 매우 긴 문장이며 동시에 강한 도메인이 아닌 범용적 특성을 갖는다. [21] BigPatent의 예제를 Table 4 에 작성하고, 본 연구에서의 방법론을 Table 4 와 비교한 도표를 Figure 2 에 나타내었다.

BigPatent 저자는 기존 상당수 요약과제 benchmark dataset은 CNN/DM, NYT, MultiNews와 같은 뉴스 기사에서 수집되는데 data 특성상 담론 구조가 평평하며 핵심 문장이 단락의 시작 부분에 위치해 있는 등의 문제점으로 생

발명의 상세한 설명(Description)	요약항(Abstract)
FIELD OF THE INVENTION [0001] This invention relates to novel calcium phosphate-coated implantable medical devices and processes of making same. The unique calcium-phosphate coated implantable medical devices minimize immune response to the implant. The coated implantable devices have the capability to store and release one or more medicinally active agents into the body in a controlled manner. BACKGROUND OF THE INVENTION [0002] Cardiovascular stents are widely used in coronary angioplasty procedures to enlarge coronary arteries and thereby allow better blood circulation. Typically this is accomplished by a balloon angioplasty procedure wherein a contracted stent, usually in the form of a metallic mesh tube, . . .	This invention relates to novel calcium phosphate-coated implantable medical devices and processes of making same. The calcium-phosphate coatings are designed to minimize the immune response to the implant (e.g. restenosis in stenting procedures) and can be used to store and release a medicinally active agent in a controlled manner.

Table 4: BigPatent Benchmark Dataset 예시

성요약에 적절하지 않다고 주장하며 이에 비해 BigPatent는 상대적으로 더욱 풍부한 담론구조를 가지며 중요 내용이 단락에 고르게 분포하고 있어 생성요약을 위한 benchmark dataset으로 적절하다고 주장한다. Mandy는 BigPatent dataset에 BigBird, BART와 같이 여러 모델들로 생성요약 과제를 수행하였고 결과를 Rouge점수로 추출하여 성능을 비교한 결과 LongT5가 가장 높은 값을 가짐을 증명하였다. [22]

선행연구 PatentSBERTa는 첫 번째 청구범위만 사용하여 길이가 510토큰을 넘어가면 자르고 부족하면 패딩하는 식으로 모두 510 토큰 값이 되도록 길이를 고정하였다. 본 연구에서는 PatentSBERTa와 비교하기 위해 가장 첫 번째의 청구범위가 아닌 전체 '청구범위'를 510 토큰으로 만드는 생성요약을 진행하며 사용 모델은 LongT5로 선정한다. LongT5는 어텐션에 따라 local, TGlobal로 구분되는데 local은 입력 텍스트를 여러 chunk로 분할하여 처리하고 각 chunk는 독립적으로 인코딩 하는 방식이며 TGlobal은 입력 텍스트 자체를 하나의 시퀀스로 처리하기에 긴 문장에 대해서 local보다 높은 성능을 가진다. 또한 local, TGlobal은 크기에

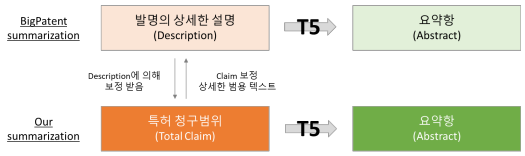

Figure 2: BigPatent dataset은 생성요약을 위한 benchmark dataset으로 특허의 발명의 상세한 설명에 해당하는 'Patent description'과 특허 요약항에 해당하는 'Abstract'가 쌍으로 구성되어 있다. Patent description이 abstract를 대상으로 생성요약을 진행한다. 특허법에 따라 특허의 청구범위와는 보정관계에 있다. 즉, 청구범위는 발명 기술요소를 간결하고 명확하게 작성한 도메인 특화된 텍스트라면 발명의 상세한 설명은 청구범위를 보정하는 매우 긴 범용적 텍스트로 구성한다. 특허 요약항을 대상으로 생성요약을 진행하고자 하는 과제는 본 연구와 일치하며 이러한 BigPatent생성요약의 SOTA 달성한 모델은 LongT5이다. 약 130만개의 data로 구성되었으며 PatentSBERTa dataset과 마찬가지로 미국특허에 해당한다.

따라 각 base, large, xlarge으로 구분된다. 본 연구는 특허의 전체 청구범위 자체가 긴 텍스트로 구성되므로 상기 TGlobal를 선택하도록 한다. BigPatent의 경우와 마찬가지로 생성요약을 위한 입력을 (전체 청구범위(claim), 요약(abstract)) 쌍으로 구성하여 '요약'을 대상으로 생성요약을 수행한다. 단, 컴퓨팅 예산을 고려하여 사전학습된 longt5모델을 미세조정을 시키는 것이 아닌 미세조정된 LongT5모델을 사용하며 크기는 base를 선택하도록 한다. LongT5, PatentSBERTa 모델 모두 Hugging face에 공개되어 있다.

3.3 방법론2: CPC description

CPC 기술분류체계는 계층적 카테고리로 9개의 최상위 level(section)에서 약 27만개의 최하위 level(subgroup)까지 구성되어 있다. PatentSBERTa는 CPC 특허 분류를 663개로 구성된 subclass level에서 수행하였다. 이 과정에서 PatentSBERTa는 Augmented SBERT를 활용하여 특허간 발명 기술의 거리를 측정하여 코사인 유사도를 도출하였는데 발명 기술의 구성요소가 상기 언급한 특허 청구범위에 들어가 있어 청구범위간 유사도를 구하는 것과 동일하다. 그런데, 청구항 쌍이 RoBERTa에 미세조정되는 과정에 Cross-encoder 방식으로 학습되는데 이는 비용적 문제를 발생시키는 단점이 존재한다. PatentSBERTa는 주요 기여점으로 바로 이전 논문인 PatentBERT의 높은 계산비용을 대폭 완화시켰다고 주장하였다. 이러한 관점에서 본 연구 또한 Cross-encoder에 학습되는 data 쌍 구성을 변화시켜 PatentSBERTa의 계산적 비용을 완화시키고자 한다. data 쌍 구성은 (청구범위, 청구범위)에서 (청구범위, CPC 분류체계 설명문)으로 변화시킨다. CPC 분류체계 설명문이란 모든 각 CPC code를 설명하는 문장을 의미한다.

Figure 3를 참고하여 구체적으로 설명해보면, Augmented SBERT는 기존 SBERT에 입력되는 두 개의 문장(문서)을 각 Gold data, Silver data로 구성한 다음 원래의 SBERT에 입력하여 향상된 코사인 유사도 값을 얻기 위한 구조이다. 먼저 Gold dataset은 STS benchmark dataset으로 구성하고 이를 사전학습된 RoBERTa에 미세조정하고 여기에 청구범위 쌍을 입력하여 Cross-encoder 방식으로 학습시켜 labeling 시켜주면 해당 labeling 된 청구범위 쌍을 sampling하여 이를 Silver dataset이라 지칭한다. 상기 Cross-encoder 부분의 문제점이라 하면 만약 n개의 청구범위가 있다고 가정할 시 결과가 수식 (1) 개 조합의 청구범위 쌍이 생성되어 높은 비용을 발생시킨다. 청구범위 쌍으로 특허 간 유사도를 구하고 높은 유사도에 할당된 특허들을 KNN을 사용하여 직관적으로 CPC코드 다중분류하는 기존 방식을 보면 CPC code 분류 과제를 수행하기 위해서 선행되는 코사인 유사도에 청구범위 쌍으로만 구성할 필요는 없다는 결론을 (청구범위, CPC 분류체계 설명문) 방법론을 통해 얻을 수 있다.

상기 PatentSBERTa는 CPC의 663개의 subclass에서 분류과제를 수행하였다는 점에서 마찬가지로 663개의 CPC 설명문이 존재하게 되는 것을 알 수 있다. PatentSBERTa의 data 수가 약 149만개 이므로 설령 최하위 level(subgroup)의 CPC 설명문을 사용하고자 해도 그 수가 27만에 불과해 원 청구범위 쌍 조합보다 적은수의 조합을 얻게 되어 낮은 계산적 비용이라는 장점을 얻게 되는 것이다. 단, 컴퓨팅 예산에 따른 현실적인 상황을 감안하여 663개의 subclass가 아닌 9개의 최상위 level(section)의 CPC를 사용하도록 한다. 향후 연구에서는 고성능 컴퓨팅 자원이 보장된다면 CPC level에 따른 각 청구범위, CPC 설명문의 유사도를 측정하는 과제를 수행할 수 있다.

4 실험 결과

4.1 방법론1: Abstractive Summarization

GPU 48GB 크기의 Workstation 서버컴퓨터를 사용하였으나 모델 자체의 크기에 다른 컴퓨팅 리소스 부족으로 인하여 결국 Pretrained LongT5 모델의 도메인 텍스트를 활용한 fine-tuning이 아닌 생성요약 작업에 대하여 기존 fine-tuning된 LongT5 모델을 Hugging face에서 선정하였다. 이에 대해 첨언하자면 LongT5 large size 모델의 경우 인코더, 디코더 블록이 각 24개로 구성되며 각 블록에는 레이어가 10개로 구성되어 있다. 이는 forward 과정 중에 레이어마다 그레디언

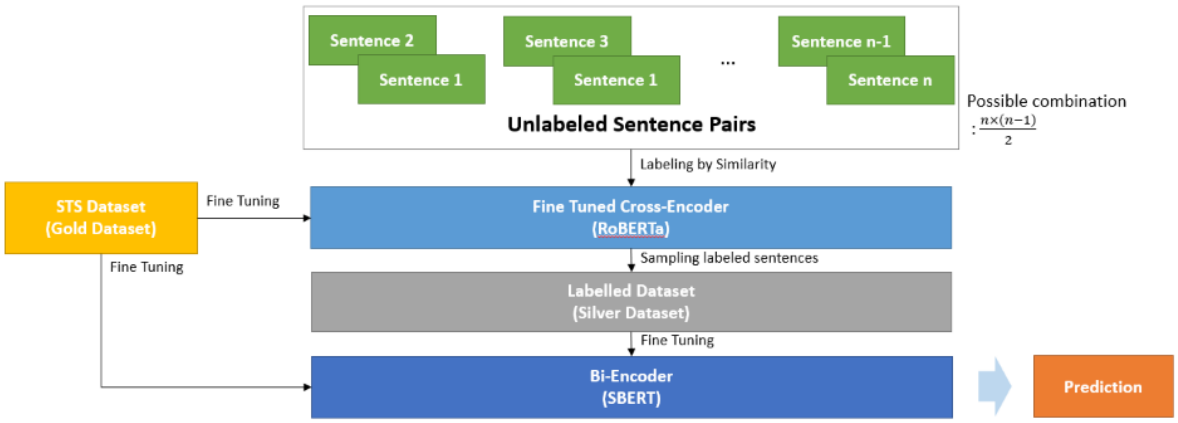

Figure 3: PatentSBERTa의 Augmented SBERT가 작동하는 전체적인 과정을 작성하였다.

트 정보를 누적해서 생성하고 미세조정할 시 모델에 대한 파라미터 전체를 업데이트 하는 방식이라 배치사이즈를 1로 설정해도 컴퓨팅 budget 한계로 학습이 제대로 수행되지 않았다.

이에 Hugging face에서 fine-tuning된 LongT5 모델인 "pszemraj/long-t5-tglobal-base-16384-book-summary" [23]을 사용하여 배치사이즈는 1로 설정하고 학습률은 선행연구 PatentSBERTa와 동일하게 0.001로 구성한다. 그리고 마지막 디코더 블락을 제외한 나머지 모든 블록에 대해 freeze 처리하여 파라미터를 고정시킨다. 이렇게까지 경량화를 시켜야 GPU 48GB 서버컴퓨터에서 겨우 학습을 시킬 수 있었다. 본 연구자는 먼저 모델 생성요약 성능에 대한 평가를 수행하였다. 입력은 특허 '발명의 상세한 설명(description)' 혹은 특허 전체 '청구범위(claim)'이며, 출력은 특허 '요약항(abstract)'이다. 첨언하자면 claim은 description에 의하여 보정받아 발명기술에 대해 간결하고 명확하게 작성되며 description은 claim을 보정해야하기에 일반적으로 매우 긴 문장으로 구성된 범용적 텍스트이다. 그리고 abstract는 해당 특허를 알기 쉽게 설명하기 위한 짧은 범용 텍스트로 구성되어 있으며 claim, description보다 짧은 sequence를 가지므로 생성요약의 target으로 정했다. 생성요약 실험 결과는 Table 5 에 굵은 글씨로 구성하였다.

먼저 Google의 LongT5(TGlobal, large)와 비교하기 위해 선정한 Pszemraj의 LongT5(TGlobal, base)을 67072 sample에 달하는 BigPatent dataset으로 생성요약 실험을 진행하였다. 또한 전체 청구범위(claim)를 입력으로 가져간 실험에서는 시간적 제약을 고려하여 BigPatent dataset 내 6000개에 달하는 sample을 비복원 추출하여 진행하였다. description 입력에 대한 두 모델의

Model	Input	Evaluation Metric[%]		
		ROUGE-1	ROUGE-2	ROUGE-L
LongT5 (large, google)	Description	70.38	56.81	50.01
LongT5 (base, pszem-raj)	**Description**	**34.76**	**7.87**	**19.98**
LongT5 (base, pszem-raj)	**Total Claims**	**35.13**	**11.22**	**30.38**

Table 5: 사용 모델에 따른 Hugging face 생성요약 성능 비교 (pszemraj/long-t5-tglobal-base-16384-book-summary)

ROUGE score 비교 결과를 보면 모델 크기에 따른 값의 차이가 크게 나는 것을 확인하였다. 여기서 입력을 기존에 하고자 했던 전체 청구범위(: Total claim)로 변경하였을 때, 생성요약 성능이 개선됨에 따라 청구범위에 대한 생성요약이 특허의 핵심내용을 잘 반영하는 것으로 확인되었다. 이는 상기 '청구범위(claim)'와 '발명의 상세한 설명(description)'이 보정관계에 있는 것을 원인으로 판단하였다. 생성요약을 진행한 다음에 본 연구자가 진행하고자 하는 1번째 청구범위와 요약된 전체 청구범위로 분류 성능을 베이스라인 모델이자 현재 특허분류에서 SOTA 달성한 PatentSBERTa로 비교하고자 한다. PatentSBERTa는 1번째 청구범위만 사용하여 특허분류를 수행하였다. 훈련 sample은 5000, 검증 sample은 1000개의 data 그리고 (k=6) k-fold 교차 검증을 진행하였다. 실험 결과는 Table 6 에 작성하였다.

Table 6 를 보면 선행연구 PatentSBERTa에 사용한 data 수는 1492294개이나 본 연구는 상기 제한된 컴퓨팅 budget, 분석 시간상의 문제로 그보

PatentSBERTa	Micro Average[%]			
Input	Accuracy	Precision	Recall	F1 score
Claim 1	78.7	38.4	38.4	38.4
Summary	**79.0**	**39.2**	**39.2**	**39.1**

Table 6: 재현한 PatentSBERTa 모델에서 입력물의 차이에 따른 분류 성능 비교

다 적은 data로 분석한 결과 전체적인 분류 성능 자체는 선행연구에서 밝힌 성능보다 낮게 측정되었다. 하지만 본 연구는 선행연구와 정확한 비교를 위해 같은 수의 dataset을 요약된 전체 청구범위 뿐 아니라 1번째 청구범위에도 같이 적용하여 비교하였으며 hyperparameter도 동일하게 설정하였다. 즉, 향후 연구에서는 단순히 data 수를 늘려서 성능비교를 그래프 상으로 비교할 가치가 있다고 판단한다. 최종 실험결과로 입력 1번째 청구범위와 전체 청구범위(: Total claim) 생성요약에 대하여 PatentSBERTa 모델을 사용하였을 경우 분류성능 차이가 존재함을 증명하였다. 이는 현재 CPC 특허분류에서 SOTA 달성한 기존 PatentSBERTa 모델을 본 연구자가 제안한 방법론(전체 청구범위 생성요약 및 PatentSBERTa에 입력)이 본 비교실험의 분류성능에서 능가하였기에 향후 추가실험으로 SOTA 달성하고자 하는 목표가 충분한 의미를 가질 것이라 사료된다.

4.2 방법론2: CPC description

본 방법론은 특허의 Multi-label classification 과제에서 직접적인 CPC description의 학습을 통해, claim 간 유사도에 기반하여 inference된 label의 학습보다 학습 효율성 및 분류 정확도를 제고할 수 있다는 가설을 바탕으로 제안되었다. 사용한 dataset 구성은 선행연구 PatentSBERTa, 방법론1과 동일하게 설정하였다. (총 1,492,294건, train set: 1,372,910건, test set: 119,384건) dataset 전처리 과정에서 실험 초기에는 train set 특허 전체에서 기존의 CPC Section 9개, Class 132개, Subclass 663개 각 label에 대해 positive Labeling을 수행 후, train set 특허 중 약 27%에 대해 기존 CPC code와 다른 전체 label의 50%에 대해 nagative labeling을 수행하였다. 그 결과, 전체 instance가 50,851,209개로, 한 epoch 학습에 1,600시간이 소모될 것으로 예상되었다. 제한된 학습 여건에 최대한 맞추기 위해, CPC Label은 9개의 Section, Nagative Labeling은 약 30%에 대해 전체 label의 10%에만 한정하여 축소하였다. 그러나, 이 방법 또한 성능이 좋지 못하다는 한계가 있었고, 이에 따라 negative sample 수를 조정하여 해결하고자 하였다. 따라서 최종적인 dataset 전처리 방법은 다음과 같다. 먼저, train set으로 지정된 특허 1,372,910건에 대해서 25%를 random sampling한 후, 각각의 특허와 전체 CPC section code를 matching한다. 9개 Section code가 각각의 특허의 기존 CPC code와 일치하는 경우(이하 positive) 1.0, 다른 경우(이하 negative) 0.0으로 유사도 score를 부여하여 instance를 labeling 한다. CPC Code를 Description으로 변환하여 Claim을 LHS로, CPC description을 RHS로 하는 sentence pair를 만든다. 이 때 sentence pair의 길이가 이를 초과할 시 truncation, 미만일 경우 padding하여 BERT input에 510개 토큰을 맞춘다. truncation의 경우 sequence 길이가 더 긴 LHS에 수행한다. 이러한 과정을 거쳐 최종 train set은 positive Instance 343,227개, negative Instance 2,745,816개로 구성되었다.

Number of Label	Methods	Micro Average[%]		
		Precision	Recall	F1
IPC Subclass:645	DeepPatent(CBOW)	71.2	38.9	50.3
	DeepPatent(Skip-gram)	81.0	38.6	52.3
	DeepPatent(fastText)	80.9	40.2	53.7
	PatentNet(BERT)	85.7	42.9	57.2
	PatentNet(XLNet)	86.0	42.9	57.2
	PatentNet(RoBERTa)	86.2	41.9	56.4
	PatentNet(ELCTRA)	86.3	41.7	56.2
CPC Section:8	ULMFIT SVM(2018)	N/A	N/A	78.0
	PatentBERT(2020)	N/A	N/A	81.0
	PatentSBERTa	79.0	90.0	82.4
CPC Section:9	방법론2*	70.8	46.5	56.1
	방법론2**	**81.9**	**53.8**	**64.9**

Table 7: 최종 모델 평가 지표가 Sampling 방법을 바꾸고 나서 이전보다 개선되었다. Benchmark 모델과 비교했을 때, Label 수가 비슷한 PatentSBERTa의 경우 F1 지표와 Recall 값은 다소 낮게 나타난 편이나, Precision 측면에서는 진전되었다. 추가로 'number of label'의 CPC section 수에서 본 연구와 PatentNET가 서로 차이나는 것을 확인할 수 있는데 이는 CPC 분류체계 자체가 정기적으로 갱신되고 있기에 발생하는 현상이다. (방법론2*: sampling 방법 변경 이전, 방법론2**: sampling 방법 변경 이후)

실험 결과 평가는 각각의 test set instance에서 CPC section 중 유사도가 가장 높은 하나만을 predicted true로 하여, Multi-label classification을 평가하였다. 이때, 평가지표로는 negative sample과 positive sample 간 imbalance를 고려하여 Micro Average Precision, Recall, F1 score를 사용하였다. 실험 결과는 PatentNET [24]과 PatentSBERTa를 Benchmark로 하여 성능을 비교하였고 결과는 Table 7 에 정리하였다. 최종 모델의 평가 지표는 sampling 방법을 개선하기 전보다 개선된 것을 확인할 수 있다. benchmark 모델과 비교했을 때, label 수가 비슷한 PatentSBERTa의 경우 F1 지표와 Recall값은 다소 낮게 나타난 편이나, precision 측면에서는 진전되었음을 확인하였다.

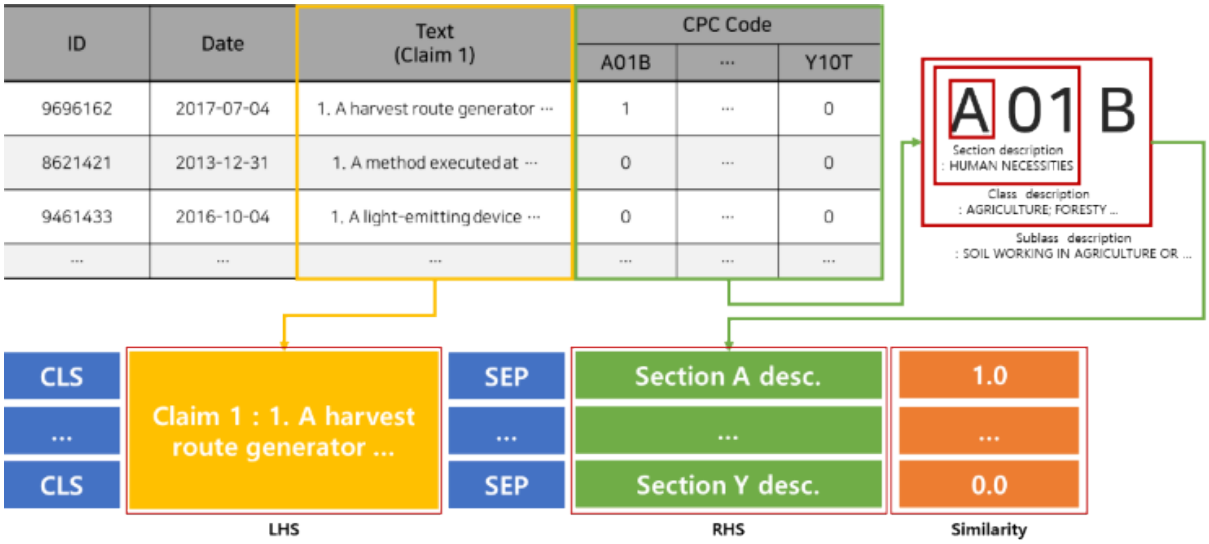

Figure 4: CPC 분류체계 설명문(CPC description)을 활용한 방법론

4.3 방법론 1 + 2

방법론1인 Total claim(전체 청구범위)을 Text Abstractive Summarization한 것과 기존 PatentSBERTa에서 입력 data로 사용한 1번째 청구범위(claim1)를 LHS로 정하여 방법론2인 CPC description으로 학습된 모델에 입력하여 비교실험을 진행하였다. 이는 방법론 간의 교호작용을 확인하기 위함으로 test set을 활용한 결과를 Table 8 에 제시하였다. 실험 결과는 Summarization한 경우보다 오히려 claim1의 평가 지표가 높게 나타남을 확인할 수 있다.

LHS	Micro Average[%]			
Input	Accuracy	Precision	Recall	F1 score
Claim 1	89.9	53.0	53.0	64.5
Summary	**89.1**	**78.6**	**50.7**	**61.6**

Table 8: LHS에서 'claim1'은 기존 PatentSBERTa에 방법론2인 CPC description을 적용했을 때, 'Summary'는 방법론1인 Text abstractive Summarization에 방법론2를 추가로 적용했을 때를 의미한다.

5 결론

본 연구에서는 자동화 특허분류에 있어 최근 선행연구인 PatentSBERTa를 baseline 모델로 정하여 해당 연구의 2가지 한계점을 해결하고자 이에 대한 새로운 방법론을 적용하여 특허 분류성능을 향상시키기 위한 실험을 진행하였다. 먼저, PatentSBERTa에서 저자는 입력 문장 토큰 수 제한이라는 한계로 오직 하나의 청구범위만 입력하였다고 언급하였다. Patent2NLP 방법론1은 BigPatent benchmark dataset의 Abstractive Summarization 과제에서 SOTA 달성한 LongT5를 전체 특허 청구범위에 사용하여 510 토큰으로 문장 임베딩 도출하여 이를 PatentSBERTa에 입력함으로서 비교실험 하였다. 결과는 모든 분류성능 지표에서 PatentSBERTa를 능가하였다. 그러나 여기에는 몇 가지 한계점이 존재한다. 먼저, 제한된 컴퓨팅 budget으로 인한 9개의 CPC Section을 제외한 다른 CPC 하위 level을 target data로 구성하지 못한 점, 전체 약 149만개 중 극히 일부의 data만으로 분석한 점, PatentSBERTa와의 성능차이가 크지 않다는 점 등 향후 추가실험의 필요성이 제기된다.

방법론2에서도 의의와 한계점이 존재한다. PatentSBERTa는 Augmented SBERT를 통해 특허 간 기술적 유사도가 높은 특허를 추출하여 최종적으로 분류성능을 향상시키는데 기여한다. 이 때, 높은 유사도를 얻게 해줄 SBERT에 들어갈 두 문장을 Gold dataset, Silver dataset이라 지칭하는데 여기서 STS benchmark dataset이 Gold dataset으로 구성되며 Silver dataset은 labeled된 특허 청구범위 쌍을 sampling하여 구성한다. 문제는 그 이전에 청구범위 쌍을 RoBERTa에 labeling 시키는 과정에서 cross-encoder 방식으로 학습되는 부분이 높은 시간적 비용을 야기한다는 것이다. Patent2NLP의 방법론2는 청구범위 쌍이 아닌 (청구범위, CPC 분류체계 설명문)으로 구성 변경하여 cross-encoder에 들어가는 시간적 비용을 줄이는 것이다. 여기에는 PatentSBERTa 저자가 이전 연구인 PatentBERT보다 낮은 시간적 비용을 기여점이라 하였는데 마찬가지로 더욱 빠른 분석 시간을 도출한다면 충분히 기여점이 될것이라 판단하였다. 방법론2의 실험 결과로 기존 방법보다 빨리 학습을

하는데는 성공하였으나 한계점 또한 존재한다. 본 연구자는 방법론1과 2사이에 교호작용을 확인하기 위해서 PatentSBERTa와 마찬가지로 첫 번째 청구범위만을 사용하고 방법론2를 사용한 것과 방법론1을 적용하고 나서 추가로 방법론2를 사용한 결과를 비교하였으나 오히려 모든 분류성능에서 2가지 방법론 결합한 실험의 성능이 감소하였다. 이는 방법론2의 모델이 1번째 청구범위를 통해 학습되었다는 점을 원인으로 판단하였다. 그리고 또한 PatentSBERTa에서 하나의 청구범위당 평균 167 토큰을 가지고 있다고 기재된 것처럼 특허가 많은 항목의 청구범위를 보유한 상태에서 강제로 510 토큰으로 생성요약하면 많은 정보가 한정된 길이의 문장에 있는 상태가 된다. 청구범위와 CPC 설명문 쌍의 cross-encoder 학습 부분에서 사용된 CPC 설명문은 9개 Section의 최상위 level로 긴 텍스트를 가지는 설명문은 아니기에 해당 청구범위와 CPC 설명문 간의 정보량 차이가 극심하게 커져 낮은 유사성을 갖게된 점이 주 원인으로 판단하였다. 향후에는 충분한 컴퓨팅 budget으로 27만개의 CPC subgroup까지 level 별로 실험할 시 특허 분류성능 향상의 가능성을 기대한다.

참고 문헌

[1] M. Risov and K. Kasravi, "Patent mining - discovery of business value from patent repositories," *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pp. 54–54, 2007. DOI: 10.1109/HICSS.2007.427.

[2] L. Wang, G. li Luo, A. Sari, and X.-F. Shao, "What nurtures fourth industrial revolution? an investigation of economic and social determinants of technological innovation in advanced economies," *Technological Forecasting and Social Change*, vol. 161, p. 120 305, 2020. DOI: https://doi.org/10.1016/j.techfore.2020.120305.

[3] S. Lee, H. joo Lee, and B. Yoon, "Modeling and analyzing technology innovation in the energy sector: Patent-based hmm approach," *Computers Industrial Engineering*, vol. 63, pp. 564–577, 2012. DOI: https://doi.org/10.1016/j.cie.2011.12.002.

[4] "World intellectual property indicators 2018," *WIPO*, pp. 5–35, 2018.

[5] S. Li, J. Hu, Y. Cui, and J. Hu, "Deeppatent: Patent classification with convolutional neural networks and word embedding," *Scientometrics*, vol. 117, pp. 721 –744, 2018.

[6] 강지호, 김종찬, 이준혁, 박상성, and 장동식, "Iot와 wearables 기술융합을 위한 특허동향분석," *한국지능시스템학회 논문지*, vol. 25, pp. 306–311, 2015.

[7] B. Degroote and P. Held, "Analysis of the patent documentation coverage of the cpc in comparison with the ipc with a focus on asian documentation," *World Patent Information*, vol. 54, S78–S84, 2018. DOI: https://doi.org/10.1016/j.wpi.2017.10.001.

[8] 윤기웅, "Understanding and interpreting specification and claims," *특허청 kipo 특허심사제도과*,

[9] V. Korde and C. N. Mahender, "Text classification and classifiers: A survey," *International Journal of Artificial Intelligence & Applications*, vol. 3, pp. 85–99, 2012.

[10] J.-S. Lee and J. Hsiang, "Patentbert: Patent classification with fine-tuning a pre-trained bert model," *ArXiv*, vol. abs/1906.02124, 2019.

[11] H. Bekamiri, D. S. Hain, and R. Jurowetzki, "Patentsberta: A deep nlp based hybrid model for patent distance and classification using augmented sbert," 2021.

[12] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," pp. 2227–2237, 2018.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[14] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.

[15] N. Thakur, N. Reimers, J. Daxenberger, and I. Gurevych, "Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks," *arXiv preprint arXiv:2010.08240*, 2020.

[16] S. Sarica, J. Luo, and K. L. Wood, "Technet: Technology semantic network based on patent data," *Expert Systems with Applications*, vol. 142, p. 112 995, 2020.

[17] D. Somaya, "Patent strategy and management: An integrative review and research agenda," *Journal of management*, vol. 38, pp. 1084–1114, 2012.

[18] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[19] J. Hepburn, "Universal language model fine-tuning for patent classification," *Proceedings of the Australasian Language Technology Association Workshop 2018*, pp. 93–96, 2018.

[20] "Https://paperswithcode.com/dataset/bigpatent,"

[21] E. Sharma, C. Li, and L. Wang, "Bigpatent: A large-scale dataset for abstractive and coherent summarization," *arXiv preprint arXiv:1906.03741*, 2019.

[22] M. Guo, J. Ainslie, D. Uthus, S. Ontanon, J. Ni, Y.-H. Sung, and Y. Yang, "Longt5: Efficient text-to-text transformer for long sequences," *arXiv preprint arXiv:2112.07916*, 2021.

[23] A. Haghighian Roudsari, J. Afshar, W. Lee, and S. Lee, "Patentnet: Multi-label classification of patent documents using deep learning based language understanding," *Scientometrics*, pp. 1–25, 2022.

[24] "Pszemraj/long-t5-tglobal-base-16384-book-summary," DOI: https://huggingface.co/pszemraj/long-t5-tglobal-base-16384-book-summary.

A Appendix

Google Patents Public Datasets on BigQuery 에서 특허 데이터를 가져오기 위한 SQL문으로 PatentBERT 논문 하단에 첨부된 것을 Figure 5 에 가져온 것이다. 본 연구인 Patent2NLP는 선행연구 PatentSBERTa를 벤치마킹하였고 PatentSBERTa는 바로 이전연구 PatentBERT를 벤치마킹하였다. 다만 PatentSBERTa는 데이터를 어떻게 추출하였는지에 대한 구체적 SQL문을 공개하지 않았기에 본 연구자는 PatentBERT에 작성되어 있는 SQL문을 기본 바탕으로 구성한 다음 PatentSBERTa 데이터 구성에 맞게 변화하여 Figure 6에 작성하였다. 즉, PatentBERT SQL 쿼리는 Figure 5에, Our model인 Patent2NLP SQL 쿼리는 Figure 6 에 작성하였다.

```
SELECT STRING_AGG(distinct t2. group_id order by t2.
group_id) AS cpc_ids, t1.id, t1.date, text
 FROM `patents-public-data.patentsview.patent` t1,
 `patents-public-data.patentsview.cpc_current` t2,
 `patents-public-data.patentsview.claim` t3
 where t1.id = t2.patent_id
 and t1.id = t3.patent_id
 and timestamp(t1.date) >= timestamp('2013-01-01')
 and timestamp(t1.date) <= timestamp('2013-12-31')
 and t3.sequence='1'
 and t1.type='utility'
 group by t1.id, t1.date, t3.text
```

Figure 5: PatentBERT SQL 쿼리

```
SELECT
  STRING_AGG(DISTINCT t3.text, ' ' ORDER BY t3.text ASC) AS claim_text,
  t1.id,
  t1.date,
  t1.title,
  t1.abstract,
  STRING_AGG(DISTINCT t2.group_id ORDER BY t2.group_id) AS cpc_ids
FROM
  `patents-public-data.patentsview.patent` t1
JOIN
  `patents-public-data.patentsview.claim` t3
ON
  t1.id = t3.patent_id
JOIN
  `patents-public-data.patentsview.cpc_current` t2
ON
  t1.id = t2.patent_id
WHERE
  t1.type = 'utility'
  AND TIMESTAMP(t1.date) >= TIMESTAMP('2013-01-01')
  AND TIMESTAMP(t1.date) <= TIMESTAMP('2017-12-31')
GROUP BY
  t1.id,
  t1.date,
  t1.title,
  t1.abstract
ORDER BY
  t1.id ASC;
```

Figure 6: PatentBERT SQL 쿼리 기반하여 PatentSBERTa dataset 추측한 SQL 쿼리, Our model인 Patent2NLP dataset 에 동일하게 구성