

# Preprocessing Technique for Malicious Comments Detection Considering the Form of Comments Used in the Online Community

Kim Hae Soo<sup>†</sup> · Kim Mi Hui<sup>††</sup>

## ABSTRACT

With the spread of the Internet, anonymous communities emerged along with the activation of communities for communication between people, and many users are doing harm to others, such as posting aggressive posts and leaving comments using anonymity. In the past, administrators directly checked posts and comments, then deleted and blocked them, but as the number of community users increased, they reached a level that managers could not continue to monitor. Initially, word filtering techniques were used to prevent malicious writing from being posted in a form that could not post or comment if a specific word was included, but they avoided filtering in a bypassed form, such as using similar words. As a way to solve this problem, deep learning was used to monitor posts posted by users in real-time, but recently, the community uses words that can only be understood by the community or from a human perspective, not from a general Korean word. There are various types and forms of characters, making it difficult to learn everything in the artificial intelligence model. Therefore, in this paper, we propose a preprocessing technique in which each character of a sentence is imaged using a CNN model that learns the consonants, vowel and spacing images of Korean word and converts characters that can only be understood from a human perspective into characters predicted by the CNN model. As a result of the experiment, it was confirmed that the performance of the LSTM, BiLSTM and CNN-BiLSTM models increased by 3.2%, 3.3%, and 4.88%, respectively, through the proposed preprocessing technique.

Keywords : Natural Language Processing, Preprocessing, Deep Learning, Malicious Comment

## 온라인 커뮤니티에서 사용되는 댓글의 형태를 고려한 악플 탐지를 위한 전처리 기법

김 해 수<sup>†</sup> · 김 미 희<sup>††</sup>

### 요 약

인터넷이 보급되면서 사람들 간의 소통을 위한 커뮤니티가 활성화됨과 함께 익명 커뮤니티가 나타났고 익명성을 이용한 공격적인 게시물, 댓글을 남기는 등 타인에게 피해를 주는 행위를 하는 이용자가 많아지고 있다. 과거에는 관리자가 직접 글과 댓글을 확인하며 삭제 및 차단했지만, 커뮤니티 이용자가 늘어나면서 관리자가 계속 감시할 수 없는 수준에 이르렀다. 초기에는 특정 단어가 포함되면 해당 글을 게시하거나 댓글을 달 수 없는 형태로 악의적인 글이 게시되는 것을 막는 단어 필터링 기법을 사용하였으나 유사한 단어를 사용하는 등 우회하는 형식으로 필터링을 피해 갔다. 이를 해결하는 방법으로 딥러닝을 이용하여 실시간으로 이용자들이 게시하는 글들을 감시하였으나 최근 커뮤니티에서는 해당 커뮤니티에서만 이해할 수 있는 단어를 사용하거나 일반적인 한글이 아닌 인간의 시야에서만 이해할 수 있는 문자를 사용하고 있다. 이들이 사용하는 문자의 종류나 형태가 다양하여 인공지능 모델에 모든 것을 학습시키기에 어려움이 있다. 이에 본 논문에서는 한글의 자음과 모음 띄어쓰기 이미지를 학습시킨 CNN 모델을 이용해서 문장의 각 문자를 이미지화해 인간의 시야에서만 이해할 수 있는 문자를 모델이 예측한 문자로 변환하는 전처리 기법을 제안한다. 실험 결과, 제안한 전처리 기법을 통해 LSTM, BiLSTM, CNN-BiLSTM 모델에서의 성능이 각각 3.2%, 3.3%, 4.88% 증가함을 확인했다.

키워드 : 자연어 처리, 전처리, 딥러닝, 악성댓글

## 1. 서 론

인터넷이 활성화되고 모바일 기기의 보급으로 공간의 제약 없이 인터넷에 접근할 수 있어 사람들 간의 소통을 위한 커뮤니티가 활성화되었다[1]. 많은 커뮤니티 중에 이용자들의 신원을 감춰 주는 익명 커뮤니티 또한 활성화되었고 이

※ 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2018R1A2B6009620).

† 준 회 원 : 한경대학교 컴퓨터응용수학부 석사과정

†† 종신회원 : 한경대학교 컴퓨터응용수학부 컴퓨터시스템연구소 교수

Manuscript Received : August 22, 2022

First Revision : October 18, 2022

Accepted : November 9, 2022

\* Corresponding Author : Kim Mi Hui(mhkim@hknu.ac.kr)

를 이용해 공격적인 문장이 포함된 글을 게시하거나 타 이용자의 글에 악성댓글을 다는 등의 행위를 통해 타 이용자들에게 손해를 끼치고 있다[2]. 과거에는 관리자가 직접 글과 댓글을 감시하며 삭제, 이용자 차단했었지만, 인터넷을 사용할 수 있는 기기들의 보급이 늘어남[3]에 따라 시간, 공간의 제약이 사라져 관리자가 계속 감시할 수 없는 수준에 이르렀다.

악성댓글 처리를 자동화한 초기에는 특정 단어가 포함되면 해당 글을 게시하거나 댓글을 달 수 없는 형태로 악의적인 글이 게시되는 것을 막는 단어 필터링 기법을 사용하였으나 유사한 단어를 사용하는 등 필터링되는 단어를 제외하고 다른 단어를 사용하여 차단 프로그램을 우회하는 형식으로 필터링을 피해 갔다.

이를 해결하기 위해 딥러닝을 이용하여 실시간으로 이용자들이 게시하는 글들을 감시하였으나 최근 커뮤니티에서는 특정 커뮤니티에서만 이해할 수 있는 단어를 사용하거나 일반적인 한글이 아닌 인간의 시야에서만 이해할 수 있는 문자를 섞어 일명 야민정음이라는 형태의 한글을 사용하고 있다 [4]. Table 1은 야민정음을 한글이 아닌 문자를 이용해서 표현한 예시이다.

이들이 사용하는 문자의 종류나 형태가 다양하여 인공지능 모델에 모든 것을 학습시키기에는 어려움이 있다. 또한 입력의 형태가 일관되지 않으면 학습과 예측을 하는 과정에서 모델의 성능 저하가 일어나게 된다.

이에 [5]에서 CNN(Convolutional Neural Network)[6]을 이용하여 한글이 아닌 문자로 만든 문장을 한글 문장으로 바꾸고 문장 교정 기법을 이용하여 일관된 형태의 문장을 인공지능 모델에 학습할 수 있는 전처리 기법을 제안하였다. 본 논문의 기여는 다음과 같다.

- [5]의 전처리과정 개선
- [5]에서 제안된 전처리 기법 상세하게 구현 및 실험 수행
- 전처리 유무에 의한 자연어 처리 모델의 정확도 비교

본 논문의 2장에서는 논문에 나오는 기법들에 대한 배경지식을 소개하고, 3장에서는 제안하는 전처리 기법을 자세하게

설명한다. 4장에서는 실험 데이터, 성능 평가 방법의 설명 및 제안한 전처리 기법의 성능을 평가하고 5장에서 결론을 맺는다.

## 2. 관련 연구

[7]의 논문에서 다양한 유형의 댓글을 수집하기 위해 영상의 카테고리마다 댓글을 추출하여 라벨링, 결측값 제거, 한글과 공백을 제외한 모든 문자를 제거하는 정규 표현식 변환, 토큰화, 정수 인코딩 등, 모든 입력의 길이를 맞추는 패딩을 하는 데이터 정제 과정 후, RNN(Recurrent Neural Network)[8]의 단점이 드러나는 시계열 데이터의 후반부에서는 앞에서 학습된 데이터의 정보를 잃어버리는, 즉, Vanishing Gradient Problem을 해결하기 위해 LSTM(Long-Short Term Memory)[9]을 이용하여 일정하지 않은 댓글의 길이는 물론 길이가 긴 댓글도 학습할 수 있도록 하였다.

[10]의 논문에서는 의사결정 나무, 로지스틱 회귀분석, 베르누이 나이브 베이즈, 랜덤 포레스트, 선형 커널 기반 서포트 벡터 머신, 가우시안 커널 기반 서포트 벡터 머신을 이용하여 가장 많은 모델이 예측한 클래스를 선택하는 다수결 기법과 예측한 확률의 평균을 통해 가장 높은 평균을 가진 클래스를 예측하는 앙상블 모델을 제안하였다.

[7, 10]의 연구에서 여러 기법을 이용해 유튜브에서 악성 댓글을 탐지하는 방법을 소개하였고 높은 성능을 보였다.

그러나, 문장의 극히 일부에서 새로운 형태나 한글이 아닌 문자가 포함되어 있으면 모델에 큰 영향을 주지 않을 수 있다. 또한, 한글을 대체하는 문자의 비율이 점차 높아진다면 잠재적으로 모델에 큰 영향을 주게 될 것이다.

## 3. 제안하는 전처리 기법

본 장에서는 본 논문에서 제안하는 전처리 기법을 설명한다.

[5]에서 제안한 전처리 모델은 파이썬을 이용해 문자를 이미지로 변환하고, 변환한 이미지를 CNN 모델로 예측한 문자로 변환하여 오타 및 띄어쓰기 교정을 하는 과정이었다.

본 논문에서는 문장을 자음과 모음으로 분리 후 분리된 문자들을 이미지로 변환하고 변환한 이미지를 CNN을 이용해 예측한 문자로 변환하여 자음과 모음으로 분리된 문장을 모델에 예측하는 과정으로 개선하였다.

[5]의 전처리 모델은 11,172자를 데이터로 사용하기 때문에 해당 데이터의 클래스를 학습하기 위해서는 전처리과정에 사용되는 CNN 모델에서 많은 파라미터가 사용되어야 하고 이후 실제 데이터를 처리할 때 시간이 걸린다는 단점이 있었다. 해당 단점을 해결하기 위해 문장을 자음과 모음으로 분

Table 1. Example of Yaminjeongeum

|   | Korean word | Replacement word | Example     |
|---|-------------|------------------|-------------|
| 1 | ㅌ           | E                | 태 -> E#     |
| 2 | ㅣ           | I, 1             | 이 -> oI, o1 |
| 3 | ㅙ           | H                | 태 -> #EH    |
| 4 | ㄱ           | 7                | 가 -> 7#     |
| 5 | ㄴ           | L                | 나 -> L#     |
| 6 | ㅏ           | K                | 야 -> oK     |

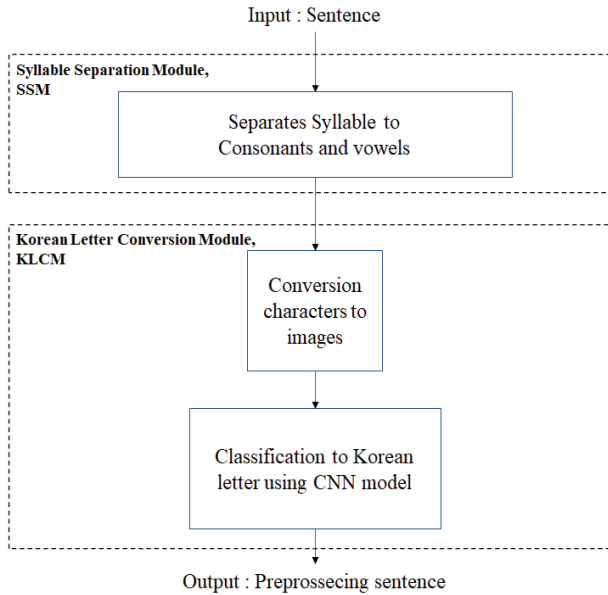


Fig. 1. Proposal Preprocessing

리해서 자음과 모음, 띄어쓰기만 사용하는 52개의 클래스만 학습하여 처리하는 과정으로 개선하였다.

Fig. 1은 제안하는 전처리 기법의 전체 과정을 보여주는 그림이다. 문장이 입력되면 SSM(Syllable Separation Module)에서 한글인 경우, 모든 음절에 대하여 자음과 모음을 분리한다. 이후 KLCM(Korean Letter Conversion Module)에서 각 자음과 모음, 기타 문자들을 이미지로 변환하고, 해당 문자는 CNN 모델이 예측한 문자로 변환된다.

3.1 문장을 자음과 모음 단위로 분리하는 모듈(SSM, Separates Syllable Module)

SSM은 입력으로 들어온 문장에서 한글을 자음과 모음 단위로 분리하는 모듈이다.

방식은 한글의 음절의 유니코드[11]를 이용해 초성 19자, 중성 21자, 종성, 28자로 이루어진 각각의 인덱스를 도출한다[12].

Table 2[12]는 초성, 중성, 종성의 인덱스 값을 표기한 표이다.

Fig. 2는 한글 ‘않’을 분리하는 과정을 보여주는 그림이다.

Fig. 2에서 1)은 한글 ‘않’의 유니코드 16진수(0xC54A)와 10진수(50,506)이다. 2)는 인덱스를 계산하기 위해 한글 유니코드의 시작 값인 ‘가’의 유니코드(0xAC00, 44032)를 뺀다. 3)은 초성의 유니코드는 모든 중성과 종성이 지난 다음에 다음 초성이 나오기에 중성과 종성 개수의 곱을 나누고 소수 점을 버린 값이 초성 인덱스가 된다. 4)는 중성은 모든 중성이 지난 다음에 나오기 때문에 종성 개수만큼 나눈 값에 중성의 길이만큼 나눈 값의 나머지가 인덱스가 된다. 5) 종성은 종성의 길이만큼 나눈 값의 나머지가 인덱스가 된다.

Table 2. Index of Korean Syllable[12]

|    | 초성(Onset) | 중성(Nucleus) | 종성(Coda) |
|----|-----------|-------------|----------|
| 0  | ㄱ         | ㅏ           |          |
| 1  | ㄲ         | ㅑ           | ㄱ        |
| 2  | ㅋ         | ㅓ           | ㄲ        |
| 3  | ㆁ         | ㅕ           | ㄱ        |
| 4  | ㆁ         | ㅗ           | ㄴ        |
| 5  | ㄴ         | ㅛ           | ㄴ        |
| 6  | ㄷ         | ㅜ           | ㄴ        |
| 7  | ㄷ         | ㅠ           | ㄴ        |
| 8  | ㄷ         | ㅡ           | ㄴ        |
| 9  | ㄷ         | ㅝ           | ㄴ        |
| 10 | ㄷ         | ㅞ           | ㄴ        |
| 11 | ㄷ         | ㅟ           | ㄴ        |
| 12 | ㄷ         | ㅠ           | ㄴ        |
| 13 | ㄷ         | ㅡ           | ㄴ        |
| 14 | ㄷ         | ㅢ           | ㄴ        |
| 15 | ㄷ         | ㅣ           | ㄴ        |
| 16 | ㄷ         | ㅤ           | ㄴ        |
| 17 | ㄷ         | ㅥ           | ㄴ        |
| 18 | ㄷ         | ㅦ           | ㄴ        |
| 19 | ㄷ         | ㅧ           | ㄴ        |
| 20 |           | ㅣ           | ㅈ        |
| 21 |           |             | ㅇ        |
| 22 |           |             | ㅉ        |
| 23 |           |             | ㅊ        |
| 24 |           |             | ㅋ        |
| 25 |           |             | ㅌ        |
| 26 |           |             | ㅍ        |
| 27 |           |             | ㅎ        |

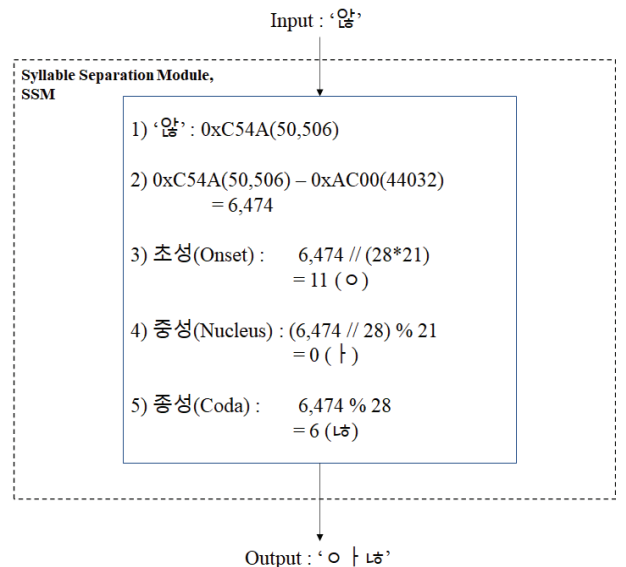


Fig. 2. Example of the Processing of SSM

3.2 한글로 변환하는 모듈(KLCM, Korean Letter Conversation Module)

KLCM은 3.1에서 분리된 문자들을 이미지로 변환하고 CNN을 통해 예측한 문자로 변환한다.

Fig. 3은 KLCM에서 입력과 출력의 예시이다.

- 1) 문자를 이미지로 변환(Conversion characters to Image)  
파이썬의 Pillow 라이브러리[13]의 ImageDraw 함수를 이용해서 3.1에서 분리된 문자들과 그 외 한글이 아닌 문자들을 모두 이미지로 변환하고 학습을 위해 넘파이 배열에 추가하여 이미지 데이터 셋으로 변환한다.

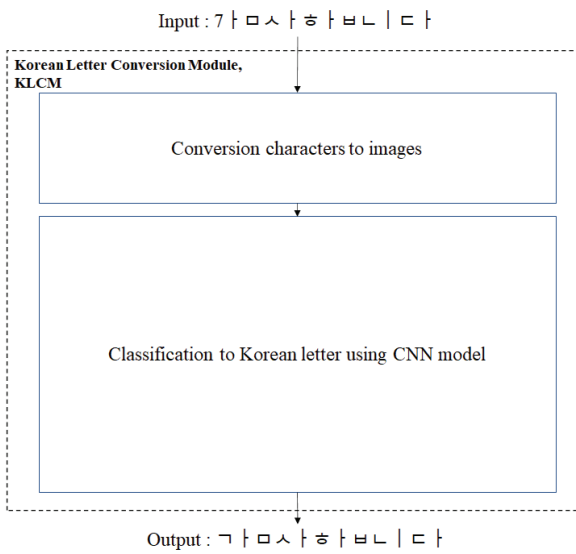


Fig. 3. Input and Output Example of KLCM

Fig. 4는 문자를 이미지로 변환하는 과정의 예시이다.

- 2) CNN 모델을 이용한 한글로 분류(Classification to korean letter using CNN Model)

1)의 과정을 통해 만들어진 이미지를 한글이 학습된 CNN 모델이 예측으로 각 문자의 0과 51 사이의 인덱스 값을 출력한다. 중복을 제외하면 모든 자음과 모음 및 띄어쓰기는 52개이다. 자음과 모음이 입력된 리스트에서 인덱스값에 해당하는 문자로 변환해서 문장을 만든다.

Fig. 5는 CNN 모델이 한글로 분류하는 예시이다.

전처리기에서 사용되는 CNN 모델은 Conv2D 층과

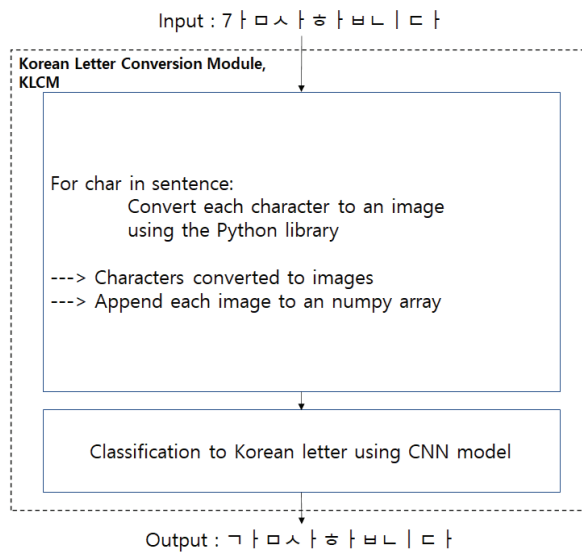


Fig. 4. Example of the Processing of Conversion Characters to Image

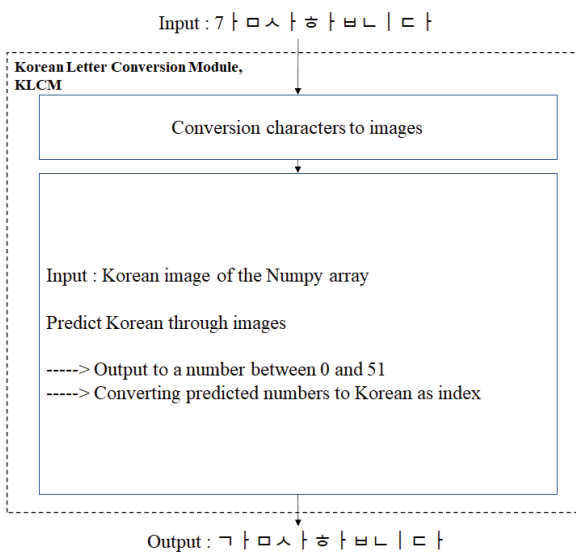


Fig. 5. Example of the Processing of Classification to Korean Letter Using CNN Model

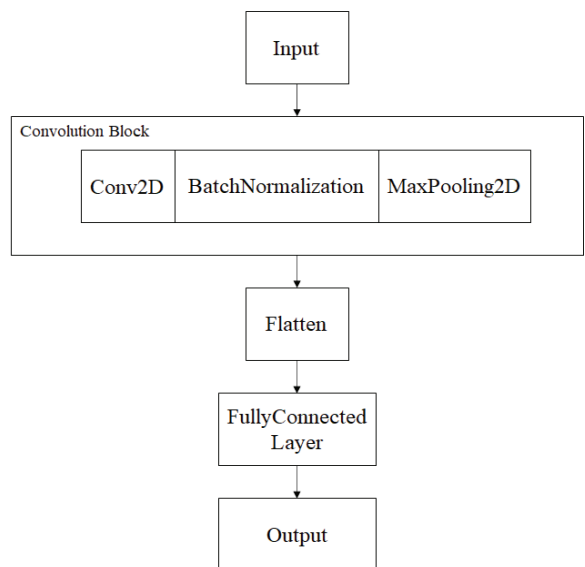


Fig. 6. Structure of KLCM's CNN Model

MaxPooling2D 층으로 이루어진 합성곱 블록 여러 개를 이어 붙였다. 또한 모델의 깊이가 깊어지고 값들이 누적되면 이후 들어오는 입력데이터는 기존에 학습된 데이터와는 다른 데이터라고 예측하는 과대적합에 빠질 수 있기에 Batch-Normalization 레이어를 통해 학습되는 가중치를 정규화하여 이를 해결하는 모델 구조를 가진다. Fig. 6은 전처리기에 사용되는 모델의 구조도이다.

#### 4. 성능 평가

본 장에서는 제안한 전처리 기법의 성능을 평가하기 위한 실험 환경, 실험 데이터 및 성능 평가 방법을 소개하고 실험 결과를 분석한다.

##### 4.1 실험 환경

실험 환경은 Intel® Xeon(R) Silver 4215R CPU @ 3.20GHz CPU, 256GB RAM, NVIDIA RTX A6000 GPU 환경에서 진행하고 파이썬 3.7.13, 텐서플로우 2.7.0에서 전처리기 CNN 모델 및 테스트 텍스트 분류기 모델을 구성한다.

##### 1) 실험 데이터

CNN 모델에 학습을 위해 사용한 데이터는 무료로 공개된 한글 폰트를 이용해 자음과 모음, 띄어쓰기로 이루어진 52개의 클래스를 가진 데이터를 생성하여 한글의 자음과 모음으로 이루어진 이미지 데이터이다.

아래 Fig. 7과 Fig. 8은 생성된 데이터의 예시이다.

또한, 제한된 수의 폰트로 생성된 데이터로 인한 모델의 성능 저하를 해결하기 위해 데이터 증강의 일환으로서 글자의 크기와 문자의 위치를 변경하는 등의 작업을 통해 총 53,664개의 데이터를 학습 데이터 세트로 사용하였다.



Fig. 7. Example of Character Data : 'ㄱ'



Fig. 8. Example of Character Data : 'ㅏ'

전처리 성능 테스트에 사용된 데이터는 욕설 감지 데이터 세트[14]과 Korean Hate Speech Dataset[15]이다. [14]은 여러 커뮤니티의 댓글에서 수집된 5,825개의 데이터로 단순 욕설, 인종 차별적인 말, 정치적 갈등을 조장하는 말, 성적·성차별적인 말, 타인을 비하하는 말, 그 외에 불쾌감을 주거나 욕설로 판단되는 말이 욕설로 데이터 세트가 구성되어있고 [15]은 연예 뉴스의 댓글에서 수집된 9,381개의 데이터이며 테스트 데이터는 None으로 되어있는 문장을 평균, 나머지를 악성댓글로 사상하여 테스트 데이터를 구성하였다.

##### 2) 성능 평가 방법

전처리 기법의 성능은 LSTM, BiLSTM, CNN-BiLSTM 모델에 성능 평가 데이터를 6:4 비율로 학습데이터와 평가 데이터로 이용한다.

각 모델은 전처리기의 성능 측정만을 위한 텍스트 분류기이기 때문에 단순한 모델 구조로 구성한다. Fig. 9, Fig. 10, Fig. 11은 텍스트 분류기들의 모델 구조를 시각화한 그림이다.

평가 방법은 각 텍스트 분류기에 대해 전처리하지 않은 문장과 한글, 띄어쓰기가 제거된 문장 두 가지와 제안한 전처리 기법으로 나누어서 비교한다. 성능 평가 지표는 F1-Score를 이용한다. F1-Score는 모델이 악성댓글이라 분류한 것 중 실제로 악성댓글인 것의 비율인 정밀도와 실제 악성댓글인 것 중 모델이 악성댓글이라고 예측한 것의 비율인 재현율의 조화평균이다. 단순히 전체 예측 중에 정답을 맞춘 것의 비율을 따지는 정확도에 비해 재현율과 정밀도 중 더 작은 값에 영향을 많이 받아 오탐에 민감한 지표이기에 성능 평가 지표로 선택한다.

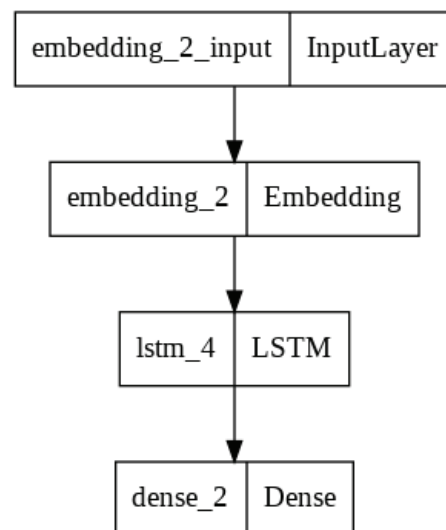


Fig. 9. Structure of LSTM Model

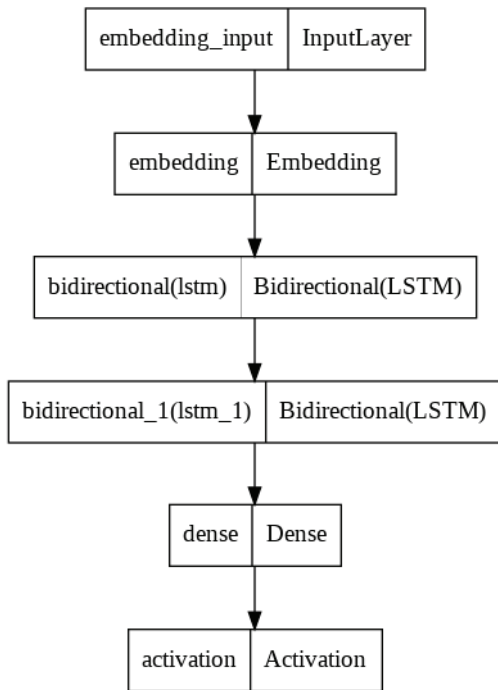


Fig. 10. Structure of BiLSTM Model

4.2 실험 결과 분석

본 절에서는 성능 평가 방법을 통해 전처리 기법을 평가하고자 한다. Fig. 12는 각 텍스트 분류기에 대하여 전처리하지 않은 것(T1. Before Preprocessing), 한글과 띄어쓰기를 제외하고 전부 제거한 것(T2. Remove characters except Korean and spacing) 그리고 제안한 전처리 기법(T3. Proposed Preprocessing)을 적용한 세 가지 기법에 대한 F1-Score 차이를 보여주는 그림이다.

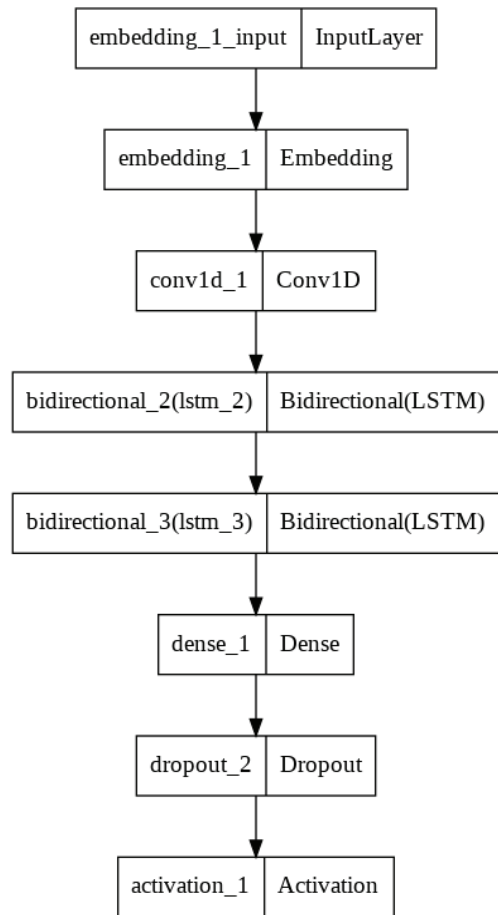


Fig. 11. Structure of CNN-BiLSTM Model

LSTM 모델에서 제안한 전처리 기법(T3)은 전처리 하지 않은 것(T1)에 비해 F1-Score가 3.2% 증가하였고 한글을 제외한 문자를 전부 제거한 것(T2)에 비해 2.26% 증가하였

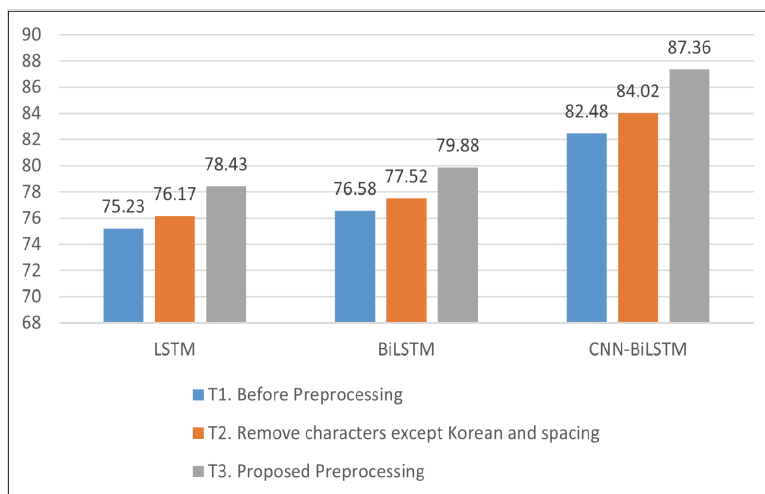


Fig. 12. Compare Each Technique with F1-Score

다. BiLSTM 모델에서 T3는 T1에 비해 3.3% 증가했고 T2보다 2.35% 증가하였다. CNN-BiLSTM 모델에서는 T1, T2보다 T3가 각각 4.88%, 3.34% 증가하였다. 이를 통해 한글이 아닌 문자가 포함된 문장을 인공지능 모델에 학습할 때 해당 문자들이 모델에게 영향을 주고 있으며 이를 제거함으로써 영향을 줄였지만 제거된 문자에 문장이 악성댓글인지 아닌지를 판별하는 요소가 포함되어 있을 수 있다. 그렇기에 한글이 아닌 문자를 변환하는 전처리 기법의 성능이 단순히 문자들을 제거하는 것보다 높은 성능을 보였다.

### 5. 결 론

기존의 필터링 기반 탐지를 우회하기 위해 한글이 아닌 문자를 사용하고 그것을 탐지하기 위해 해당 문자들을 제거하는 기존의 전처리 방법은 한글이 아닌 문자를 제거하지 않은 것보다 성능은 더 높았지만, 해당 문자가 문장의 악성 댓글 여부를 판별하는 주요한 문자일 수 있기에 학습되는 문장의 수가 많아지면 예측 모델의 성능에 영향을 주게 될 것이다. 문장의 극히 일부에서 한글을 대체하는 문자가 사용된다면 문맥을 파악할 수 있는 모델에서는 해당 문자들이 제거되어도 충분히 탐지할 수 있지만 제거되는 문자가 많고 문장의 길이가 짧을수록 탐지의 정확도가 확연히 떨어지게 될 것이다.

본 논문에서는 CNN을 이용하여 한글 대신 쓰인 문자를 한글로 변환하는 전처리 기법을 제안하였으며, 실험 결과를 통해 문자들을 제거하는 것보다 해당 요소를 학습할 수 있도록 한글로 변환해주는 전처리 기법이 효과적임을 보였다. 모든 문장에 사용된 한글이 아닌 문자들이 한글을 대체해서 사용된 것이 아니기에 이를 고려하면 정확도가 더 높아질 것이다.

향후 연구에서는 이용자들의 한글 사용 패턴을 분석해서 주로 사용되는 문자는 남기고 영향력이 없는 문자는 제거하는 과정을 추가해 전처리 성능을 높이는 방안을 연구할 것이다.

### References

[1] K. Suh, "The effects of the characteristics of internet communities and individuals on user loyalty," *Asia Pacific Journal of Information Systems*, Vol.13, No.2, pp.1-22, 2003.

[2] J. Shin, K. Kim, and J. Park, "The effect of social anonymity on community quality: Mediated by self expression, self control, and community ties," *The Journal of Society for e-Business Studies*, Vol.16, No.4, pp.257-281, 2011.

[3] Gallup Report, [Internet], <https://www.gallup.co.kr/gallupdb/reportContent.asp?seqNo=1309>

[4] O. Kang, "A study on the expression of deconstructionism for Yaminjeongeum and Geupsikche," Institute of Humanities, Chosun university - the study of humanities, Vol.56, pp.325-349, 2018.

[5] H. Kim and M. Kim, "Preprocessing technique for natural language processing considering the form of characters used in malicious comment," *Proceedings of the Annual Spring Conference of Korea Information Processing Society Conference (KIPS) 2022*, Vol.29, pp.543-545, 2022.

[6] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[7] J. Kim and J. Kook, "Design and implementation of a LSTM-based YouTube malicious comment detection system," *Smart Media Journal*, Vol.11, No.2, pp.18-24, 2022.

[8] M. I. Jordan, "Serial order: A parallel distributed processing approach," Tech. rep. ICS 8604. San Diego, California: Institute for Cognitive Science, University of California, May, 1986.

[9] S. Hochreiter and J. Schmidhuber, "LONG SHORT-TERM MEMORY," *Neural Computation*, Vol.9, No.8, pp.1735-1780, 1997.

[10] M. Jeong, J. Lee, and H. Oh, "Ensemble machine learning model based YouTube spam comment detection," *Journal of the Korea Institute of Information and Communication Engineering*, Vol.24, No.5, pp.576-583, 2020.

[11] The Unicode Standard, Version 14.0-Hangul Syllables, [Internet]. <https://www.unicode.org/charts/PDF/UAC00.pdf>

[12] H. Kwon, J. Kim, and H. Kwak "Researches to divided Hangul spelling for program based on Uni-Code," *Proceedings of the 31th KISS Fall Conference*, Vol.31, No.2, pp.781-783, 2004.

[13] Pillow, [Internet], <https://pillow.readthedocs.io/en/stable/>

[14] Curse Detection Dataset, [Internet]. <https://github.com/2runo/curse-detection-data>

[15] Korean HateSpeech Dataset, [internet]. <https://github.com/kocohub/korean-hate-speech>



김 해 수

<https://orcid.org/0000-0003-1844-1958>

e-mail : [ww232330@hknu.ac.kr](mailto:ww232330@hknu.ac.kr)

2022년 한경대학교 컴퓨터응용수학부  
(학사)

현재 한경대학교 컴퓨터응용수학  
석사과정

관심분야 : 네트워크 보안, 인공지능, 네트워크 슬라이싱, 자연어 처리



김 미 희

<https://orcid.org/0000-0002-4896-7400>

e-mail : mhkim@hknu.ac.kr

1997년 이화여자대학교 전자계산학과(학사)

1999년 이화여자대학교 컴퓨터학과(석사)

2007년 이화여자대학교 컴퓨터학과(박사)

1999년 ~ 2003년 한국전자통신연구원

연구원

2007년 ~ 2009년 이화여자대학교 컴퓨터학과 전임강사

2009년 ~ 2010년 노스캐롤라이나주립대학교 연구원

2011년 ~ 현 재 한경대학교 컴퓨터응용수학부

컴퓨터시스템연구소 교수

관심분야: 네트워크 성능 분석 및 보안, 무선네트워크 보안,

침입대응, 클라우드센싱, 블록체인