# 뉴스의 제목과 본문 내 낚시 탐지

## : 특화 데이터셋 학습 및 앙상블 기법을 통해서

우영석
*인공지능융합학과*
*성균관대학교 소프트웨어융합대학*
*2024711029*

***Abstract*—대형 언어 모델이 대중적으로 사용됨에 따라 언론사는 낚시성 기사를 양산하고 있다. 게다가 낚시성 뉴스는 제목을 통한 클릭 유도와 본문에서 구매 유도를 하는 등 다양한 형태로 존재한다. 하지만 낚시성 기사 탐지는 주로 제목과 본문 간의 불일치를 탐지하는 방향으로 연구되어 왔다. 따라서 본 프로젝트는 본문 내 불일치도 함께 탐지할 수 있도록 새로운 모델링을 제안한다. 이는 특화된 데이터셋으로 학습한 모델들을 앙상블하는 기법이다. 이는 낚시성 기사 중 많은 수를 탐지하는 재현율(recall)에서 높은 성능을 보여주었다. 하지만 형태소 분석기와 모델 선정에 추가적인 연구가 이뤄진다면 성능을 향상시킬 수 있는 여지가 있다.**

***Keywords*—*낚시성 기사, 제목 낚시, 본문 낚시, 앙상블 기법, LightGBM***

## I. Introduction

최근 몇 년 동안, 자연어 처리 및 기계 학습 기술의 발전으로 인해 대량의 정보가 빠르게 생성되고 공유되고 있다. 특히 ChatGPT 와 같은 대형 언어 모델이 본격적으로 대중들에게 소개되었고, Youtube 등을 기반으로 개인이 정보 생산의 중심이 되었다. 언론사도 인터넷 포털 뉴스를 통해 이전보다 훨씬 많은 양의 컨텐츠를 생산하고 있으며, 가짜 뉴스 및 낚시성 기사[1] 또한 더욱 양산되고 있다. 따라서 낚시성 기사 탐지 문제는 데이터 공학에서 가장 시급한 문제 중 하나이다.[4] ai-hub 의 2023 년 12 월에 "낚시성 기사 탐지 데이터"[1]가 정식 공개된 것도 이 문제의 시의성을 드러내는 구체적 사례 중 하나이다. 특히 경제 및 정치 분야에서의 낚시성 기사는 대중의 중요한 의사 결정에 영향을 미칠 수 있다. 경제 분야에서는 투자 결정에 영향을 주고, 정치 분야에서는 투표 및 정책에 대한 혼란을 야기한다.

최근 3 년간의 논문 중에서 가짜 뉴스 혹은 클릭베이트 탐지의 키워드로 검색되는 논문 53 편을 분석한 결과, 가짜 뉴스의 경우 제목과 본문 간의 불일치를 탐지하는 경우가 많았고, 낚시성 기사에 대한 접근법 역시 유사했다.[2] 그러나 현실에는 더 다양한 종류의 낚시성 기사가 존재한다. 예를 들어, 상품, 주식이나 부동산 홍보를 위해 뉴스인 것 마냥 작성된 홍보성 기사들이 있다. 따라서 본 연구는 이러한 학계와 현실의 간극을 메우는 작업을 수행하고자 한다. 다시 말해, 제목을 통한 낚시는 물론 본문에서 홍보를 위한 정보를 끼워 넣는 낚시도 탐지해내려고 한다.

## II. Method

본 프로젝트를 진행하기 위해 공부한 내용으로는 주로 선행 연구 및 다양한 모델의 이해가 있었다. 특히, 활용한 데이터셋의 '구축 활용 가이드라인'을 정독하여 데이터셋이 어떻게 구축되었는지를 명확히 이해하기 위해 노력했다. 이를 통해 낚시성 기사는 1 세부: 클릭을 유도하는 제목을 담은 데이터와 2 세부: 판매 정보를 노출하는 본문을 담은 데이터로 가공되었다는 것을 파악했다. 이러한 내용을 이해한 채로 선행 연구를 탐색했을 때, 홍보성 기사 탐지에 대한 선행 연구가 부족하다는 것[2]을 알게 되었고 프로젝트의 방법론을 구상하였다. 직접 KCI(한국학술지인용색인)[3]을 통해 '가짜 뉴스'[3]를 키워드로 검색했을 때와 다르게 '홍보'를 기준으로 재검색했을 때 공학 논문이 존재하지 않았다.

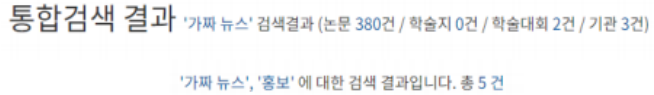


그림 1. KCI 키워드 검색 결과

수집한 데이터셋은 ai-hub 의 "낚시성 기사 탐지 데이터"로 구성은 다음과 같다.

| 세부구분 | | 가공패턴유형 |
|---|---|---|
| 1세부 | 제목과 본문의 불일치 기사 | 의문 유발형(부호) |
| | | 의문 유발형(은닉) |
| | | 선정표현, 사용형 |
| | | 속어/품위 사용형 |
| | | 사실 과대 표현형 |
| | | 의도적 주어 왜곡형 |
| | | 소 계 |
| 2세부 | 본문의 도메인 일관성 부족 기사 | 상품 판매정보 노출 광고형 |
| | | 부동산 판매정보 노출 광고형 |
| | | 서비스 판매정보 노출 광고형 |
| | | 의도적 상황 왜곡/전환형 |
| | | 소 계 |

그림 2. 낚시성 기사 탐지 데이터셋 구성

[1] 이하 가짜 뉴스와 낚시성 기사를 통칭할 시 낚시성 뉴스로 칭하겠다.

[2] 서론에서 밝혔듯이 제목과 본문 간의 불일치를 탐지하려는 시도가 많았다.[1] 따라서 데이터셋의 2 세부와 같은 낚시성 기사를 탐지하려는 새로운 접근이 더 필요하다.

[3] 낚시성 기사로 검색하면 총 10 건밖에 출력되지 않는다.

각 세부는 같은 낚시성 기사라도 완전히 다르게 가공된 데이터이기 때문에 특징 추출 시에 큰 차이를 보일 것으로 예상된다. 이에 각 데이터 세트를 별도로 학습한 모델을 구축하여 voting 방법을 사용하는 것이 효과적일 것으로 판단했다. 또한, 본 프로젝트의 목표는 '낚시성 기사를 최대한 많이 탐지해내는 것이기 때문에 전체적인 정확도보다는 재현율(recall)이 가장 중요한 평가요소'[3] 로 고려되었다.

방법론의 개괄은 다음과 같다.

1) Data Load

1 차적으로 AI Hub[1]에서 제공하는 데이터 통계와 데이터 구축 활용 가이드라인을 참고하여 데이터셋 구조를 파악하려고 했다. 특히 직접 데이터를 출력함으로써 1 세부과 2 세부의 데이터 간의 차이를 파악했다. 이를 바탕으로 각 세부의 json 파일로부터 필요한 정보를 추출했다. 최종적으로 프로젝트에 사용한 데이터셋은 경제, 정치 카테고리의 기사로 구성했고, 8:2 비율로 학습 데이터와 테스트 데이터로 구성했다.

2) EDA

전체 데이터셋과 각 세부 데이터셋의 클래스 분포를 확인하였고 기사 텍스트를 직접 출력하여 확인해보는 작업을 거쳤다.

3) Data Preprocessing

한글과 영어를 제외한 문자는 제거하였다. 뉴스 기사이므로 영어도 고유명사 등 필요한 정보일 가능성이 높아 제거하지 않았다. 또한, 형태소 분석기는 spaCy 의 "ko_core_news_sm"과 Kiwi 를 비교하여 성능이 더 뛰어난 Kiwi 선택하였다. 최종적으로 Kiwi 를 통해 낚시성 기사의 맥락을 파악하는 데 필요한 특정 품사를 추출하여 모델 학습에 활용하였다.

4) Feature Extraction & Modeling

TF-IDF 임베딩을 적용하여 뉴스 기사의 특징을 추출하였다. 이를 통해 모델이 자연어 데이터를 학습할 수 있도록 하였으며, TF-IDF 의 하이퍼파라미터는 n-gram 범위를 unigram 부터 trigram 까지 설정하였다.

모델링은 전체 데이터셋을 학습한 모델, 세부 데이터셋을 학습한 모델 둘을 앙상블하여 majority voting 을 실시하였다. 이때 모델은 Logistic Regression 과 트리 기반 부스팅 모델을 고려했다. 트리 기반 모델은 학습 속도를 고려해 XGBoost 대신 LightGBM 을 선택하였다. 하이퍼파라미터 튜닝은 공식 Documentation[6]에 따라 LGBM 의 중요한 파라미터를 선택하고, GridSearCV 를 실시했다.

5) Results Report

낚시성 기사에 해당하는 클래스 0 을 잘 분류하는 것이 중요하므로 confusion matrix 를 그렸다. 또한 classification report 를 통해 클래스 0 에 대한 재현율을 중심으로 성능을 확인하였다. 이때 전체 데이터셋을 학습한 모델, 각 세부 데이터셋을 학습한 모델의 성능도 같이 파악함으로써 앙상블 기법이 미친 영향을 이해할 수 있도록 하였다.

## III. Result

1) Data Load

데이터셋의 구조는 다음과 같다. 원천 데이터와 라벨링 데이터, 그 내부에 학습 및 검증을 위한 데이터로 구성되어 있으며, 각 가공 유형과 기사 카테고리 별로 구분된 zip 파일들이 존재한다. zip 파일의 압축을 풀면 각 기사에 대한 모든 메타데이터와 기사 데이터가 담긴 json 파일을 얻을 수 있다. 1 세부의 가공된 데이터는 제목에 클릭을 유도하는 내용이 담겨 있으므로 본문은 원문 데이터를 추출했다. 반면 2 세부는 본문 내에서 판매 정보가 추가되는 식으로 가공이 이뤄졌기 때문에 한 문장씩 문자열을 통합해서 추출했다. 결론적으로 162,966 개의 학습을 위한 데이터셋과 20,371 개의 테스트를 위한 데이터셋을 구축하였다.

2) EDA

전체 데이터셋과 각 세부 데이터셋의 클래스 간 비율 차이는 거의 없었다. 또한 직접 기사 데이터를 확인한 결과, 문어체로 잘 작성된 자연어로 구성되어 있었다.

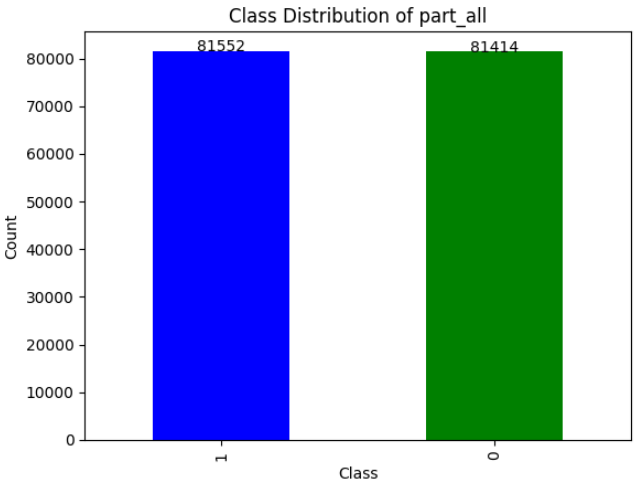


그림 3. 전체 데이터셋의 클래스 분포

3) Data Preprocessing

spaCy 와 Kiwipiepy 의 한국어 형태소 분석기는 장단점이 존재하였다. "ko_core_news_sm"은 표제어 추출을 지원하지만 표제어 처리가 각 토큰의 의미 형태소만 남기지 않으므로 아쉬웠다. 반면, Kiwi 는 표제어 처리 기능을 제공하지 않았고 품사 태깅을 통한 일부 품사 추출이 가능했다. 같은 데이터에 대한 전처리 예시는 다음과 같다.

| | newsTitle | newsContent |
|---|---|---|
| 0 | 공동+주택 하자분+쟁 조정 본격화 | 최종+구 금융+위원 회 의 안착 정부+가 기금+을 위원+장은 사회적금융포럼추진 +위원+... |
| 1 | 강자 코오롱 글로+벌 드 리+미 국내 론칭 | 요즘 형 사회 공헌 활 동+을 회사+가 회사 +를 기업에+서 능력 +을 공익 형 사회 공... |

그림 4. spaCy 전처리 예

| | newsTitle | newsContent |
|---|---|---|
| 0 | 공동 주택 하자 분쟁 조정 본격 | 최종 구 금융 위원회 위원장 임팩트 금융 사회 금융 안착 차원 정부 직접 기금 만들... |
| 1 | 로봇 청소기 강자 코오롱글로벌 드리미 W 국내 론칭 | 요즘 프로보노 사회 공헌 활동 펴 회사 늘 회사 경영 기업 갖추 능력 공익 차원 무... |

그림 5. Kiwipiepy 전처리 예

두 형태소 분석기가 장단점이 있었기 때문에 예측 모델의 성능을 기반으로 선택하였다. 그 결과 spaCy 전처리를 거친 경우에 비해 Kiwipiepy 전처리를 거친 경우가 동일 모델에 대해 훨씬 나은 성능을 보였다.

```
LGBMClassifier (TF-idf) - Classification Report:
validation_accuray: 0.6640461819055591
test_accuracy: 0.7178832654263414
```

그림 6. spaCy 전처리의 LGBM 성능

```
LGBMClassifier (TF-idf) - Classification Report:
validation_accuray: 0.7217145770689051
test_accuracy: 0.7742869765843601
```

그림 7. Kiwipiepy 전처리의 LGBM 성능

최종적으로 Kiwi 형태소 분석기를 통해 뉴스 기사의 “일반명사/고유명사, 동사, 형용사, 일반 부사, 감탄사, 어근, 알파벳”을 정제하여 모델 학습에 활용하였다.

4) Feature Extraction & Modeling

spaCy 의 한국어 형태소 분석기로 전처리했을 때 CountVectorizer 와 TfidfVectorizer 로 인한 LogisticRegression 의 성능 차이를 실험했다. 그 결과는 아래와 같다.

| | CountVectorizer | TFidfVectorizer |
|---|---|---|
| 검증 정확도 | 0.6346 | 0.6531 |
| 예측 정확도 | 0.6804 | 0.6954 |

위 결과에 따르면 “낚시성 기사 탐지 데이터셋”은 TF-idf 임베딩을 거쳤을 때 모델의 성능이 더 향상되었다. 따라서 이후 모든 임베딩은 TFidfVectorzier 를 통해 이루어졌다.

본 프로젝트는 앙상블 기법을 사용하였지만 보다 좋은 성능을 보여준 LightGBM 을 동일하게 사용하였다. 대신 LightGBM 을 각각 특화 데이터셋으로 학습시켜 Majoritiy Voting 을 실시했다. Logistic Regression 은 LGBM 보다 낮은 성능을 보였기 때문에 제외했다.

```
LogisticRegression (Bow) - Classification Report:
validation_accuray: 0.6531299000905214
test_accuracy: 0.6954494133817682
```

그림 8. 명사추출 데이터에 대한 Logistic Regression 성능

```
LGBMClassifier (TF-idf) - Classification Report:
validation_accuray: 0.6640461819055591
test_accuracy: 0.7178832654263414
```

그림 9. 명사추출 데이터에 대한 LightGBM 성능

하이퍼파라미터 튜닝은 기존에 베이지안 최적화로 hyperopt 라이브러리를 고려했지만 학습 속도가 매우 느렸고 성능 향상이 기대만큼 이뤄지지 않아 폐기했다. 대신 GridSearCV 를 통해 높은 정확도를 위한 num_leaves, max_depth 와 overfitting 방지에 핵심적인 min_data_in_leaf 의 최적인 조합을 탐색했다.

하이퍼파라미터 튜닝을 거친 후, 전체 낚시성 기사를 학습한 모델의 가중치를 0.5 로 각 세부 낚시성 기사를 학습한 모델의 가중치를 0.25 로 설정하여 Weighted Majority Voting 을 실시했다. 전체 낚시성 기사를 학습한 모델이 개별적으로 예측을 진행했을 때, 0 클래스에 대해 0.76 의 재현율을 보였다. 앙상블 기법을 통한 예측을 진행했을 때는 0 클래스에 대해 0.81 의 재현율을 보였다. 즉, 특화 데이터셋 기반 앙상블 기법이 낚시성 기사를 더 많이 탐지해낼 수 있었다.

5) Results Report

단일 모델과 앙상블 모델링의 성능 차이를 명확하게 알 수 있게 Confusion Matrix 를 통해 시각화하였다. 또한 전반적인 성능을 수치적으로 이해하기 위해 Classification Report 도 출력하였다.

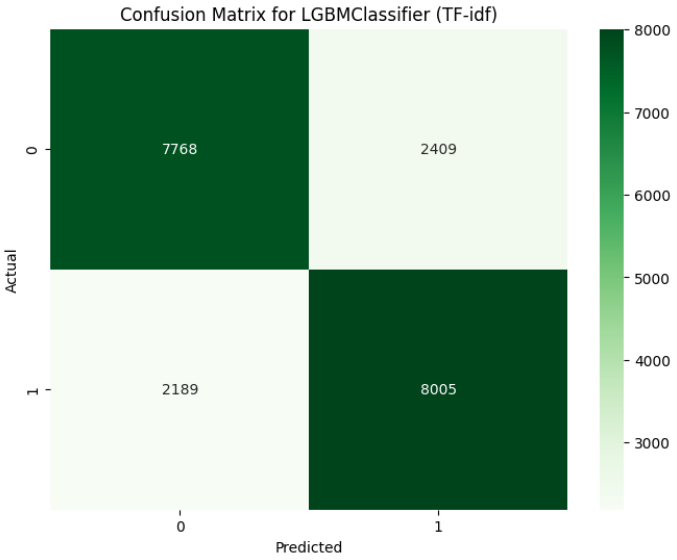


그림 10. 단일 LGBM 의 Confusion Matrix

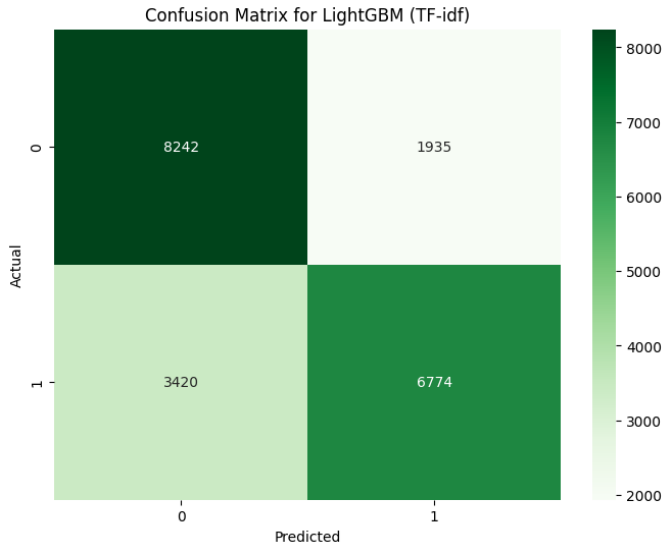


그림 11. 앙상블 모델링의 Confusion Matrix

앙상블 모델링을 진행하였을 때, 전체 데이터셋을 학습한 단일 모델보다 클래스 0 에 대해 474 개(10,194 개의 약 5%) 더 많이 예측할 수 있었다.

```
=== Classification Reports ===
LGBMClassifier (TF-idf) - Classification Report:
validation_accuray: 0.7217145770689051
test_accuracy: 0.7742869765843601
              precision    recall  f1-score   support

           0       0.78      0.76      0.77     10177
           1       0.77      0.79      0.78     10194

    accuracy                           0.77     20371
   macro avg       0.77      0.77      0.77     20371
weighted avg       0.77      0.77      0.77     20371
```

그림 12. 단일 LGBM 의 Classification Reports

```
weight_majority_vote - Classification Report:
              precision    recall  f1-score   support

           0       0.71      0.81      0.75     10177
           1       0.78      0.66      0.72     10194

    accuracy                           0.74     20371
   macro avg       0.74      0.74      0.74     20371
weighted avg       0.74      0.74      0.74     20371
```

그림 13. 앙상블 모델링의 Classification Reports

위 두개의 Classification Report 를 비교하면, 확실히 본 프로젝트가 목표한 클래스 0에 대한 재현율을 향상했다는 것을 알 수 있다.

## IV. Conclusion

본 프로젝트는 시의성이 있는 문제인 낚시성 기사 탐지를 위해 머신러닝을 진행하였다. 기존 연구가 제목과 본문 간의 불일치 탐지를 중심으로 발전해왔다는 점과 실제 낚시성 기사에는 본문 내에서 불일치가 일어나는 사례도 존재한다는 점에서 착안해 특화 데이터셋 기반의 앙상블 모델링을 기획했다. 최종적으로 프로젝트 기획의도에 맞게 낚시성 기사에 대한 재현율을 높이는 모델링을 구현함으로써 연구 목표를 성취했다. 본 프로젝트는 데이터셋에 대한 철저한 이해와 선행 연구에 대한 깊은 탐색이 연구 기획의 핵심적인 과정임을 보여준다. 또한 특정 클래스의 탐지가 중요한 경우에 재현율을 높이기 위해 앙상블 기법이 효과적이라는 것을 제안하고 증명했다. 부가적으로 자연어 데이터에 대한 전처리의 중요성도 부각되었다. 임베딩 방식, 앙상블 기법과 하이퍼파라미터 튜닝 등 많은 기법들이 성능 향상에 일조했지만 텍스트 전처리에 따른 성능 차이가 가장 컸다. 즉, 자연어 데이터를 기반으로 인공지능을 구축할 때, 형태소 분석과 전처리에 가장 심혈을 기울여야 할 것이다.

본 프로젝트는 형태소 분석기를 다양하게 적용하여 성능 실험을 실시하거나 딥러닝 모델도 연구에 적용해본다면 더 좋은 모델 성능을 보고할 여지가 많다. 그리고 본 프로젝트의 모델링이 재현율을 향상하는 데 성공했지만 전반적인 예측 성능은 저하되었기에 다른 모델링도 실시해볼 필요가 있다. 예를 들어 다양한 인공지능 모델을 채택하고 전체 데이터셋을 학습시켜 Voting 을 실시한다면 정확도와 재현율 모두 높이는 방법이 될 수 있다.

## V. References

[1] *낚시성 기사 탐지 데이터*. (2024 년 5 월 5 일). AI Hub. https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&dataSetSn=71338

[2] 좌희정, 오동석, & 임희석. (2019). 자동화기반의 가짜 뉴스 탐지를 위한 연구 분석. *한국융합학회논문지*, 10(7), 18-19.

[3] 허성완 & 손경아. (2016). 낚시성 인터넷 신문기사 검출을 위한 특징 추출. *Journal of KIISE*, 43(11), 1214.

[4] Jeong-Jae Kim, Sang-Min Park & Byung-Won On. (2023). A Pooled RNN-based Deep Learning Model based on Data Augmentation for Clickbait Detection. *한국정보기술학회논문지*, 21(4), 46, 10.14801/jkiit.2023.21.4.45

[5] *KCI 통합검색*. (2024 년 5 월 5 일). KCI(한국학술지인용색인). https://www.kci.go.kr/kciportal/main.kci

[6] *LightGBM Parameters Tuning*. (2024, May 5). LightGBM documentation. https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html