JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 24, No. 7, pp. 1493-1498, Jul. 2023

Check for updates

# KoBERT에서 데이터 불균형 문제 경감을 위한 정렬 알고리즘을 이용한 학습 데이터 구성

고 영 서[1] · 최 승 호[2*]
[1]한동대학교 AI·컴퓨터공학심화 학사과정
[2*]몰팩바이오, CTO

# Constructing Training Data Using a Sorting Algorithm to Reduce Data Imbalance in KoBERT

**Young-seo Go[1] · Seoung-Ho Choi[2*]**

[1]Undergraduate's Course, Department of Artificial Intelligence·Computer Science and Engineering, Handong Global University, Gyeongbuk 37554, Korea
[2*]CTO, Molpaxbio, Korea

**[요 약]**

불균형 데이터로 학습한 모델은 소수의 이상 데이터에 대하여 분류를 잘 수행하지 못하게 된다. 정렬 알고리즘을 이용하여 자연어 처리에서의 불균형 문제를 경감하고자 한다. 각 정렬 알고리즘이 소비한 비용을 측정하고 비교하여 데이터 처리에 적합한 알고리즘을 도출한다. 정렬된 데이터는 세가지의 불균형 기준에 따라 전처리하여 학습 데이터를 생성하고, 이를 활용하여 자연어 처리 모델인 KoBERT 모델을 파인튜닝한다. 학습 데이터의 불균형 척도에 따른 정확도, 재현율, 정밀도를 측정하여 데이터 조정의 성능을 평가한다. 제안 방법을 통해 자연어 처리에서 데이터 불균형 문제를 경감하기 위해서 정렬 알고리즘을 적용한 결과 문제를 경감할 수 있음을 확인했다.

**[Abstract]**

Models trained with imbalanced data do not perform well in classification for a small sample of abnormal data. We aim to reduce the imbalance problem in natural language processing by using a sorting algorithm. The computation cost of each sorting algorithm is measured and compared to derive an algorithm suitable for data processing. The sorted data are preprocessed according to three imbalance criteria to create training data and then fine-tuned using KoBERT which is a natural language processing model. The performance of data adjustment was evaluated by measuring accuracy, recall, and precision according to the imbalance scale of the training data. We confirmed that the data imbalance problem in natural language processing could be alleviated by applying the sorting algorithm of the proposed method.

**색인어 :** 자연어 처리, 데이터 불균형, 정렬 알고리즘, KoBERT, 모델 성능

**Keyword :** Natural Language Processing, Data Imbalance, Sorting Algorithms, KoBERT, Model Performance

**http://dx.doi.org/10.9728/dcs.2023.24.7.1493**



**Received** 21 January 2023; **Revised** 14 February 2023
**Accepted** 24 February 2023

***Corresponding Author; Seoung-Ho Choi**

**Tel:** ※ 개인정보 표시제한

**E-mail:** jcn99250@naver.com

## Ⅰ. 서 론

머신러닝 및 딥러닝에서 이루어지는 지도 학습에 사용하는 데이터는 실제의 다양한 분야에서 수집된다. 현실에서 수집하는 데이터는 클래스 불균형 문제를 자주 가지게 된다. 어떤 데이터에서 각 클래스가 가지고 있는 데이터 수의 차이가 큰 경우, 클래스 불균형이 있다고 한다. 예를 들어, 병원에서 특정 질병에 걸린 사람과 걸리지 않은 사람의 데이터를 수집했다고 한다면, 일반적으로 질병에 걸린 사람이 걸리지 않은 사람에 비해 적다. 이와 같이 많은 경우 현실에서 수집한 데이터는 클래스 불균형 문제가 있다.

클래스의 균형은 소수의 클래스에 관심이 많은 경우 즉, 목표가 특정 소수 클래스에 대한 정확한 예측일 때 필요하다. 위의 문단에서 예시로 들었던 병원 데이터를 사용하여 특정 질병에 걸릴 사람을 예측하는 모델을 들 수 있다. 모델이 정확히 예측해야 하는 것은 병에 걸릴 사람으로, 이 모델에서의 중요도는 병에 걸리지 않을 사람에 대한 예측보다 높게 된다. 반면에 오직 전체 예측의 정확도에만 관심이 있는 경우에는 클래스의 균형을 맞출 필요는 없을 것이다. 어떤 클래스이든 정확한 예측을 출력하여 정확도를 높이는 것이 가장 중요한 숙제인 것이다[1]-[3].

데이터 불균형에 대해서는 이미 많은 선행 연구가 진행되어 있으며, 조정을 위한 기법들이 많이 존재한다. 본 논문은 기존의 방법과 정렬 알고리즘을 결합한다. 기존 방법의 이론적 배경을 바탕으로 정렬 알고리즘을 통해 직관적인 전처리를 진행하여, 데이터 불균형 조정을 단순화한다. 지도 학습을 진행하는 사용자는 이 방법을 사용하여, 쉽게 데이터 불균형을 해결하고 그 결과를 근거로 사용하여 데이터 불균형 확인 및 조정의 필요성을 인식할 수 있다. 이를 통해 기존 조정 방법 사용 여부를 결정하고, 추가로 불균형을 개선할 수 있다.

본 논문에서는 소수의 클래스에 대한 예측에 대해서 좋은 정확도를 요구하는 상황에서의 데이터 불균형을 다룬다. 동영상 플랫폼 유튜브에서 동영상 제목을 통해 조회 수의 범위를 예측하는 모델을 통해 데이터의 불균형을 해소할 것이다. 조회 수는 100만 회 이상과 100만 회 미만으로 나눠, 상대적으로 적은 수를 가지는 100만 회 이상의 동영상을 이상 범주 데이터, 100만 회 미만의 동영상을 정상 범주 데이터로 다룬다. 모델의 주된 목표는 동영상 제목을 통해 100만 회 이상의 조회 수가 기록되는지 예측하는 것으로, 이 결과를 통해 적합한 유튜브 동영상 제목에 대한 통찰을 얻을 수 있다. 본 연구의 공헌도는 아래와 같다.

(1) 정렬 알고리즘을 통한 불균형 데이터 조정 및 성능 평가
(2) 정렬 알고리즘 비용 비교 및 효과적인 알고리즘 도출
(3) 유튜브 플랫폼에서 제목을 통한 조회 수 예측 모델 도출 및 개선

## Ⅱ. 관련 연구

불균형 데이터 해결 방법 및 자연어 처리에 관한 선행 연구를 다룬다. 데이터를 조정해서 불균형 데이터를 해결하는 샘플링 기법들을 다루고, 실험에서 사용할 한국어 자연어 처리 도구 KoBERT, 정렬 알고리즘에 대해 다룬다.

언더 샘플링은 많은 데이터 수를 가지는 범주의 데이터를 적은 데이터 수를 가지는 범주의 데이터 수에 맞춰 줄이는 샘플링 방식이다. 언더샘플링은 다수 범주 데이터의 제거로 계산 시간이 감소하나, 데이터 제거로 인한 정보 손실이 발생할 수 있다. 대표적인 언더 샘플링으로는 랜덤 샘플링이 존재한다[4]. 랜덤 샘플링은 다수 범주에서 무작위로 샘플링하여 적은 데이터의 수만큼 무작위로 샘플링 하는 방법이다. 언더 샘플링은 무작위 샘플링을 하기 때문에 각 시도마다 다른 결과를 얻는다는 단점을 갖는다.

오버 샘플링은 적은 수의 데이터를 많은 수의 데이터 수에 맞게 늘리는 샘플링 방식이다. 오버 샘플링은 데이터를 증가시키기 위해 정보의 손실이 없지만, 데이터의 증가로 인해 계산 시간이 증가할 수 있으며 과적합 가능성이 존재한다. 대표

적인 오버 샘플링으로는 리샘플링이 있다[5]. 리샘플링 방법은 적은 수를 가진 범주의 데이터 수가 많은 수를 가진 범주의 데이터 수와 비슷해지도록 증가시키는 방법이다. 이때 소수 범주의 데이터는 무작위로 복제된다. 이 방법은 소수 범주에 대해 과적합이 발생할 수 있다는 단점을 가지고 있다

SK텔레콤에서 제공하는 KoBERT(Korean Bidirectional Encoder Representations from Transformers)는 구글에서 공개한 BERT(Bidirectional Encoder Representations from Transformers)의 한국어 성능을 개선하기 위해 개발되었다[6],[7].

BERT는 말뭉치 데이터를 토큰 임베딩, 세그먼트 임베딩, 포지션 임베딩을 사용하여 인코딩한 후 사전 훈련하여 모델을 생성한다. 생성된 모델은 파인 튜닝을 통해 다른 작업에 대해 파라미터 재조정이 이루어질 수 있으며, 이를 통해 다양한 작업을 수행할 수 있다[8].

KoBERT는 한글 위키 등의 수백만 개의 한국어 문장으로 이루어진 대규모 말뭉치를 학습하고, 한국어의 불규칙한 언어 변화의 특성을 반영하기 위해 데이터 기반 토큰화 기법을 적용하여 높은 성능 향상을 이끌어냈다.

정렬 알고리즘은 버블 정렬 알고리즘, 선택 정렬 알고리즘, 삽입 정렬 알고리즘, 병합 정렬 알고리즘, 퀵 정렬 알고리즘, 힙 정렬 알고리즘, 기수 정렬 알고리즘 총 7개이다. 버블 정렬(Bubble Sort)은 구현이 쉽고 직관적이지만 굉장히 비효율적이며 $O(N^2)$의 시간 복잡도를 가진다. 선택 정렬(Selection Sort) 또한 구현이 쉬운 정렬로 정렬을 위한 비교 횟수는 많지만, 실제 교환하는 횟수가 적어 버블 정렬보다 상대적으로 효율적이다. 시간 복잡도는 $O(N^2)$이다. 삽입 정렬(Insertion Sort)은 최선의 경우 O(N)의 시간 복잡도를 가져서 좋은 효

율성을 가지고 있지만, 최악의 경우 $O(N^2)$의 시간 복잡도를 가지기 때문에 성능의 편차가 굉장히 심한 정렬법이다. 병합 정렬(Merge Sort)은 배열을 분할해가면서 정렬하며 분할 하는 과정에서 log N만큼의 시간이 걸리며, 최종적으로는 O(NlogN) 의 시간 복잡도를 가진다. 다만 추가적인 메모리가 필요하다는 단점이 있다. 퀵 정렬(Quick Sort)은 기준값에 의한 분할을 통해서 구현하는 정렬로 분할에서 logN의 시간이 걸린다. 기준값을 이상적으로 선택한다면 O(N logN)의 시간 복잡도를 갖지만, 최악의 기준값을 고르면 $O(N^2)$의 시간 복잡도를 가진다. 힙 정렬(Heap Sort)은 추가적인 메모리를 필요하지 않으며 항상 시간 복잡도가 O(NlogN)으로 보장된다. 실제 측정했을 때에는 퀵 정렬보다 느리다고 한다. 기수 정렬(Radix Sort) O(N)의 시간 복잡도를 가져 굉장히 속도에 강점을 가지지만, 버킷이라는 추가적인 메모리가 할당되어야하여 메모리를 많이 소비한다.

## Ⅲ. 제안 방법

그림 1은 본 논문에서 제안한 정렬 알고리즘을 적용하여 실험하는 방법이다. 먼저 데이터 수집에 대해서 설명한다.

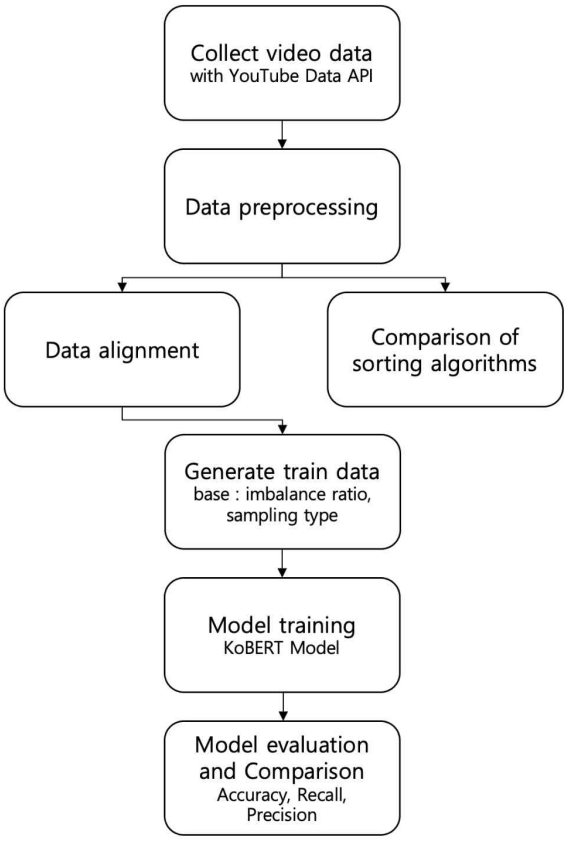


**그림 1.** 제안 방법
**Fig. 1.** Our proposal

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 몰타로 만든 병소 vs 항화탕으로 만든 모주 | 24 | 1985644 | 20416 | - | - | 2022-01-01T10:00:00Z |
| 1 2021년은 어땠나? 2022년은 어떨까? | 24 | 1130874 | 10966 | <null> | <null> | 2022-01-02T10:00:20Z |
| 2 지인계의 냉전시대 | 24 | 735998 | 8056 | <null> | <null> | 2022-01-03T10:00:00Z |
| 3 90년대 매니 여게 틀러보기 | 24 | 872340 | 7453 | <null> | <null> | 2022-01-04T10:00:05Z |

**그림 2.** 원시 데이터 예시
**Fig. 2.** Source data example

YouTube Data API를 통해 유튜브의 영상 데이터를 수집한다. 일반화를 위해 무작위 영상 데이터를 추출하는 것이 이상적이지만, 상대적으로 적은 데이터로 결과를 보기 위해 특정 유튜버로 한정하여 데이터를 수집한다. 유튜버로는 '침착맨'을 선정했다. 침착맨은 약 200만의 구독자를 지닌 유튜버로, 근 3년간 꾸준히 영상 활동이 있었고, 그 분류가 잘 이루어져 데이터 전처리가 용이하다. 또한, 조회수의 분포가 방향성이 있고, 각 동영상의 조회수에 불균형이 존재하여 연구의 실험을 진행하기에 알맞다. 침착맨 채널의 동영상 정보를 사용하여 실험을 진행하기 때문에 실험 모델은 침착맨 채널 동영상의 제목에 따른 조회 수 100만 달성 여부를 예측하는 모델이다. 유튜브 동영상 데이터는 침착맨 채널의 재생목록을 기준으로 수집했다. 각 연도에 따라 '2019년 침착맨 정주행', '2020년 침착맨 정주행', '2021년 침착맨 정주행', '2022년 침착맨 정주행' 재생목록에서 데이터를 수집했으며 그 개수는 이상 데이터를 제외, 각 710개, 432개, 347개, 319개로 총 1,809개이다. 이는 학습이 미리 학습된 KoBERT 모델을 파인 튜닝하기 위한 데이터이기 때문에 그 수는 충분하다. 결측값을 처리를 제외한 총 데이터 중 100만 조회 수 이상 동영상은 1449개, 100만 미만 동영상은 342개이다. 1449:342의 비율은 불균형하기는 하지만, 불균형의 정도가 극단적인 수준은 아니다.

그림 2는 수집한 데이터는 속성으로 동영상의 제목(title), 카테고리(category_id), 조회 수(views), 좋아요 수(likes), 날짜(date)를 포함하고 있으며, 실제로 사용할 데이터는 동영상의 제목과 조회 수다. 다음으로 데이터 전처리에 대해 설명한다.

조회 수 값으로 결측값을 가지는 데이터를 제외한 모든 동영상 데이터를 하나의 데이터프레임으로 모은다. 모인 데이터에서는 원시 데이터의 속성 중 동영상 제목, 동영상 조회 수만을 포함한다. 만들어진 데이터프레임은 조회 수를 기준으로 버블 정렬 알고리즘, 선택 정렬 알고리즘, 삽입 정렬 알고리즘, 병합 정렬 알고리즘, 퀵 정렬 알고리즘, 힙 정렬 알고리즘, 기수 정렬 알고리즘을 사용하여 정렬하고 각 정렬에 걸리는 시간을 측정한다. 총 10회 측정하여 그 평균을 계산하여 조회 수를 기준으로 한 유튜브 데이터를 가장 좋은 속도로 정렬하는 알고리즘을 도출한다. 정렬된 알고리즘을 통해 얻은 데이터프레임의 조회 수를 기준으로 라벨값을 지정한다. 조회 수가 100만 회 이상일 경우 1,100만 회 미만인 경우 0을 라벨링 한다. 그림 3에서는 0으로 라벨링 된 데이터를 보여주는 예시이다. 라벨링 된 데이터를 학습 데이터와 테스트 데이터로 나눈다. 기준에 따라 4개의 학습 데이터와 1개의 테스트 데이터가 만들어진다. 정렬 결과에 따라 라벨로 0을 갖는 데이터, 1을 갖는 데이터를 나눠서, 학습과 테스트 데이터에 골

고루 분포하게 한다. 모든 데이터는 먼저 랜덤하게 섞어주고, 모든 학습 데이터는 라벨을 1로 갖는 데이터의 상위 80%에서 샘플링한 결과와 라벨을 0으로 갖는 데이터의 상위 80%에서 샘플링한 결과를 합친 값에서 샘플링된다. 라벨을 1로 갖는 데이터의 하위 20%와 라벨을 0으로 갖는 데이터의 하위 20%를 합친 데이터가 테스트 데이터가 된다. 이를 통해 학습 및 테스트 데이터는 서로 겹치지 않지만, 각각 라벨로 0과 1이 골고루 분포한 데이터가 된다. 4가지 학습 데이터의 샘플링 방식은 다음과 같다. 먼저 기본값은 라벨1 80%, 라벨0 80%를 그대로 사용한다. 균형 데이터는 2가지로 나뉘는데, 하나는 언더 샘플링(랜덤 샘플링), 다른 하나는 오버 샘플링(리샘플링)을 사용한다. 오버 샘플링은 라벨1에서 라벨0의 수만큼만 무작위로 샘플링한 결과와 라벨0을 결합하여 사용한다. 언더 샘플링은 라벨0에서 라벨1의 수만큼 무작위로 샘플링한 결과와 라벨1을 결합하여 사용한다. 마지막 학습 데이터는 불균형 샘플링인데, 불균형의 정도가 크지 않았던 기존의 학습 데이터에 인위적으로 강한 불균형을 준다. 라벨1은 기존의 10분의 1만 사용하고 라벨0은 그대로 사용한다. 이 데이터는 다른 데이터와 비교를 위해 사용된다. 만들어진 데이터는 모델을 학습할 때 사용되게 된다. 표 1은 수집된 데이터에서 100만 회 이상 조회 수와 미만 조회 수 영상을 기준으로 샘플링한 데이터 수이다.

KoBERT에서 제공하는 KoBERT 모델을 불러온 후, 4가지 학습 데이터를 각각 학습하여 모델의 가중치를 저장한다. 정답 데이터가 될 클래스를 2가지로 하고, 기존 모델에 추가로 파인 튜닝(Fine Tuning)하여 각 학습 데이터에 적합한 모델을 생성한다. 하이퍼 파라미터로는 제목의 최대 길이 32, 배치 사이즈 32, 최대 에폭 수 30, 학습률은 5 × 10-5을 사용했다. 학습 환경은 그래픽 카드 Geforce RTX 2080 Ti를 사용했고, 리눅스 버전 18.04에서 실행했다.


**그림 3.** 학습 및 테스트 데이터 예시
**Fig. 3.** Train and test data example

**표 1.** 데이터별 구성 동영상 수
**Table 1.** Configuration videos and per data

| Data | Over 1M | Under 1M |
|---|---|---|
| (Default) Train | 273 | 1159 |
| Balanced Train (Over sampling) | 1159 | 1159 |
| Balanced Train (Under sampling) | 273 | 273 |
| Imbalanced Train | 27 | 1159 |
| Test | 69 | 290 |

## Ⅳ. 실험 결과

시간 복잡도 측면에서 봤을 $O(N^2)$의 평균 복잡도를 가지는 삽입, 선택, 버블 정렬이 비슷하게 느린 성능을 보이고, O(NlogN)인 퀵, 병합, 힙 정렬이 그다음, O(N)의 복잡도를 가지는 기수 정렬이 가장 적은 시간이 걸릴 것으로 예상되었다. 실제 데이터 전처리 시 사용한 각 정렬 알고리즘별 시간 측정 결과는 표 2와 그림 4와 같다. 예상과 같이 전체적으로 O(N), O(NlogN), $O(N^2)$의 순서로 적은 시간을 기록했다. 다만 O(NlogN)의 복잡도를 가진 퀵 정렬이 기수 정렬보다 미세하게 더 좋은 결과를 기록했다. 그다음으로는 기수, 병합, 병합, 힙, 선택, 삽입, 버블 정렬 순으로 짧은 시간을 기록했다. 결과적으로 가장 짧은 시간 동안 정렬을 실행한 퀵 정렬이 가장 적은 비용으로 작업을 수행했다. 따라서 침착맨 유튜브 동영상 제목-조회 수 데이터를 가장 빠르게 정렬하는 알고리즘은 퀵 정렬이고, 정렬 시 이를 채택하는 것이 가장 적은 비용을 소모한다. 더하여 일반화된 유튜브 동영상을 조회 수를 기준으로 정렬하는 작업에서도 퀵 정렬이 가장 우세한 결과를 보여줄 것이라고 예상할 수 있다.

**표 2.** 정렬 알고리즘별 10회 평균 경과 속도
**Table 2.** Average elapsed rate of time per sorting algorithms

| Sorting | Time(sec) |
|---|---|
| Quick | 0.00158 |
| Radix | 0.00185 |
| Merge | 0.00346 |
| Heap | 0.00436 |
| Selection | 0.05617 |
| Insertion | 0.06947 |
| Bubble | 0.10267 |

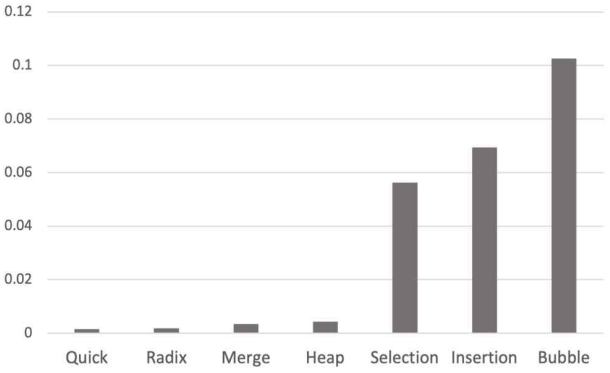


**그림 4.** 정렬 알고리즘별 10회 평균 경과 속도 그래프
**Fig. 4.** Graph of average elapsed rate of time per sorting algorithms

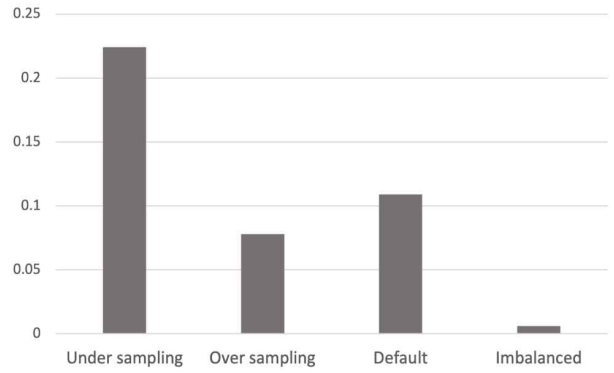


**그림 5.** 데이터별 라벨 1에 대한 재현율
**Fig. 5.** Recall of label 1 per data

**표 3.** 학습 데이터별 모델 정확도, 재현율, 정밀도
**Table 3.** Accuracy, recall, precision, per train data

| Data | Accuracy | Recall (0) | Recall (1) | Precision (0) | Precision (1) |
|---|---|---|---|---|---|
| Under sampling | 0.671 | 0.775 | 0.224 | 0.925 | 0.343 |
| Over sampling | 0.773 | 0.921 | 0.078 | 0.840 | 0.400 |
| Default | 0.790 | 0.890 | 0.109 | 0.868 | 0.442 |
| Imbalanced | 0.808 | 0.993 | 0.006 | 0.810 | 0.400 |

각각 다른 샘플링을 하여 다른 불균형 정도를 준 각 학습 데이터의 분류 결과는 표 3과 같다. 전체 테스트 정확도를 기준으로 했을 때, 오버 샘플링, 기본값, 불균형 데이터가 약 80%의 정확도를 기록했으며, 언더 샘플링은 67%의 낮은 정확도를 기록했다. 언더 샘플링은 총 학습 데이터의 수가 약 500개로 1300개 이상인 나머지 데이터에 비해 낮은 전체 정확도를 기록했다고 해석한다. 불균형 데이터는 다른 80%를 기록한 다른 데이터들에 비해 데이터 수가 적긴 하지만 그 차이가 크지 않다. 오버 샘플링은 데이터 수가 더 많음에도 3% 정도 낮은 정확도를 기록했는데, 같은 데이터를 복제하여 사용했기 때문에 그 의미가 크지 않고 오히려 과적합이 되었을 것으로 볼 수 있다. 100만 회 이상의 조회 수를 기록할 동영상의 제목을 판단하는 것이 모델의 목표인 만큼, 주목해서 볼 점은 라벨 1에 대한 재현율이며 그 결과는 그림 5에서 확인할 수 있다. 샘플링별 라벨 1에 대한 재현율은 언더 샘플링, 기본값, 오버 샘플링, 불균형 데이터 순으로 높은 수치를 기록했다. 언더 샘플링은 그다음 높은 기본값의 2배 이상 좋은 재현율을 기록했다. 데이터가 약 3배 적음에도 압도적인 재현율을 보이며 이 모델이 라벨 1에 대해 특별히 훌륭한 예측을 보인다고 할 수 있다. 데이터 수를 줄이면서도, 클래스 간의 비율을 비슷하게 맞추는 것만으로 소수 범주에 대한 재현율을 크게 높일 수 있다. 반면 데이터에서의 라벨 0과 1의 비율을 거의 40:1로 만든 불균형 데이터는 높은 정확도에도 불구하고 소수 범주인 라벨 1을 거의 예측하지 못했다. 오버 샘플링은 라벨 간의 비율은 같으나, 특별히 좋은 재현율을 기록하지 못했다. 언더 샘플링은 적은 데이터 탓에 상대적으로 낮은 정확도를 기록했으나 불균형을 해결하며 소수 범주에 대한 높은 재현율을 기록했다. 오버 샘플링은 불균형은 해결했으나, 단순히 데이터를 복제하는 리샘플링 기법을 채택하여, 오버피팅이 일어나 낮은 정확도 및 재현율을 기록한 것으로 보인다. 불균형 정도를 인위적으로 준 데이터는 소수 범주의 라벨 데이터를 거의 예측하지 못하여, 높은 정확도를 의미 없는 수치로 만들었다. 높은 정확도를 요구하는 목적에 부합하려면 5:1 정도의 비율을 가지며, 라벨 1의 재현율도 상대적으로 좋은 기본값을 사용하는 것이 좋다. 하지만 모델의 목적이 100만 조회 수 이상 동영상의 제목을 판단하는 것이기 때문에 언더 샘플링이 가장 적합하다는 결과들을 얻을 수 있다. 하지만 낮은 정확도를 개선하기 위해, 추가적인 데이터를 수집하거나 샘플링 과정에서 없애는 데이터의 수를 줄이는 과정이 필요하다.

## V. 결 론

유튜브 동영상 자연어 데이터를 사용한 분류 모델에서의 정렬 알고리즘을 통한 불균형 데이터 조정이다. 이에 가장 적합한 알고리즘을 찾고, 조정된 데이터 학습 결과의 성능을 평가하는 것이 실험을 통해 얻고자 한 것이다. 정렬 알고리즘은 퀵, 기수, 병합, 병합, 힙, 선택, 삽입, 버블 정렬 순으로 좋은 성능을 기록했으며, 따라서 조회 수를 기준으로 한 정렬을 진행할 때는 퀵 정렬을 사용하는 것이 가장 좋다는 결과를 얻었다.

정렬된 알고리즘을 기준으로 인위적인 샘플링을 거쳐 불균형 정도가 다른 서로 다른 학습 데이터를 만들었다. 불균형이 큰 데이터는 소수 범주에 대해서 극악의 재현율을 기록했으며, 리샘플링(오버샘플링)을 사용하여 균형을 맞춘 데이터는 오버피팅되어 조금 떨어지는 정확도와 재현율을 보였다. 랜덤 샘플링(언더 샘플링)을 사용하여 불균형을 해소한 모델은 적은 데이터 수 탓에 낮은 정확도를 기록했으나, 소수 범주에 대한 예측에 압도적으로 특화된 모델을 만들었다. 소수 범주인 100만 조회 수 이상 동영상에 시선을 둔 작업인 만큼 언더 샘플링이 가장 좋은 결과를 보이며, 데이터 불균형을 조정을 잘 해결했다. 하지만 부족한 정확도는 데이터 추가 수집, 추가 샘플링 등의 방법을 통해 개선할 필요는 있다. 결과와 같이 단순한 정렬 알고리즘을 사용한 언더 샘플링을 사용하여 데이터 불균형을 해소할 수 있다. 실제 데이터 분석에서 사용되는 불균형 해소가 복잡한 수학적 원리 및 구현을 가지고 있다는 점을 생각한다면, 직관적이고 단순한 해당 방법은 고도화된 불균형 조정 방법을 사용하기 전 사전 조정 방법으로 사용할 수 있을 것이다. 그뿐 아니라 해당 방법을 사용함으로써 수학적 지식이 부족한 데이터 분석 초심자에 대해서도 직관적으로 데이터 불균형과 샘플링에 대한 개념을 이해시킬 수 있다.

해당 논문은 유튜브 데이터, 특히 분포가 다양하고, 잘 정리되어있던 특정 유튜버의 데이터에 대해 실험을 진행했다. 따라서 전체 유튜브 데이터 및 넓은 범위의 자연어 처리 분류 문제에서도 이상적인 성능을 보일 것이라 확실하기는 어렵다. 더 일반화되고 광범위한 분야에 활용하기 위해서는 더 다양하고 많은 데이터에 대한 실험이 필요하다. 논문에서는 가장 간단한 수준의 샘플링(랜덤 샘플링, 리샘플링)에 정렬 알고리즘을 사용하여 단순한 불균형 조정을 이뤄냈다. 하지만 데이터 불균형 해소 기법에는 더 훌륭하고 고도화된 샘플링 및 기법들이 존재한다. 이들에 대해서도 더 단순하고 직관적으로 데이터 불균형을 해결할 수 있다면 한 층 더 강력한 사전 불균형 조정 기법으로 사용할 수 있을 것으로 전망한다.

## 감사의 글

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학지원사업의 연구결과로 수행되었음(2017-0-00096)

## 참고문헌

[1] H. Haibo and E. A. Garcia, “Learning from Imbalanced Data,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 21, No. 9, pp. 1263-1284, September 2009. https://doi.org/10.1109/TKDE.2008.239

[2] Y. Sun, A. K. C. Wong, and M. S. Kamel, “Classification of Imbalanced Data: A Review,” *Interational Journal of Pattern Recognition and Artificial Intelligence*, Vol. 23, No. 04, pp. 687-719, 2009. https://doi.org/10.1142/S0218001409007326

[3] N. Chawla, N. Japkowicz, and A. Kotcz, “Special Issue on Learning from Imbalanced Data Sets,” *ACM SIGKDD Explorations Newsletter*, Vol. 6, No. 1, pp. 1-6, June 2004. https://doi.org/10.1145/1007730.1007733

[4] S.-J. Yen and Y.-S. Lee, Under-Sampling Approaches for Improving Prediction of the Minority Class in an Imbalanced Dataset,” in *Intelligent Control and Automation*, Berlin: Springer, pp. 731-740, 2006. https://doi.org/10.1007/978-3-540-37256-1_89

[5] C. Drummond and R. C. Holte, “C4.5, Class Imbalance, and Cost Sensitvity: Why Under-Sampling beats Over-Sampling,” Workshop on Learning from Imbalanced Dataset II, 2003.

[6] KoBert [Internet]. Available: https://sktelecom.github.io/project/kobert/

[7] KoBert Github [Internet]. Available: https://github.com/SKTBrain/KoBERT

[8] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “Bert:Pre-Training of Deep Bidirectional Transformers for Language Understanding,” arXiv, 2018. https://doi.org/10.48550/arXiv.1810.04805

**고영서(Yeong-seo Go)**

2022년 : 한동대학교 AI 컴퓨터공학심화 학사과정

※관심분야 : 딥러닝

**최승호(Seoung-Ho Choi)**

2018년 : 한성대학교 전자정보공학과(공학사)
2020년 : 한성대학교 전자정보공학과(공학석사)
2021년～현 재: 한성대학교 기초교양학부 시간강사
2022년～현 재: 몰팩바이오 CTO
※관심분야 : 딥러닝