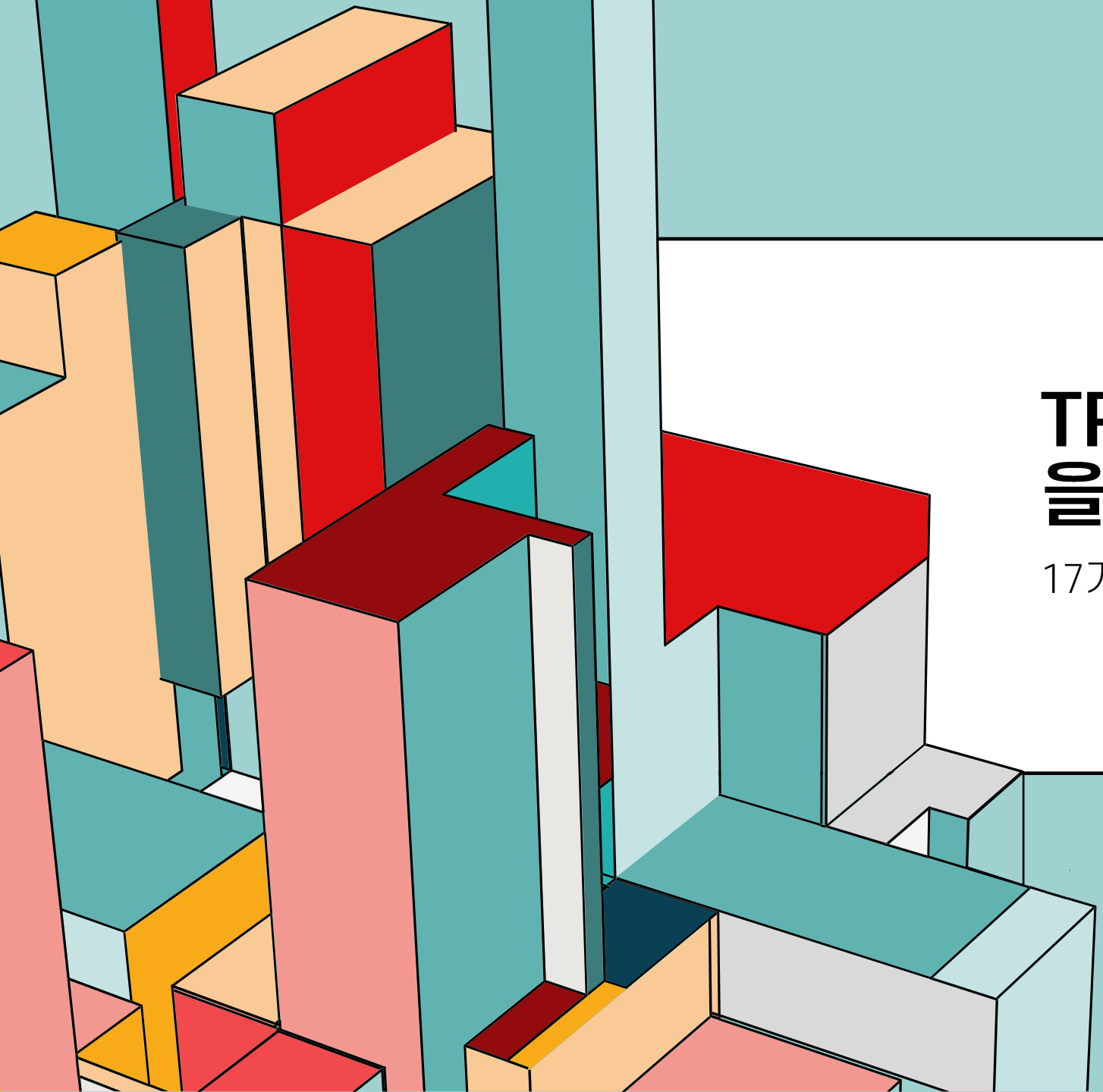
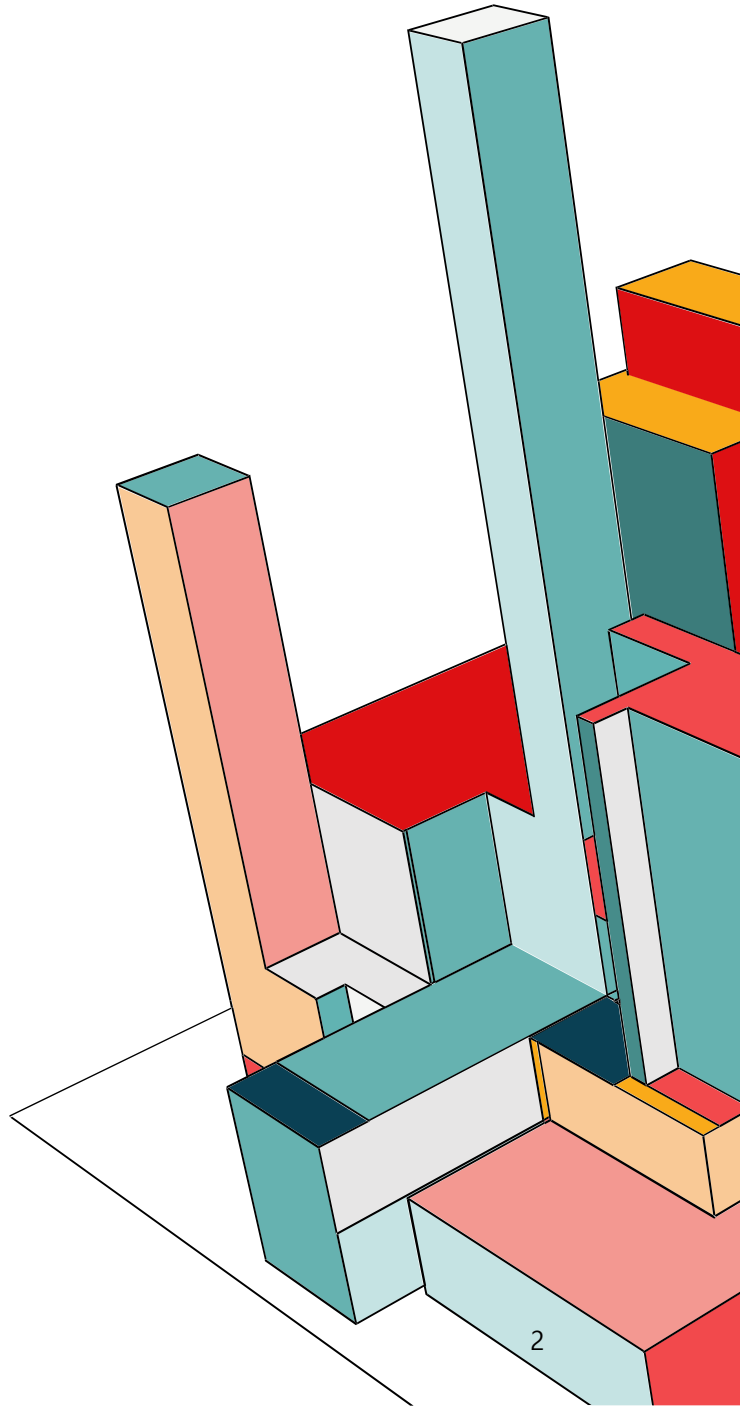


# TRANSFORMER 모델을 이용한 표절 탐지

17기 송지훈, 우윤규, 황우현

# 목차





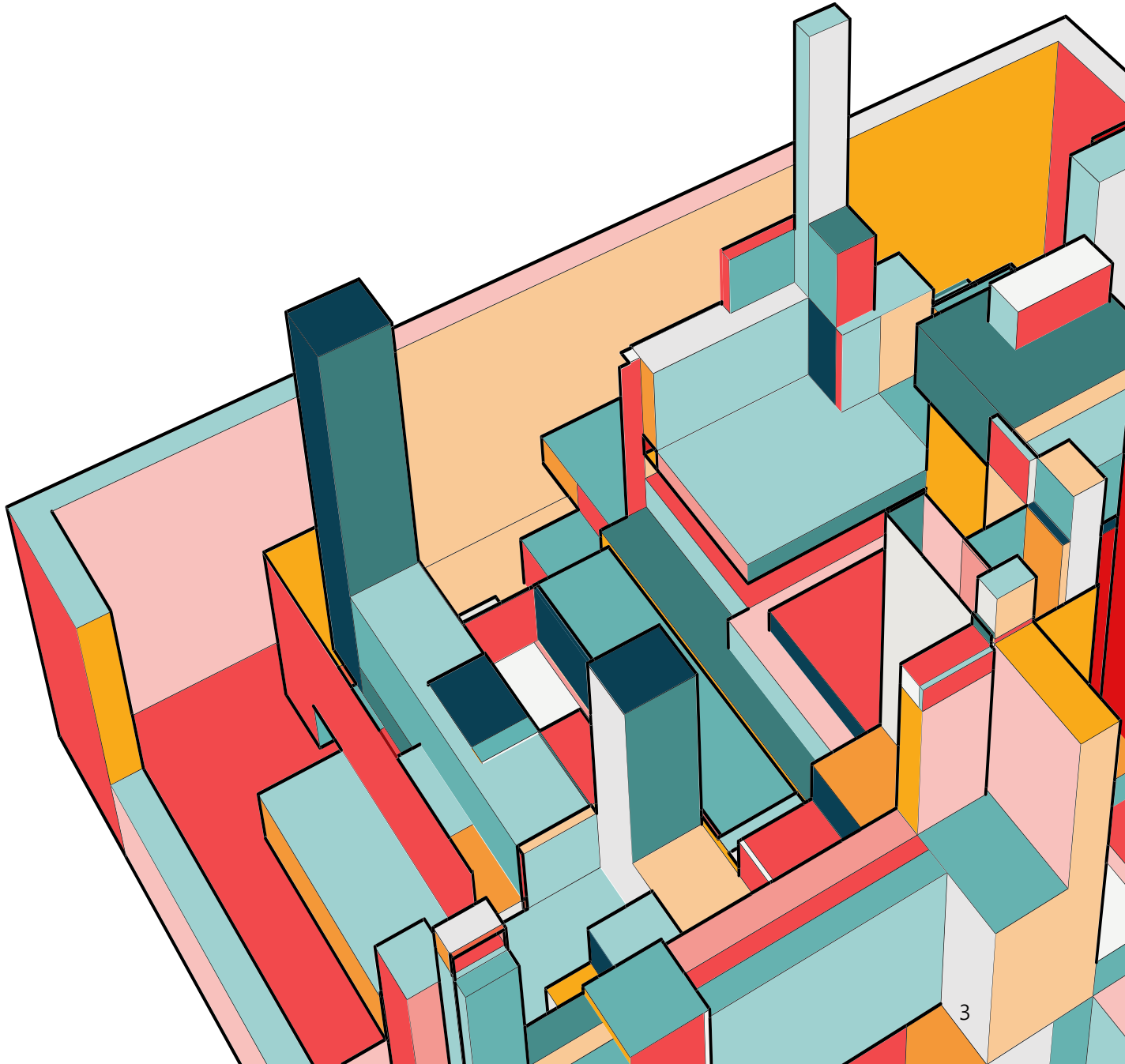
# 1. 표절 탐지

표절 탐지에는 2가지 주요한 제한사항이 있습니다.

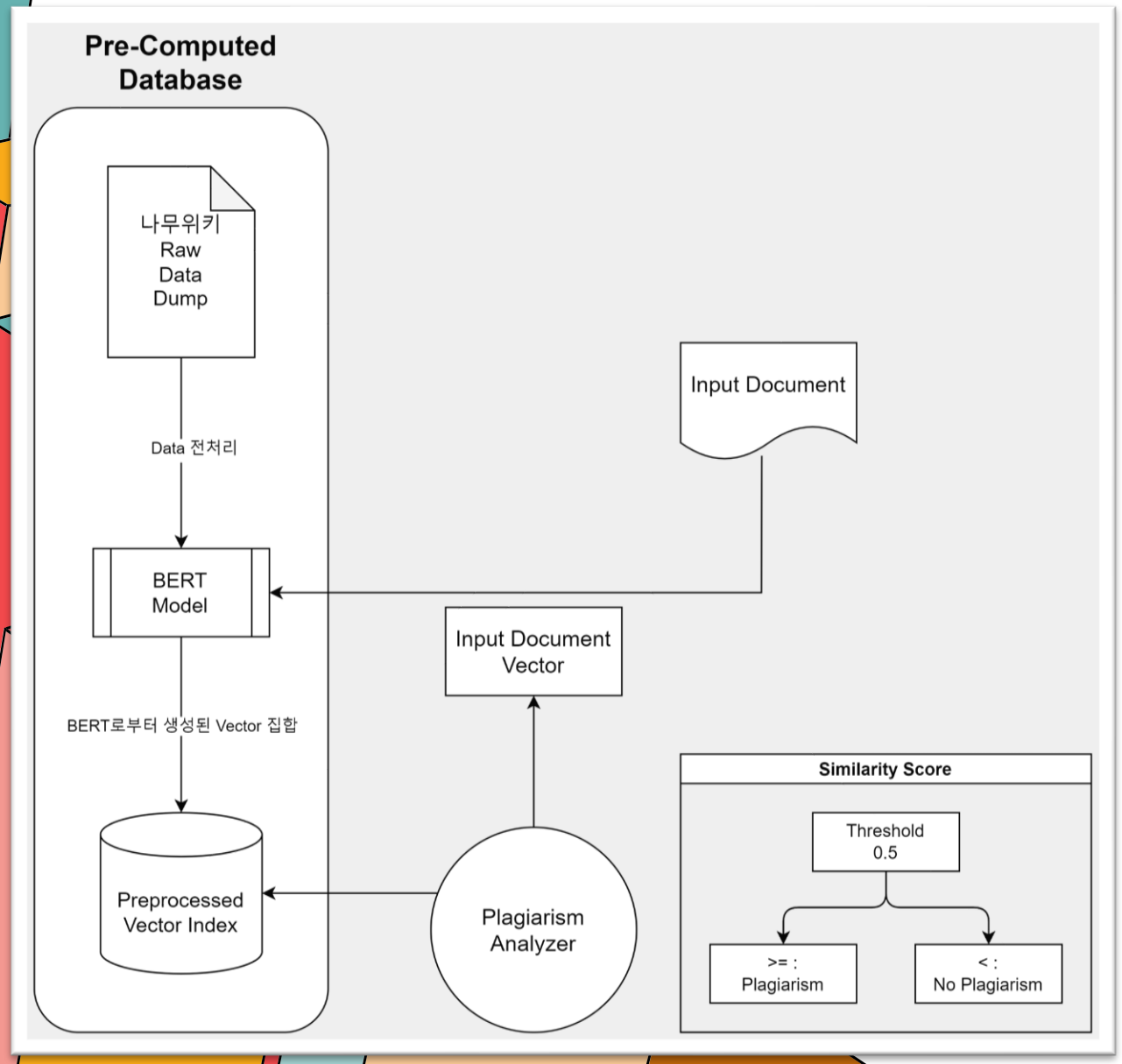
첫 번째로는 전체 문맥을 보면서 동의어와 반의어를 고려하기가 어렵습니다.

두 번째로는  맥락과 아예 다른 내용이 나온다면 탐지하기가 어려울 수 있습니다.

본 프로젝트에서는 이를 극복하기 위해서 Transformer 기반 Pre-trained모델을 활용하였습니다.



# WORK FLOW




## 데이터 베이스 구축

나무위키의 데이터(86만 7천건)을 미리 토큰화 시킨 Database를 미리 구축

## Pre-trained 모델 사용

KoBERT 모델을 활용하여 토큰화 된 데이터를 벡터화

## 유사도 계산

벡터화된 데이터들을 기반으로 유사도 계산(코사인 유사도)

## 표절 유무 판단

사용자가 미리 설정한 threshold값 기준으로 유사도가 그 값 이상이면 표절, 아니면 정상으로 판단

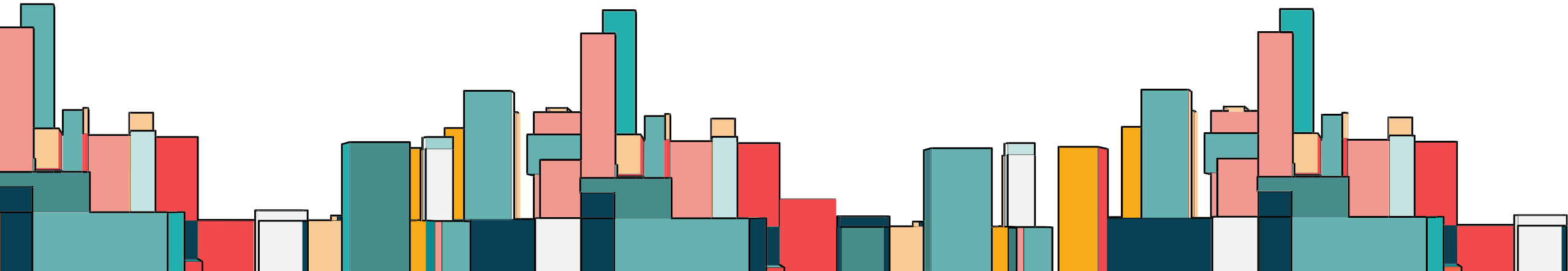
# 2. 데이터 전처리

## 1) 단일 문서의 index 생성

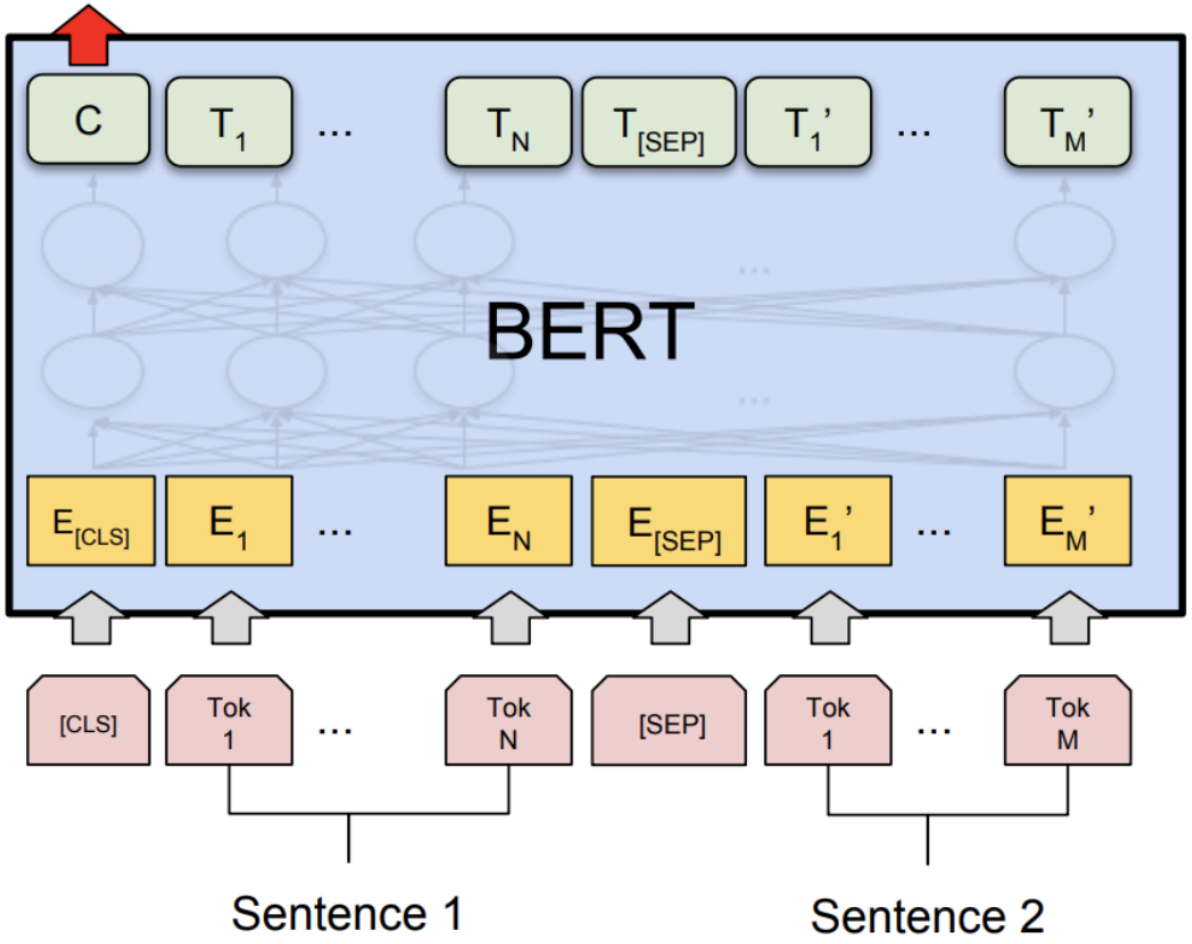
나무위키 데이터들을 Pre-trained된 KoBERT Model의 tokenizer을 활용하여 input_ids, attention_mask를 반환. 또한, 길이를 모두 동일하게 맞춰주기 위해서 Post-padding 수행.

## 2) 단일 문서의 벡터 표현 생성

1)에서 수행한  input_ids, attention_mask을 기반으로 모델에 넣음으로써 해당 문서를 나타내는 벡터 계산



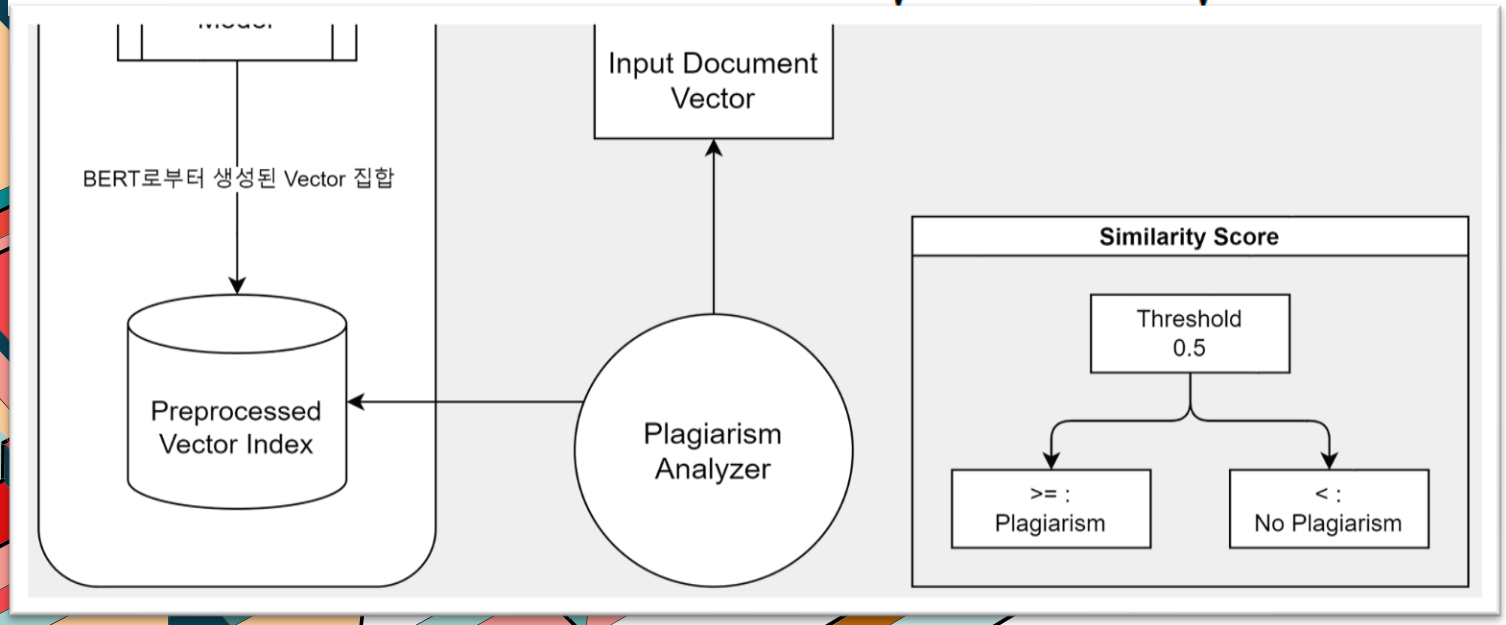
# 3. KoBERT, 유사도 계산




- ✓ 한국어 위키의 데이터를 기반으로 만든 BERT모델
- ✓ 본 프로젝트에서는 해당 모델을 활용해서 문서 간의 유사도를 계산
- ✓ 이때, BERT모델의 마지막 결과값이 Class Label이 아닌, 그 전의 Vector값을 기반으로 코사인 유사도를 계산하여 문서 간의 유사도를 계산하여 새로운 문서가 나무위키로부터의 표절 유무를 가릴 수 있습니다.

# 3. KoBERT, 유사도 계산

$$similarity = cos(\Theta) = \frac{A \cdot B}{||A||\ ||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$




좌측의 코사인 유사도를 계산하는 식을 활용하여 새로운 문서가 들어왔을 때, 원래 Database에 있던 문서들과 얼마만큼 유사한지를 나타내는 코사인 유사도를 계산합니다.

또한, 이에 대해서 사용자가 미리 지정해둔 특정 기준(Threshold)를 넘으면 나무위키에서 표절을 해왔다고 볼 수 있습니다.

이렇게 사전 학습된 모델을 통해서 전체적인 문맥을 고려하면서 벡터화 시킨 후에 표절 유무를 가림으로서 처음에 표절 탐지에 있어 제기하였던 문제점을 어느정도 극복할 수 있습니다.

# 4. 결론

```
plagirism_check(namu,vector_index,plagiarism_threshold=0.5)
```

```
{'article_submitted': '연구성과가 최초임을 어필하고자 동료평가(Peer Review)를 건너뛰고 선공개하는 곳이다. 즉 공개만으론 학술적 가치가 부여되지 않은 상태다. 동료평가의 의미 및 의의는 해당 문서 참고.₩n부정적으로는 애초 걸러져야 할 논문도 올라올 수 있다. 대표 사례로는 송유근 논문 표절 사건 이후, 송유근이 올린다던 새 논문을 빙자한 또 다른 표절 논문을 올렸던 것도 arXiv였다. ##narXiv 프리프린트의 양식을 보면 APL 제출을 위해 작성 했음을 알 수 있으며 arXiv 공개일자와 제출날짜가 크게 차이가 나지 않아 큰 수정 없이 제출했을 것으로 보인다.',
 'similarity_score': 0.74130684,
 'most_similar_article': '#redirect 김해 버스 220, 부산 버스 221₩n',
 'is_plagiarism': True}
```

## 실제 Test Run : 나무위키에서 긁은 글

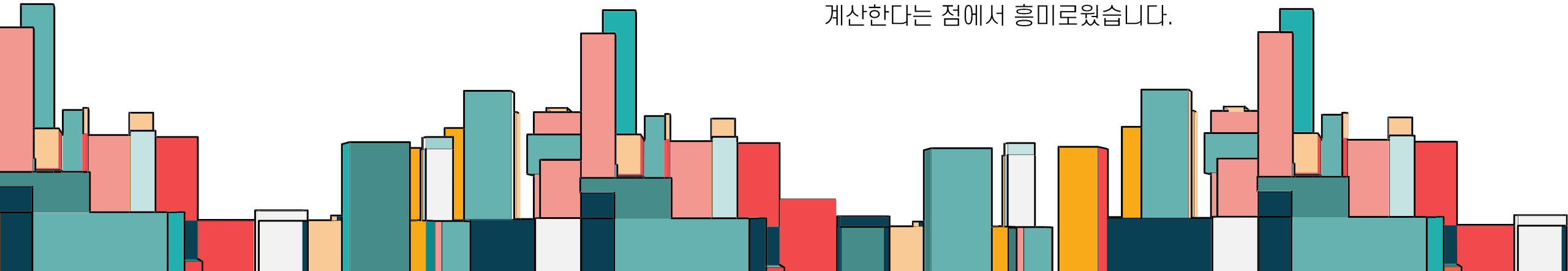
유사도 점수가 0.741로 나무위키에서 표절한 글임을 분류해냈음

## 전체적인 Workflow 구축

실제로 대량의 데이터를 데이터베이스화 시킴으로써 쉽게 데이터를 불러들일 수 있게 하였습니다. 또한, 전체적으로 데이터들을 목적에 맞게 전처리시켜 향후에 다른 task가 생겼을 때에도 그에 맞게 잘 대응할 수 있습니다.

## 사전 학습된 모델 사용

Hugging face에서 사전 학습된 모델 (KoBERT)를 사용해보았고, 이를 통해서 모델을 불러들일 수 있고 목적에 맞게 사용할 수 있다는 점을 알게 되었습니다. 특히, 이번 프로젝트에서는 텍스트 데이터들을 정수 인코딩, 벡터화 시킨 후에 코사인 유사도를 계산한다는 점에서 흥미로웠습니다.



# 5. 한계

```
1 namu = """안녕하세요 17기 우윤규 입니다."""
```

```
1 plagirism_check(namu,vector_index,plagiarism_threshold=0.5)
```

```
{'article_submitted': '안녕하세요 17기 우윤규 입니다.',
 'similarity_score': 0.5927999,
 'most_similar_article': '#redirect N.EX.T/Home₩n',
 'is_plagiarism': True}
```

```
1 plagirism_check(namu,vector_index,plagiarism_threshold=0.5)
```

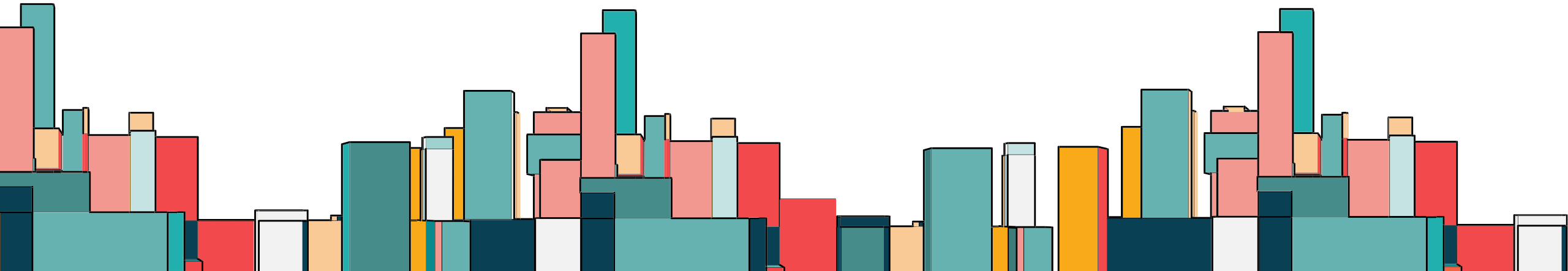
```
{'article_submitted': '전 세계를 떠들썩하게 했던 상온 · 상압초전도체 LK-99의 이슈가 식어가고 있다. 지난달 22일 온라인상에 등장해 화제를 모은 LK-99는 ₩'노벨과학상감₩'이라며 일반 대중에게 관심 받았다. 하지만 논란은 한 달 만에 정리되고 있다. 네이처는 지난 16일 독일 슈투트가르트의 막스플랑크 고체연구소 연구팀의 검증결과를 통해 "LK-99는 초전도체가 아니라 수백만 옴의 저항을 가진 절연체₩'라고 보고했다. 이들은 LK-99에 대해 ₩'강자성과 반자성을 띄지만 부분적인 부양이 충분하지 않았다. 따라서 초전도성의 존재를 배제한다₩'는 결론을 내렸다.',
 'similarity_score': 0.8211528,
 'most_similar_article': '[[SD건담 G제네레이션 시리즈]]의 오리지널 캐릭터이다.₩n₩nNEO에서 추가된 5인방 중의 한 명으로 일단은 [[샤논 마시아스]]와 정반대로 ₩'₩'₩'미형 여성 캐릭터₩'₩'₩'들의 대표를 맡고 있긴 한데, 분홍색 단발머리 이외에는 포인트가 전혀 없다. 모빌파이터에 탑승했을 경우 "수련의 성과를 보여주겠어"라고 말하는 것 이외에는 캐릭터적으로 별다른 특징이 전혀 없기 때문. 다만 말투가 비슷한 여성 지휘관계인 [[시마 가라하우]]보다 훨씬 고압적이고, 특절 대사에서 흥분을 감추지 못해 텐션에 따른 바이브레이션이 들어가는 등 샤논에 버금가는 실전파(내지는 전투광)인 것으로 추측된다.₩n₩n초기 지휘치가 높고 사격이 그럭저럭 성장하기 때문에 곧바로 함장 자리에 앉히는 것도 좋고 사격계 MS에 태우는 것도 괜찮다. 하지만 높은 초기치에 비해 성장치가 낮고 능력치 최종 합계도 그리 높지 않아서 얼마 지나지 않아 곧바로 함장 위치로 고정될 운명. 능력치가 안정되지 못했던 시절의 [[니키 테일러]] 정도쯤 되는 포지션이며 역시나 니키의 아성을 넘지 못하고 NEO와 SEED를 마지막으로 더 이상 등장하지 않는다.₩n₩n₩n[[분류:G제네레이션 시리즈]]',
 'is_plagiarism': True}
```

## 나무위키 Dump 데이터 정제의 어려움

나무위키 전체 데이터를 크롤링하기엔 시간/컴퓨팅 리소스 모두 부족하여 Dump 데이터를 사용했는데, 이때 정제되지 않은 데이터로 인해 약간의 오류 발생

## 나무위키 데이터의 광범위함

토큰화-벡터화를 거친 후 이의 코사인 유사도를 계산하는 과정으로 표절을 분류하는데, 나무위키의 광범위한 텍스트 데이터로 인해 많은 글이 표절 검사에 걸리는 경우 발생



# 감사합니다