[UCI]I804:24011-200000490323

2021년 8월

석사학위 논문

품사기반 자질 추출을 통한 악성댓글 판별 성능 향상 방법

조선대학교 산업기술창업대학원

소프트웨어융합공학과

문 종 민

품사기반 자질 추출을 통한 악성댓글 판별 성능 향상 방법

Malicious comment discrimination performance improvement method through feature extraction based on part-of-speech

2021년 8월 27일

조선대학교 산업기술창업대학원

소프트웨어융합공학과

문 종 민

품사기반 자질 추출을 통한 악성댓글 판별 성능 향상 방법

지도교수 김 판 구

이 논문을 공학석사학위신청 논문으로 제출함.

2021년 4월

조선대학교 산업기술창업대학원

소프트웨어융합공학과

문 종 민

문종민의 석사학위논문을 인준함

위원장 조선대학교 교수 신 주 현 (인)

위 원 조선대학교 교수 최 준 호 (인)

위 원 조선대학교 교수 김 판 구 (인)

2021년 5월

조선대학교 산업기술창업대학원

목 차

표 목 차

그 림 목 차

ABSTRACT

Malicious comment discrimination performance improvement method through feature extraction based on part-of-speech

Jongmin Moon
Advisor : Prof. PanKoo Kim, Ph.D.
Department of Software Convergence Engineering
Graduate School of Industrial Technology and Entrepreneurship, Chosun University

One of the social aspects that have changed with the widespread use of the Internet is communication in the online space. In the past, only one-on-one conversations were possible remotely, except when they were physically in the same space, but nowadays, technology has been developed to enable communication with a large number of people remotely through bulletin boards, communities, and social network services.

Due to the development of such information and communication networks, life becomes more convenient, and at the same time, the damage caused by rapid information exchange is also constantly increasing. Recently, cyber crimes such as sending sexual messages or personal attacks to certain people with recognition on the Internet, such as not only entertainers but also influencers, have occurred, and some of those exposed to these cybercrime have committed suicide.

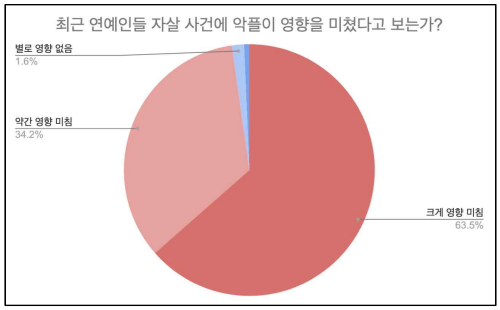
In this paper, in order to reduce the damage caused by malicious comments,

research a method for improving the performance of discriminate malicious comments through feature extraction based on parts-of-speech.

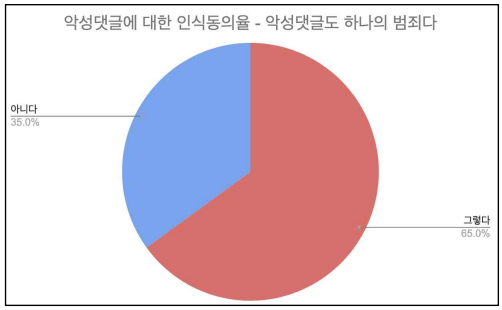
Ⅰ. 서론

A. 연구 배경 및 목적

인터넷 사용이 보편화됨에 따라 변화된 사회적 양상 중 하나는 온라인 공간의 커뮤니케이션이다. 과거에는 물리적으로 같은 공간에 있는 경우를 제외하면 원격으로는 일대일 대화만 가능했지만, 현재는 게시판이나 커뮤니티, 사회 관계망 서비스 등을 통해 원격으로 다수와 커뮤니케이션이 가능할 수 있게 기술이 발달하였다.[1] 이러한 정보통신망의 발전으로 인하여 생활이 편리해짐과 동시에 신속한 정보교환으로 인한 폐해 또한 지속적으로 증가하고 있다. 최근 들어서는 연예인 뿐만 아니라, 인플루언서와 같이 인터넷상에서 인지도가 높은 특정인에게 성적 메시지를 보내거나, 인신공격하는 등의 사이버 범죄도 발생하고 있으며, 이러한 사이버 범죄에 노출되는 사람들 중 일부는 극단적인 선택을 하기도 하였다.



[그림 1-1] 악플이 자살에 미치는 영향
※한국언론진흥재단 미디어연구센터 온라인 설문조사,(2019년 12월)



[그림 1-2] 악성댓글도 범죄
※ SM C&C Tilon, '악성댓글이 없는 세상을 꿈꾸며' (2019년 11월)

악성댓글을 예방하는 방법은 크게 비기술적 방법, 기술적 방법으로 분류된다. 비기술적 방법은 정책적으로 댓글 쓰기를 금지한다거나 로그인한 실명 확인 사용자만 작성이 가능하도록 제한하는 방식 등으로 영미권 국가의 언론사들이 주로 선택하는 방법이다. 우리나라에서도 다음(2019년 10월), 네이버(2020년 3월)가 연예뉴스 댓글을 폐지한 바 있으며, 2020년 3월 19일에는 네이버가 댓글 이력을 전면 공개하는 제도적 조치를 단

행하였다. 기술적 방법은 IT 기술을 활용해 악성댓글을 자동 분류하는 방안으로 특정 문자를 필터링하여 치환하거나, 머신러닝을 활용하는 방법을 의미한다. 해외에서는 구글, 유튜브, 페이스북, 인스타그램, 스냅 등 IT 기업들이 주로 채택하고 있는 방식이다.

본 논문에서는 머신러닝 기법의 하나인 SVM(Support Vector Machine)을 사용한 악성댓글 판별 모델을 구현하고, 판별 성능 향상을 위한 품사 기반 자질 추출방법에 관해서 연구하고자 한다.

Ⅱ. 관련 연구

악성 댓글은 줄여서 악플이라 부르는데, 타인에게 피해를 주는 댓글로 상대를 비방하거나 저주 혹은 협박의 내용이 있는 댓글, 사회 통념에 위배되는 내용을 담은 댓글, 혹은 게시판의 정상 운영을 방해하는 댓글을 말한다.[2] 학계에서는 '악성 댓글', '악의적 표현'이라는 단어보다 '혐오 표현'이라는 단어로 주로 연구되고 있다.[3]

본 논문에서 악성댓글은 '인터넷상에 게시된 내용에 대하여 자신의 일방적인 의견을 포함하여 타인에게 부정적인 영향을 미치는 댓글'로 정의한다.

A. 형태소 기반 자질 추출

본 절에서는 악성댓글 판별을 위해 기존 연구에서 사용하는 형태소 기반 자질 추출 방법에 대해서 분석한다. 자질 추출이란 문장 혹은 단어로 구성되어있는 데이터를 머신러닝 모델에서 사용할 수 있는 데이터 형태로 가공하는 작업을 의미한다. 자질 추출은 데이터 유형, 머신러닝 모델에 따라서 매우 다양한 형태로 나타난다.

그중 하나가 품사나 형태소를 통한 자질 추출이다. 특수문자를 제거하고 반복되는 패턴을 통일하고, 형태소(조사, 의존명사)를 분리하여 머신러닝에 사용한 연구[4], 형태소 분석 이후 감성사전에 적용한 연구[5]가 있다. 그 외에 어휘자질과 품사, 형태소 기반 자질을 사용한 연구도 있다. 댓글의 어조를 존대어와 예사어로 구분하고, 어휘자질(댓글 길이)과 형태소 분석을 통해 가중치를 부여하는 연구[6], 명사만을 추출해 사용하는 연구[7], 형태소 분류와 노이즈만 제거하는 연구[8]도 있다. 공백을 구분자로 토큰화하고, 품사 태깅에 사용하는 연구[9], 체언, 용언, 부사, 형식형태소로 분류 후 전처리 과정을 수행하는 연구[10]도 있다.

본 논문에서는 기존 연구에서 나타난 자질 추출방법 중, 명사, 형용사, 조사와 같은 특정 품사에 해당하는 단어를 추출하는 방법을 사용하고, 추출한 단어를 악성댓글 판별모델의 자질로 사용하는 실험을 진행한다. 실험을 통해 특정 품사의 포함/미포함에 따른 성능의 변화를 확인하고자 한다.

B. 텍스트 벡터화

본 절에서는 텍스트를 머신러닝의 자질로 사용하는데 필요한 텍스트 벡터화 작업에 대해서 살펴본다. 텍스트 벡터화(Text Vectorization)는 인간이 사용하는 문자를 기계가 이해할 수 있도록 수치화시켜주는 작업을 의미한다. 텍스트 벡터화는 크게 국소표현(Local Representation)과 분산표현(Distributed Representation)이 있다. 국소표현(Local Representation)은 단어 그 자체를 사용하여 값을 추출하여 사용하는 방법으로 One-hot Vector, N-gram, BoW(Bag of Words) 등이 있고, 분산표현(Distributed Representation)은 단어와 단어 주변의 정보를 참고하여 값을 추출하여 사용하는 방법으로 Word2Vec, LSA, Glove 등이 있다.

본 논문에서는 특정 품사를 기반으로 하여 단어를 추출하여 빈도수에 따른 성능 변화 실험을 진행하기 때문에 국소표현 방법 중 빈도수 기반의 BoW 를 사용한다. 단어별 빈도수 추출은 Count Vectorizer를 사용하고, TF-IDF를 통해 가중치를 적용한다. TF-IDF(Term Frequency - Inverse Document Frequency)는 Count 기반의 벡터화에서 나타나는 문제점을 보완하기 위해 사용하는 방식으로, 모든 문서에서 전반적으로 자주 등장하는 단어는 페널티를 주고, 개별 문서에서만 등장하는 단어에는 가중치를 주어 벡터화시키는 방식이다. TF-IDF 벡터값을 구하는 수식은 (1)과 같다.

$$w_{i,j} = tf_{i,j} \times \log(\frac{N}{df_i}) \qquad (1)$$

$tf_{i,j}$는 특정 문서 j에서 i라는 단어가 나타나는 횟수를 의미한다. df_i는 i라는 단어를 포함하고 있는 문서의 개수를 의미한다. N은 전체 문서의 개수를 의미한다.

C. 악성댓글 판별

본 절에서는 악성댓글 판별을 위한 다양한 머신러닝 방법에 관한 기존 연구를 분석한다.

그 중 몇 가지를 살펴보면, 인터넷 신문의 댓글을 데이터로 사용하여 KTL 형태소

분석기로 자질(명사, 동사, 형용사, 어절 자체)을 선택한 후, TF-IDF 가중치 적용 후 SVM을 사용한 연구가 있다.[11] 이 연구에서는 형태소 분석기를 통한 품사 중에서 명사만, 명사와 형용사와 동사, 모든 품사, 어절 자체와 명사만, 어절 자체와 명사, 형용사, 동사 추출, 어절 자체와 모든 품사의 통합 추출 6가지 유형으로 분류하여 실험하였다. SVM과 Random Forest를 비교 분석한 연구도 있다.[12] 이 연구에서는 악성 댓글 판별에 가장 중요한 것은 자질 추출이라고 하였다. 연구방법은 전처리 과정 후, 빈도수가 높은 글자에서 초성만을 추출하여 자질로 구성하고 SVM과 Random Forest를 이용하여 악성댓글 판별을 실험하였다. CNN을 활용한 악성댓글 판별 연구도 있다.[13] 이 연구에서는 한국어 온라인 뉴스 댓글 데이터를 활용하여, Kim-CNN 기법으로 악성댓글 판별을 실험하였다. 입력받은 데이터를 Word2Vec 기법을 통해 임베딩하고, 여러 유형의 레이어를 적용한 CNN 기법을 사용하였다. 여러 가지 알고리즘을 비교 실험하는 연구도 있다[3]. 이 연구에서는 네이버 댓글 데이터를 활용하여, [명사], [명사 + 형용사], [명사 + 형용사 + 동사], [모든 품사] 4가지 유형으로 분류 후 RNN, LSTM, GRU 3가지 알고리즘을 적용하여 총 12개 모델의 성능을 비교하였다. 그 외에도 6가지 머신러닝 알고리즘(Decision-Tree, 로지스틱 회귀분석, Naïve Bayes Classification, Random Forest, Linear-SVM, Gaussian-SVM)을 실험하고 성능을 비교·분석하는 연구[14], 신문기사의 본문을 사용하여 5가지의 머신러닝 기법(Naïve Bayes Classification, k-NN, Decision-Tree, SVM. 인공신경망)의 성능분석을 진행한 연구도 있다.[15]

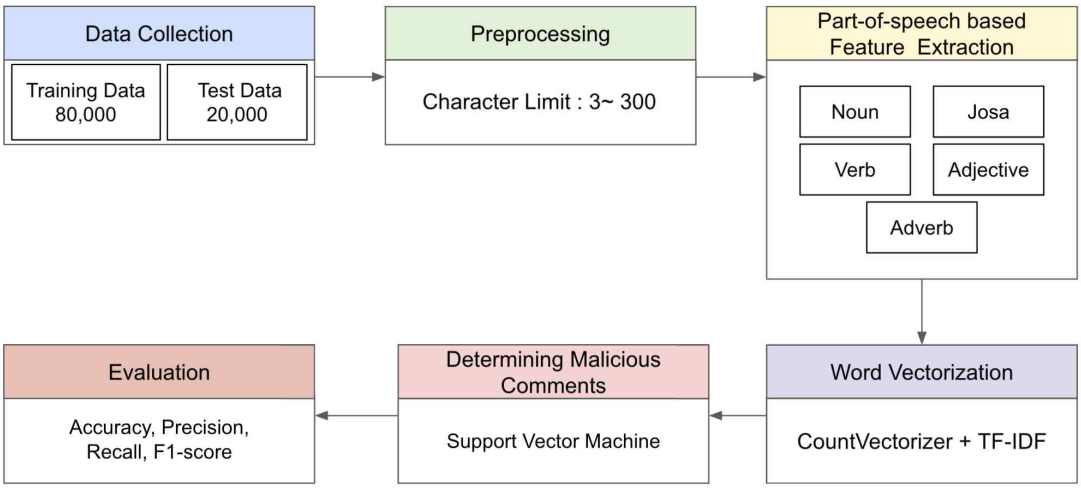
위의 연구를 살펴볼 때, SVM이 악성댓글 판별에 있어 널리 사용되고 있으며, 전반적으로 우수한 성능을 나타냄을 알 수 있었다. 그래서 본 논문에서는 SVM을 사용하여 악성댓글 판별 모델을 구축한다.

Ⅲ. SVM 기반 악성댓글 판별모델

본 장에서는 악성댓글 판별 성능 향상을 위한 품사 기반 자질을 알아보기 위하여 본 논문에서 제안한 실험방법에 관해 서술한다.

A. 시스템 구성도

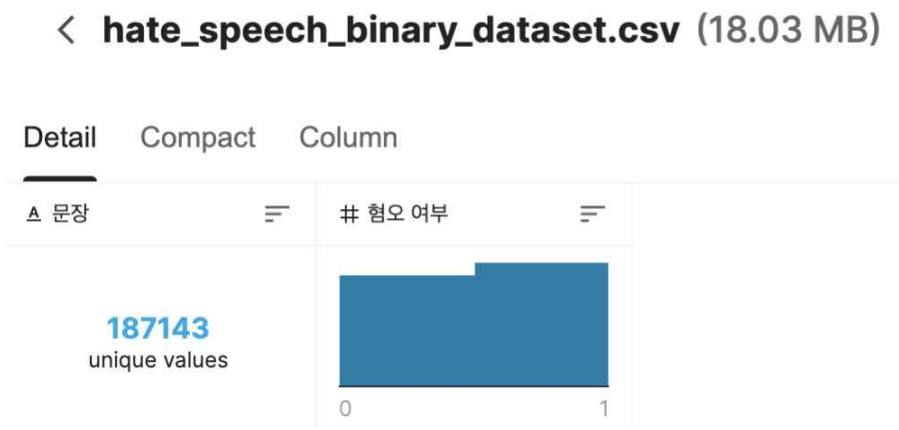
[그림 3-1]은 제안하는 악성댓글 판별모델에서 품사 기반 자질에 따른 성능 변화를 알아보기 위하여 설계한 전체 시스템 구성도이다. Data Collection, Preprocessing, Word Vectorization, SVM, Evaluation 5단계로 나뉘어진다.



[그림 3-1] 시스템 구성도

B. 데이터 전처리

본 실험에서 악성 댓글 판별에 사용한 데이터는 Kaggle 사이트에서 제공하는 한국어 혐오 표현 DataSet(hate_speech_binary_dataset)을 사용하였다. 해당 데이터는 악성 댓글은 0, 비악성 댓글은 1로 라벨링이 되어있으며, 데이터의 구성은 인터넷 기사 댓글과 영화 리뷰로 구성되어 있다.



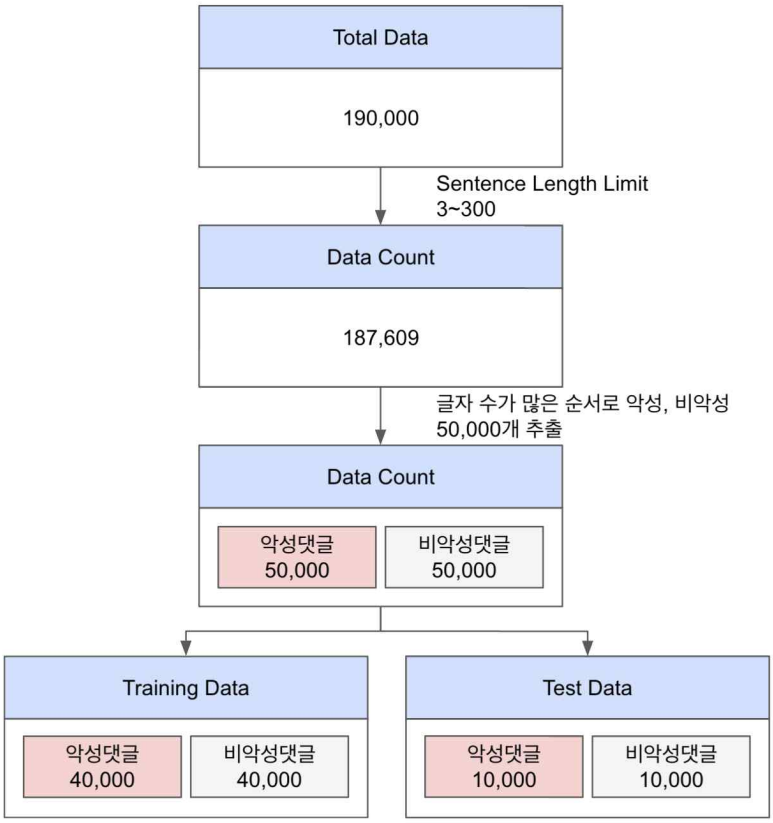
[그림 3-2] Kaggle - 한국어 혐오 표현 DataSet

0 문장, 혐오 여부 1 정말 재밌다 연기도 좋고 디카프리오 짱,1 2 후!!!!!!! 쩔었다....,1 3 “저 개돼지들은 왜 꼭 당해봐야 아는걸까??? 그래서 개돼진가?”, 0 4 “아니 고작 딴따라새끼들 입국 못 하게 했다고 트럼프니 뭐니 전쟁까지 들먹이네? 병신들”, 0 ... (하단 생략)

[표 3-1] 한국어 혐오 표현 DataSet

해당 데이터는 190,000개의 댓글로 구성되어 있고, 악성댓글은 0, 비악성댓글은 1로 라벨링되어 있다. 이 중 악성댓글은 90,000건, 비악성댓글은 100,000건이다. 3자 미만의 데이터는 한 글자로 된 감탄사이거나, 단어 하나로 된 내용(굳, 굿, ♥, 짱, 흐흐, 대작, 최고, 개쩜, 재밋, 호호 등)으로 악성댓글 판별에 도움이 되지 않는다고 판단하여 제거하였다. 300자 이상의 데이터는 댓글보다 별도의 게시물로 보이기 때문에 제외하였다. 문장의 길이 제한(3~300)을 통해서 일차적으로 187,609개의 댓글을 선별하고, 많은 자

질 추출을 위하여 글자 수가 높은 순으로 악성댓글 50,000건과 비악성댓글 50,000건을 추출하였다. 학습 데이터와 테스트 데이터의 비율은 데이터 셋이 적을 경우 6:4~7:3으로 분할하며, 데이터셋이 큰 경우에는 8:2~9:1을 사용하는 것이 좋다. 본 실험에서는 100,000건의 데이터셋을 사용하기 때문에 학습데이터와 테스트데이터의 비율을 8:2로 분류하였다.

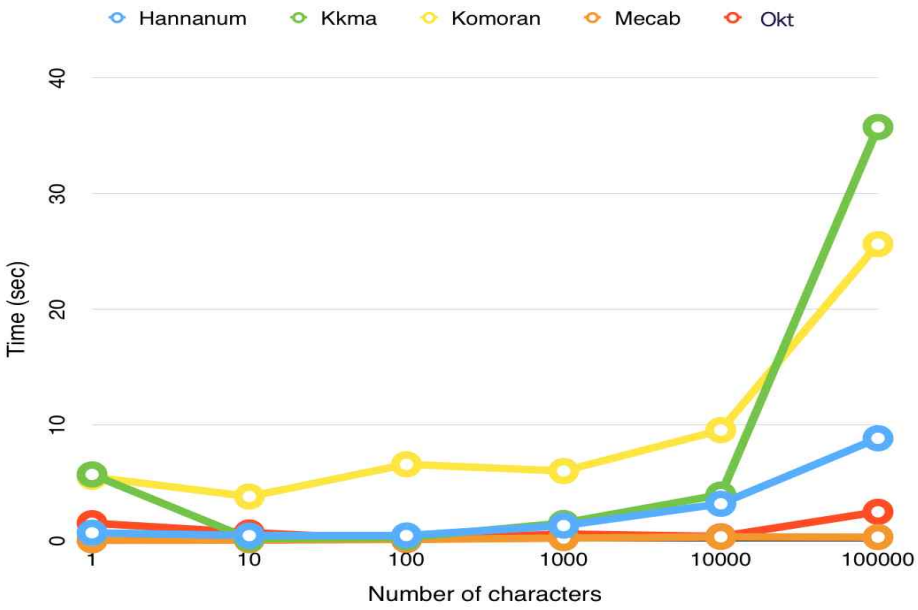


[그림 3-3] 학습데이터와 실험데이터 분류

C. 자질 추출 및 악성댓글 판별

1. 품사 선별

품사 태깅은 KoNLPy에서 제공하는 Okt 분석기를 사용한다.



[그림 3-4] KoNLPy 형태소 분석기 성능 비교

Hannanum	Kkma	Komoran	Mecab	Okt
아버지가방에들어가/N	아버지/NNG	아버지가방에들어가신다/NNP	아버지/NNG	아버지/Noun
이/J	가방/NNG		가/JKS	가방/Noun
시ㄴ다/E	에/JKM		방/NNG	에/Josa
	들어가/VV		에/JKB	들어가신/Verb
	시/EPH		들어가/VV	다/Eomi
	ㄴ다/EFN		신다/EP+EC	

[표 3-2] '아버지가방에들어가신다' 형태소 분석 결과

Hannanum	Kkma	Komoran	Mecab	Okt
나/N	나/NP	나/NP	나/NP	나/Noun
는/J	는/JX	는/JX	는/JX	는/Josa
밥/N	밥/NNG	밥/NNG	밥/NNG	밥/Noun
을/J	을/JKO	을/JKO	을/JKO	을/Josa
먹/P	먹/VV	먹/VV	먹/VV	먹는/Verb
는다/E	는/EPT	는다/EC	는다/EC	다/Eomi
	다/EFN			

[표 3-3] '나는 밥을 먹는다' 형태소 분석 결과

[그림 3-4]을 보면, Okt 형태소 분석기는 문자 수의 증가에 따른 변화가 크지 않고, [표 3-2]와 [표 3-3]를 보았을 때, 형태소 분류가 품사별로 잘 되어서 Okt 분석기를

사용하였다. Okt 형태소 분석기에서 추출 가능한 28가지 형태소 중에서 악성댓글 판별에 영향을 미칠 수 있는 빈도수가 높은 8개의 형태소를 선별한다. 악성댓글 10,000건, 비악성댓글 10,000건, 악성댓글과 비악성댓글 혼합 10,000건에서 형태소들의 출현 빈도를 확인하고, 해당 형태소를 기반으로 단어를 추출하여 자질로 사용한다.

형태소별 출현 횟수와 빈도 순위는 [표 3-4], [표 3-5], [표 3-6]과 같다.

형태소	출현 횟수	빈도 순위
Noun	238,641	1
Josa	91,709	2
Verb	71,971	3
Punctuation	30,401	4
Adjective	28,831	5
Foreign	23,834	6
Suffix	15,718	7
Adverb	9,499	8
Number	7,126	9
KoreanParticle	6,796	10
Determiner	4,385	11
Alpha	3,905	12
Conjuction	1,009	13
Exclamation	820	14
URL	560	15
ScreenName	281	16
eomi	200	17
HashTag	45	18
Email	2	19
Unknown	0	20
Proeomi	0	20

[표 3-4] 악성댓글 10,000건의 형태소별 출현빈도

형태소	출현 횟수	빈도 순위
Noun	175,897	1
Josa	80,654	2
Verb	57,128	3
Adjective	35,573	4
Punctuation	35,099	5
Suffix	10,567	6
Adverb	8,904	7
Number	4,250	8
KoreanParticle	2,924	9
Determiner	2,860	10
Alpha	2,141	11
Foreign	1,426	12
Conjuction	1,271	13
Exclamation	503	14
eomi	80	15
URL	17	16
Email	6	17
ScreenName	3	18
HashTag	0	19
Unknown	0	19
Proeomi	0	19

[표 3-5] 비악성댓글 10,000건의 형태소별 출현빈도

형태소	출현 횟수	빈도 순위
Noun	247,621	1
Josa	104,224	2
Verb	77,074	3
Punctuation	38,516	4
Adjective	37,877	5
Suffix	15,981	6
Foreign	15,575	7
Adverb	10,817	8
Number	6,420	9
KoreanParticle	5,015	10
Determiner	4,342	11
Alpha	3,292	12
Conjuction	1,453	13
Exclamation	739	14
URL	291	15
eomi	166	16
ScreenName	125	17
HashTag	37	18
Email	4	19
Unknown	0	20
Proeomi	0	20

[표 3-6] 악성댓글 , 비악성 댓글 혼합 데이터의 품사별 출현빈도

악성댓글에서는 명사, 조사, 동사, 구두점, 형용사, 외국어, 접미사가 10,000건 이상나타났고, 비악성댓글에서는 명사, 조사, 동사, 형용사, 구두점, 접미사가 10,000건 이상나타났으며, 악성댓글과 비악성댓글 혼합 데이터에서는 명사, 조사, 동사, 구두점, 형용사, 접미사, 외국어, 부사가 10,000건 이상으로 나타났다.

본 논문에서는 명사, 조사, 동사, 구두점, 형용사, 접미사, 외국어, 부사 8개의 형태소를 사용하여 문장에서 단어를 추출하여 자질로 사용하려고 하였으나, [표 3-7]을 보면, 명사, 조사, 동사, 형용사, 부사와 달리 구두점, 접미사, 외국어의 경우에는 분류에 도움이 되지 않는 데이터가 대부분을 차지하고 있어, 3개의 형태소를 제외한 5개의 형태소(품사)를 가지고 실험을 진행한다.

원문	데이트폭행범을 절대 옹호하는 건 아닌데 이런 기사나 사건 볼때마다 여자가 남자 욕하던데 그렇게따지면 여자는 남자보다 힘 없어서 남자한테 안 그랬울 뿐이지 애들한텐 폭력 행사하지 않냐? 뭐가 다른건데 ㅋㅋ 결국 자기보다 약자 공격하는 건 똑같잖아 예를들어 격투 여선수가 일반 남자랑 사귀면 댓글에 남자가 맞지 않으려면 조심해야겠다는둥 장난삼은 그런 댓글 많이 보이는 건 뭔데 ㅋㅋㅋㅋ

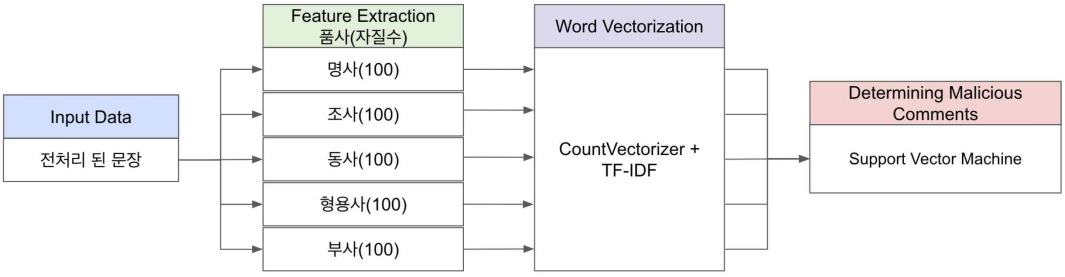
	지들 생각은 절대 안 해요. 물론 남자도 똑같이 여자입장 생각 안 하는 건 똑같더라..한심한 것들
명사	[데이트, 폭행, 범, 옹호, 건, 기사 ...]
조사	[을, 나, 마다, 가, 보다, 이지, 가, 랑, 를 ...]
동사	[하는, 따지면, 하지, 않냐, 하는, 사귀면 ...]
구두점 (제외)	[?, .., +, [,], ‘, ’, ... , ...]
형용사	[아닌데, 이런, 없어서, 그랬울, 그런, 똑같더라, ...]
접미사 (제외)	[들, ...]
외국어 (제외)	[₩n, ...]
부사	[그렇게, 결국, 많이, 물론, 똑같이, ...]

[표 3-7] 악성댓글 형태소 분류 결과

2. 품사 기반 자질 추출

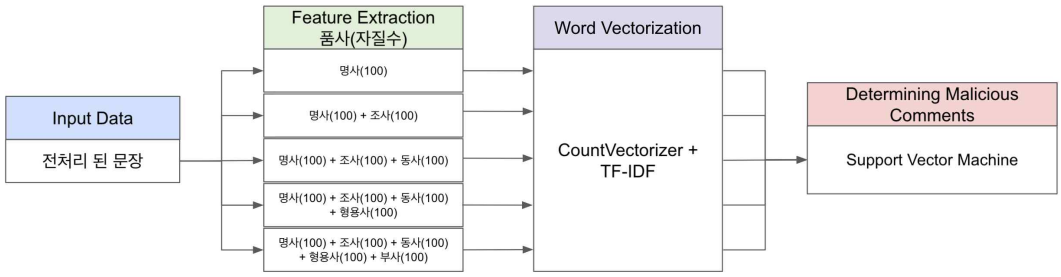
품사 기반 자질 추출 방법은 총 5가지 형태로 변화를 주며 실험한다.

첫 번째 실험은 명사, 조사, 동사, 형용사, 부사 5개 품사의 각 품사 단어로만 100개의 자질을 추출하고 악성댓글 판별 실험을 진행한다.



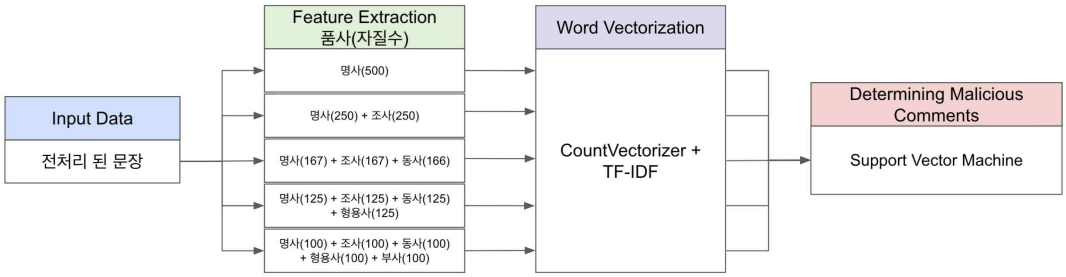
[그림 3-5] 첫 번째 실험의 프로세스 흐름도

두 번째 실험은 품사별로 100개의 자질을 추출한 후, [표 3-6]에 나오는 빈도 순위대로 더해가며 실험을 진행한다. 이 실험에서는 품사별로 100개의 자질을 가지고 있으므로 실험이 진행됨에 따라 전체 자질의 수는 100개씩 증가하게 된다.



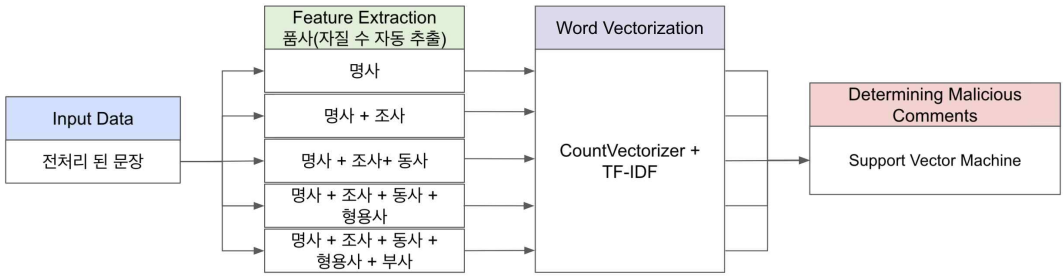
[그림 3-6] 두 번째 실험의 프로세스 흐름도

세 번째 실험은 최대 자질 수를 500개로 고정하고, 품사 기반 자질을 동일 비율로 추가하며 실험을 진행한다.



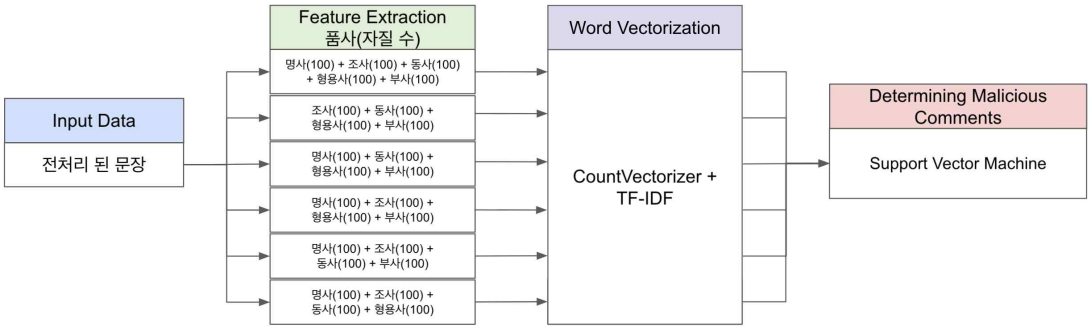
[그림 3-7] 세 번째 실험의 프로세스 흐름도

네 번째 실험은 전체 자질을 500~2500개로 변경하며 품사 유형을 추가하고, 해당 품사에 해당하는 단어의 출현 빈도에 따라 단어가 추가되도록 자질을 구성하여 실험을 진행한다. [명사 + 조사]의 경우를 예로 들면 조사에 해당하는 단어가 있더라도 출현빈도에서 명사의 단어보다 빈도가 낮다면 명사의 단어를 자질로 추출하게 된다.



[그림 3-8] 네 번째 실험의 프로세스 흐름도

다섯 번째 실험은 최대 자질 수는 500으로 고정한 상태에서 5개 품사 기반 자질을 모두 포함했을 때와 각각의 품사 기반 자질을 제거했을 때의 성능 비교 실험을 진행한다.



[그림 3-9] 다섯 번째 실험의 프로세스 흐름도

3. SVM 기반 악성댓글 판별

품사를 기반으로 추출된 단어들은 CountVectorizer와 TF-IDF를 사용하여 벡터화한 후, SVM을 사용하여 악성댓글 판별을 수행한다. CountVectorizer 설정은 analyser = word, min_df = 1로 설정한다.

```
       가슴  가장  가족  감독  감동  게이  계속  국가  국민  그냥  ...  지랄  진짜  처음  최고  평점  하나  \
0         0     0     0     0     0     0     0     0     1  ...     0     0     0     0     0     1
1         0     0     0     0     0     0     0     0     1     0  ...     0     0     0     0     0     0
2         0     0     0     0     0     0     0     0     0     0  ...     0     0     0     0     0     0
3         0     0     0     0     0     0     0     0     2     0  ...     0     0     0     0     0     0
4         0     0     0     0     0     0     0     0     1     0  ...     0     0     0     0     0     0
...      ..    ..    ..    ..    ..    ..    ..    ..    ..    ..  ...    ..    ..    ..    ..    ..    ..
79995     0     0     0     0     0     0     0     0     0     0  ...     0     0     0     0     0     0
79996     0     0     0     0     0     0     0     0     0     0  ...     0     0     0     0     0     0
79997     0     0     0     0     0     0     0     0     0     0  ...     0     0     0     0     0     0
79998     0     0     0     0     0     0     0     0     0     0  ...     0     0     0     0     0     0
79999     0     0     0     1     0     0     0     0     0     0  ...     0     0     0     0     0     0

       한국  한번  현실  홍어
0         0     0     0     0
1         0     0     0     0
2         2     0     0     0
3         1     0     0     0
4         0     0     0     0
...      ..    ..    ..    ..
79995     0     0     0     0
79996     0     0     0     0
79997     0     0     0     0
79998     0     0     0     0
79999     0     0     0     0

[80000 rows x 100 columns]
```

[그림 3-10] 품사(명사) 자질 추출 후 CountVectorizer 실행 결과

[그림 3-10]은 명사에 해당하는 단어들을 추출한 후 CountVectorizer를 수행한 결과이다. 결과 데이터에 TF-IDF를 사용하여 가중치를 적용한다. TfidfTransformer 설정은 use_idf = True이다.

	가슴	가장	가족	감독	감동	게이	계속	국가	국민	...	지랄	\
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.305265	...	0.0	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.202817	0.000000	...	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	...	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.584605	0.000000	...	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.202128	0.000000	...	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
79995	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	...	0.0	
79996	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	...	0.0	
79997	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	...	0.0	
79998	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	...	0.0	
79999	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.000000	0.000000	...	0.0	

	진짜	처음	최고	평점	하나	한국	한번	현실	홍어
0	0.0	0.0	0.0	0.0	0.318306	0.000000	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.000000	1.000000	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.000000	0.247586	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...
79995	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0
79996	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0
79997	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0
79998	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0
79999	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0

[그림 3-11] 품사(명사) 자질 추출 후 TF-IDF 가중치 적용 결과

[그림 3-11]은 TF-IDF를 통한 가중치의 결과이다. 결과 데이터를 SVM 머신러닝 모델의 자질값으로 사용하여 악성댓글 판별을 수행한다. SVM 머신러닝 모델은 kernel, C, gamma 등의 파라미터 값에 따라 결과값이 달라지는데, kernel은 결정경계 모형, C는 이상치 허용값, gamma는 결정경계의 유연한 정도를 결정한다. C가 클수록 Hard Margine 형태를 보이고, gamma가 클수록 구불구불한 구분선이 그려진다. 본 논문에서는 SVM의 파라미터 값에 따른 성능향상이 목적이 아닌 품사 기반 자질이 성능에 미치는 영향을 판단하고자 하였기에 kernel = linear, C = 1.0, gamma = auto 로 동일하게 유지한 상태로 실험을 진행한다.

```
# kernel - 결정경계모형, C - 이상치 허용값, gamma - 구분선의 유연한 정도
SVM = svm.SVC(kernel='linear', C=1.0, gamma='auto') # SVM 파라미터 셋팅
SVM.fit(X_train_tf, t_y)
# X_train_tf : TF-IDF 가 적용된 결과값
# t_y : Training Data 에서 0, 1로 라벨링된 데이터

predictions_SVM = SVM.predict(Test_X_Tfidf) # 판별 수행

print(predictions_SVM)

# sklearn 함수를 사용한 정확도, 정밀도, 재현율, F1-score 출력
print("SVM Accuracy Score -> ",accuracy_score(predictions_SVM, p_y))
print("SVM Precision Score -> ",precision_score(p_y, predictions_SVM, average="macro"))
print("SVM Recall Score -> ",recall_score(p_y, predictions_SVM, average="macro"))
print("SVM F1-score Score -> ",f1_score(p_y, predictions_SVM, average="macro"))
```

[표 3-8] SVM을 사용한 악성댓글 판별 코드

```
SVM Accuracy Score ->  0.70425
SVM Precision Score ->  0.7389551842798399
SVM Recall Score ->  0.70425
SVM F1-score Score ->  0.6931069265613619
2021-06-02 13:54:54.040884
```

[그림 3-12] 품사(명사) 자질 추출 후 SVM 판별 결과

[그림 3-12]는 TF-IDF의 결과값을 사용한 SVM 모델의 판별 결과이다. [0 0 0 ... 0 1 1] 은 실험 데이터의 악성/비악성 판별결과를 나타내며, SVM으로 시작하는 부분은 판별 성능에 관한 결과이다.

Ⅳ. 실험 및 결과

본 장에서는 악성댓글 판별 성능 향상을 위한 품사 기반 자질 추출방법을 실험하고, 품사 기반 자질에 따른 성능 변화를 정확도(Accuracy), 재현율(Recall), 정밀도(Precision), F1-score를 통해 확인한다.

본 연구에서 실험에 사용한 환경은 아래와 같다.

Division		Contents
HW	CPU	Intel Core i9 2.4GHz 8 Core
	GPU	AMD Radeon Pro 5500M 8GB
	RAM	64GB 2667MHz DDR4
	HDD	2TB SSD
SW	OS	macOS Big Sur 11.2.3
	Program Language	Python 3.5

[표 4-1] 실험 환경

A. 데이터 셋

본 절에서는 실험에 사용되는 데이터 셋에 대하여 설명한다. 데이터 셋은 Kaggle에서 제공하는 Korean Hate Speech Dataset(https://www.kaggle.com/captainnemo9292/korean-hate-speech-dataset)을 사용하였다. 해당 데이터셋에서 악성댓글은 욕설과 비난이 심각한 수준으로 나타나는 일베저장소-일간베스트의 자료들을 수집하였고, 비악성댓글은 네이버 영화의 댓글 자료를 수집하였다. 일베저장소의 경우에 정치적인 발언이나 성적인 발언들이 매우 높은 강도로 나타나고 있어 악성댓글의 표본으로 삼기에 적합하다.

전체 데이터의 수는 190,000건이고, 그 중 악성댓글은 90,000건, 비악성댓글은

100,000건으로 구성되어 있으며, 악성댓글은 0, 비악성댓글은 1로 라벨링 되어 있다. [표 4-2]는 해당 데이터셋의 샘플데이터이다.

악성/비악성	원문	라벨링
악성	좇선족 벌레들은 인간이 아니기 때문이다. 니넨 범지구적 해충이야 이 씹버러지 새끼들아	0
악성	저거 고소된다.걍 좇된거다. 절대 진짜 성인이든 미성년자든 뭐든 여자는 절대 인터넷상으로 건들지마라. 조또 아닌것도 성범죄카테고리로엮여서 남자인생 걍 믹서기로갈려버린다.	0
악성	@익명_1248ce 니나 개소리하지마라 여론이나 방송국에 정부가 개입하는거 자체가 불법이고 위법이고 이런걸 방송국 노조들이 낑세채면 바로 방송파업이나 이명박 아웃같은 좇같은 방송만 해데는데 뭐?? 정부가 방송국을 손봐 ㅋㅋㅋㅋㅋ	0
비악성	오해와 의심이 만드는 아이러니의 공간을 파고드는 코엔 형제의 농담 센스	1
비악성	좀 더 어렸다면 나도 믿어 버렸을꺼야. 지금은 믿지 않지만 믿고 싶다~	1

[표 4-2] 실험에 사용한 데이터셋 샘플

본 실험에서는 전체 190,000건 중에서 글자 수가 3~300자에 해당하는 데이터만 필터링한 후, 악성댓글 50,000건, 비악성댓글 50,000건을 추출하여 실험에 사용한다.

B. 실험 평가 및 분석

1. 실험 평가 방법

본 논문의 성능평가는 혼동행렬에 기반을 둔 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F1-score 4가지 지표를 사용한다.

		Predicted	
		Positive	Negative
Measured	Positive	TP	FN
	Negative	FP	TN

[표 4-3] 혼동행렬

[표 4-3]은 정확도, 재현율, 정밀도, F1-score를 계산하기 사용하는 혼동행렬이다.

TP(True Positive)는 실제 True인 값을 모델이 True로 올바르게 예측한 것이다.
TN(True Negative)은 실제 False인 값을 모델이 False로 올바르게 예측한 것이다.
FP(False Positive)는 실제 False인 값을 모델이 True로 잘못 예측한 것이다.
FN(False Negative)은 실제 True인 값을 모델이 False로 잘못 예측한 것이다.

정확도(Accuracy)는 전체 데이터에 대해서 모델이 얼마나 정확하게 예측하는지를 나타내는 지표이다. 실제 True인 값을 모델이 True로, 실제 False인 값을 모델이 False로 정확하게 예측한 결과값의 비율이다.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FN+FP)} \qquad (2)$$

정밀도(Precision)는 모델의 예측값이 얼마나 정확한지를 나타내는 지표이다. 모델이 True로 예측한 값 중에서 실제로 True인 값의 비율을 나타낸다.

$$Precision = \frac{TP}{(TP+FP)} \qquad (3)$$

재현율(Recall)은 실제로 True인 데이터 중에서 모델이 True로 예측한 값의 비율을 나타내는 지표이다.

$$Recall = \frac{TP}{(TP+FN)} \qquad (4)$$

F1-score는 정밀도와 재현율의 조화평균을 나타내는 지표이다.

$$F1\text{-}score = \frac{2*Recall*\mathrm{Precision}}{(Reall+\mathrm{Precision})} \qquad (5)$$

위의 4가지 성능 평가 지표는 sklearn에서 제공하는 API를 사용하여 계산하였다.

```
# Use accuracy_score function to get the accuracy
print("SVM Accuracy Score -> ",accuracy_score(predictions_SVM, p_y))
print("SVM Precision Score -> ",precision_score(p_y, predictions_SVM, average="macro"))
print("SVM Recall Score -> ",recall_score(p_y, predictions_SVM, average="macro"))
print("SVM F1-score Score -> ",f1_score(p_y, predictions_SVM, average="macro"))


>>> SVM Accuracy Score -> 0.566
>>> SVM Precision Score -> 0.5782747300707793
>>> SVM Recall Score -> 0.566000000001
>>> SVM F1-score Score -> 0.5482912085395859
```

[표 4-4] 모델의 성능평가 결과 출력

2. 실험 결과 분석

5개의 품사를 기반으로 추출한 자질들을 다양한 형태의 실험을 통해서 품사 기반 자질이 악성댓글 판별모델에 미치는 영향을 알아본다.

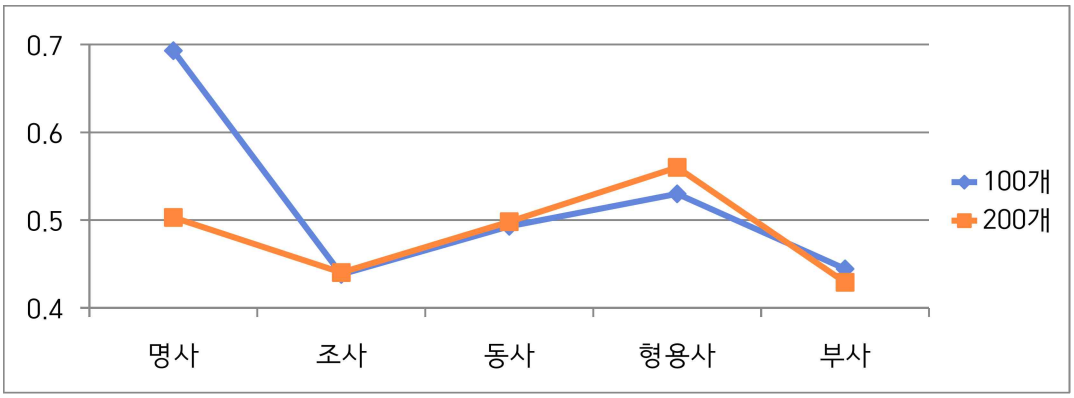
첫 번째 실험은 명사, 조사, 동사, 형용사, 부사 5개 품사에 해당하는 단어로만 100개씩의 자질을 추출하고 악성댓글 판별 실험을 진행하였다. 결과는 [표 4-5], [표 4-6], [그림 4-1]과 같다.

	Accuracy	Precision	Recall	F1-score
명사	0.7042	0.7389	0.7042	0.6931
조사	0.5127	0.5271	0.5127	0.4382
동사	0.5567	0.6138	0.5567	0.4930
형용사	0.5791	0.6359	0.5791	0.5300
부사	0.5420	0.6413	0.5420	0.4445

[표 4-5] 품사별 자질 추출 후 성능 비교 (max_features = 100)

	Accuracy	Precision	Recall	F1-score
명사	0.5278	0.5347	0.5278	0.5031
조사	0.5229	0.5558	0.5229	0.4403
동사	0.5605	0.6201	0.5605	0.4982
형용사	0.6038	0.6723	0.6038	0.5601
부사	0.5340	0.6283	0.5340	0.4292

[표 4-6] 품사별 자질 추출 후 성능 비교 (max_features = 200)



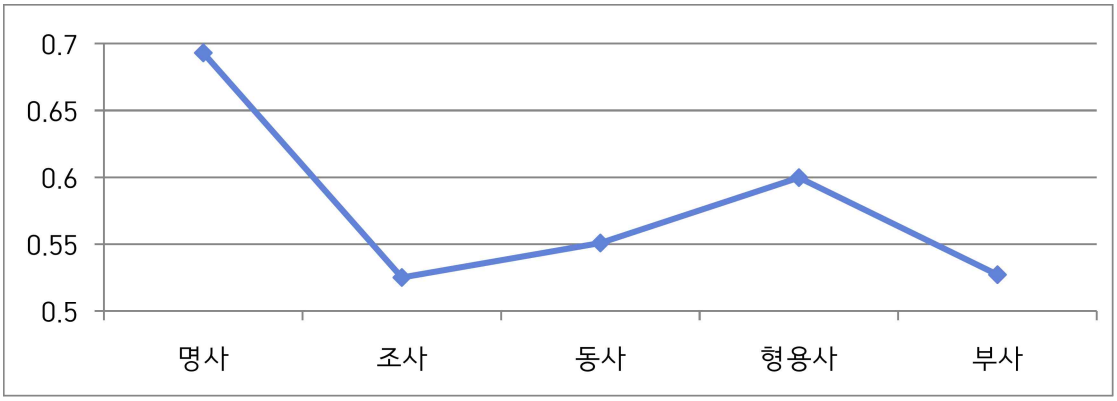
[그림 4-1] 품사별 성능 실험 결과(자질 수 100개, 200개)

이 실험에서는 명사와 형용사 2개 품사 자질을 사용하였을 때 좋은 성능을 내는 것을 알 수 있다.

두 번째 실험은 품사별로 100개의 자질을 추출한 후, [표 3-6]에 나오는 출현 빈도순으로 더해가며 실험을 진행한다.

	전체 자질수	Accuracy	Precision	Recall	F1-score
명사	100	0.7042	0.7389	0.7042	0.6931
조사	200	0.5520	0.5673	0.5520	0.5251
동사	300	0.5704	0.5852	0.5704	0.5509
형용사	400	0.6079	0.6173	0.6079	0.5998
부사	500	0.5514	0.5646	0.5514	0.5272

[표 4-7] 품사별 자질 추가에 따른 성능 변화 분석



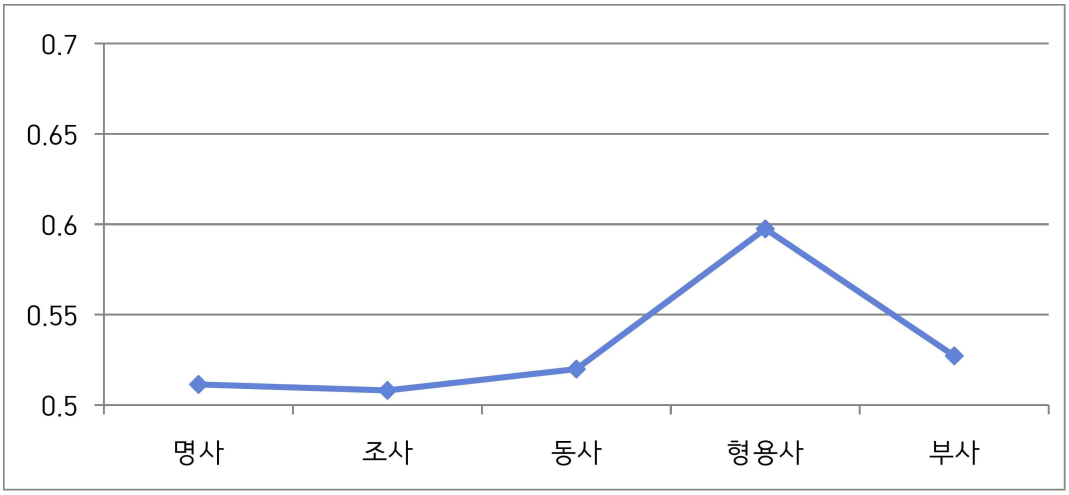
[그림 4-2] 품사별 자질 수 100개 추가 실험 결과

이 실험에서는 명사와 형용사 2개의 품사 자질을 사용했을 때 높은 성능이 나타났다.

세 번째 실험은 최대 자질 수를 500개로 고정하고, 품사 기반 자질을 동일 비율로 추가하며 실험을 진행하였고, 결과는 [표 4-8], [그림 4-3]과 같다.

	품사별 자질 수	Accuracy	Precision	Recall	F1-score
명사	명사 500개	0.5374	0.5474	0.5374	0.5114
조사	명사 250개 조사 250개	0.5367	0.5478	0.5367	0.5081
동사	명사 167개 조사 167개 동사 166개	0.5355	0.5407	0.5355	0.5199
형용사	동사 125개 조사 125개 동사 125개 형용사 125개	0.6112	0.6287	0.6112	0.5975
부사	명사 100개 조사 100개 동사 100개 형용사 100개 부사 100개	0.5514	0.5646	0.5514	0.5272

[표 4-8] 품사의 비율에 따른 성능 변화 분석 (자질 수 : 500)



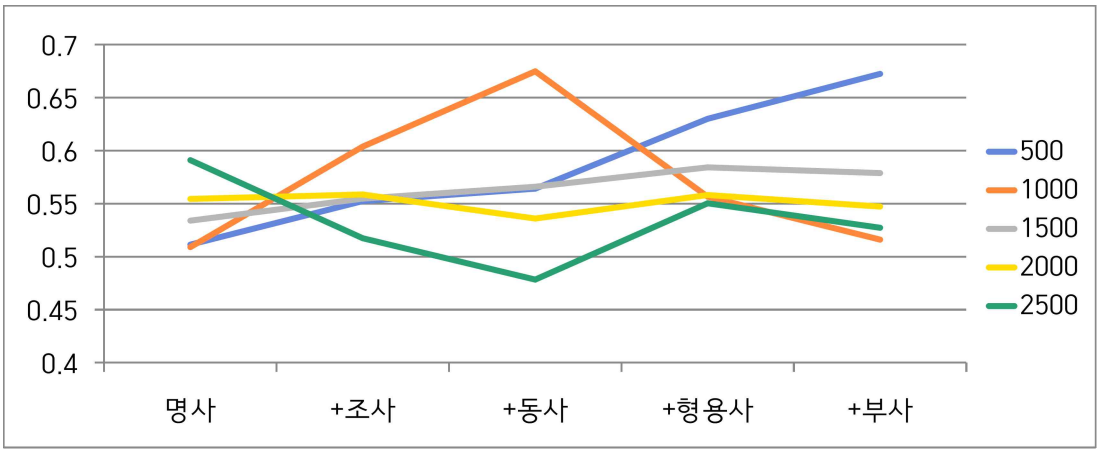
[그림 4-3] 품사별 동일 자질 수 추가 실험 결과

이 실험에서는 동일 비율로 특정 품사가 추가되는 경우에 형용사 기반 자질이 추가되었을 때 큰 폭의 성능 향상이 일어남을 알 수 있었다.

네 번째 실험은 해당 품사에 해당하는 단어의 출현 빈도에 따라 단어가 추가되도록 자질을 구성하여 실험을 진행하였다.

자질 수	품사 패턴	Accuracy	Precision	Recall	F1-score
500	명사	0.5374	0.5474	0.5374	0.5114
	명사+조사	0.5693	0.5814	0.5692	0.5525
	명사+조사+동사	0.5806	0.5948	0.5806	0.5642
	명사+조사+동사 +형용사	0.6370	0.6481	0.6370	0.6301
	명사+조사+동사 +형용사+부사	0.6782	0.6913	0.6782	0.6725
1000	명사	0.5437	0.5609	0.5437	0.5090
	명사+조사	0.6177	0.6367	0.6177	0.6039
	명사+조사+동사	0.6791	0.6888	0.6791	0.6749
	명사+조사+동사 +형용사	0.5645	0.5696	0.5646	0.5566
	명사+조사+동사 +형용사+부사	0.5356	0.5425	0.5356	0.5161
1500	명사	0.5624	0.5825	0.5624	0.5340
	명사+조사	0.5763	0.5946	0.5763	0.5548
	명사+조사+동사	0.5781	0.5878	0.5781	0.5660
	명사+조사+동사 +형용사	0.5924	0.6002	0.5924	0.5843
	명사+조사+동사 +형용사+부사	0.5903	0.6012	0.5903	0.5789
2000	명사	0.5757	0.5935	0.5757	0.5545
	명사+조사	0.5800	0.5990	0.5800	0.5587
	명사+조사+동사	0.5634	0.5830	0.5634	0.5360
	명사+조사+동사 +형용사	0.5678	0.5743	0.5678	0.5582
	명사+조사+동사 +형용사+부사	0.5615	0.5702	0.5615	0.5473
2500	명사	0.6075	0.6279	0.6075	0.5911
	명사+조사	0.54655	0.5613	0.5465	0.5174
	명사+조사+동사	0.5028	0.5035	0.5028	0.4784
	명사+조사+동사 +형용사	0.5709	0.5867	0.5709	0.5504
	명사+조사+동사 +형용사+부사	0.5488	0.5596	0.5488	0.5273

[표 4-9] 출현빈도에 따른 실험 결과 (1회 이상 출현, 자질 수 : 500-2500)

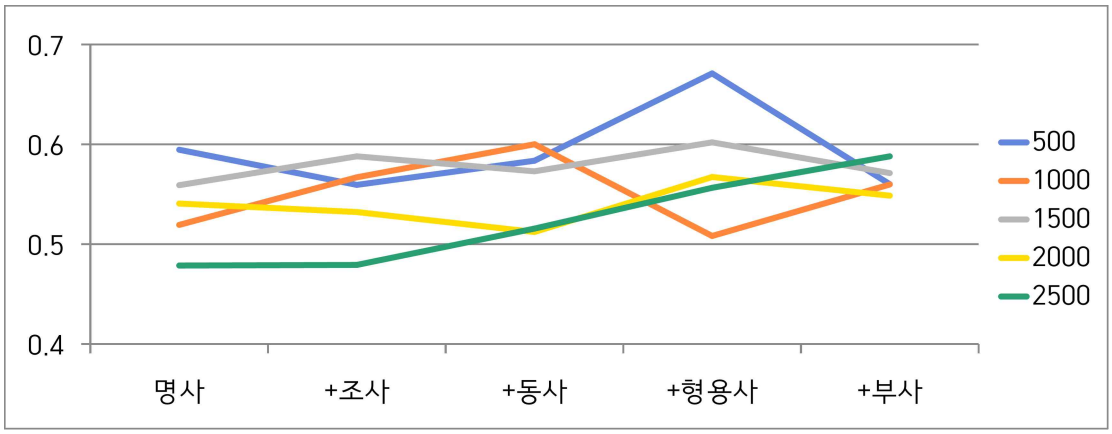


[그림 4-4] 품사 출현빈도와 자질 수에 따른 실험 결과 (출현 1 이상)

[그림 4-4]의 결과에서 유의미한 데이터를 얻지 못하여 CountVectorizer의 min_df를 1에서 2로 변경한 후에 다시 실험을 진행하였다.

자질 수	품사 패턴	Accuracy	Precision	Recall	F1-score
500	명사	0.6069	0.6214	0.6069	0.5947
	명사+조사	0.5720	0.5824	0.5726	0.5595
	명사+조사+동사	0.5984	0.6145	0.5948	0.5837
	명사+조사+동사 +형용사	0.6749	0.6831	0.6749	0.6712
	명사+조사+동사 +형용사+부사	0.5697	0.5763	0.56975	0.5602
1000	명사	0.5443	0.5558	0.5443	0.5194
	명사+조사	0.5764	0.5835	0.5764	0.5671
	명사+조사+동사	0.6082	0.6172	0.6082	0.6002
	명사+조사+동사 +형용사	0.5174	0.5188	0.5174	0.5083
	명사+조사+동사 +형용사+부사	0.5706	0.5782	0.5706	0.5599
1500	명사	0.5837	0.6075	0.5837	0.5592
	명사+조사	0.6021	0.6181	0.6021	0.5881
	명사+조사+동사	0.5873	0.6006	0.5872	0.5731
	명사+조사+동사 +형용사	0.6147	0.6312	0.6147	0.6021
	명사+조사+동사 +형용사+부사	0.5837	0.5945	0.5837	0.5713
2000	명사	0.5656	0.5837	0.5656	0.5408
	명사+조사	0.5591	0.5767	0.5591	0.5323
	명사+조사+동사	0.54	0.5517	0.54	0.5123
	명사+조사+동사 +형용사	0.5815	0.5936	0.5815	0.5674
	명사+조사+동사 +형용사+부사	0.5665	0.5790	0.5665	0.5487
2500	명사	0.5144	0.5198	0.5144	0.4787
	명사+조사	0.5220	0.5328	0.5220	0.4793
	명사+조사+동사	0.5394	0.5489	0.5394	0.5157
	명사+조사+동사 +형용사	0.5671	0.5740	0.5671	0.5566
	명사+조사+동사 +형용사+부사	0.6013	0.6162	0.6013	0.5881

[표 4-10] 출현빈도에 따른 실험 결과 (2회 이상 출현, 자질 수 : 500-2500)



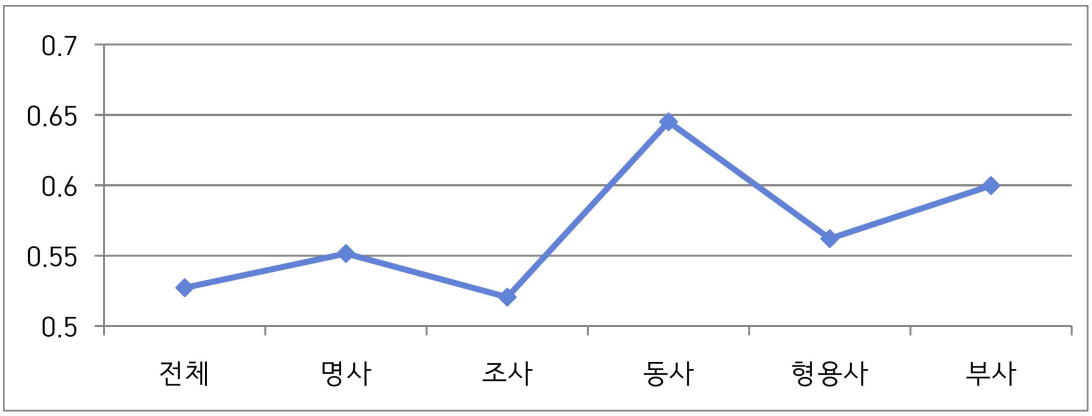
[그림 4-5] 품사 출현빈도와 자질 수에 따른 실험 결과 (출현 2 이상)

[그림 4-5]를 보면 500, 1500, 2000 실험구간에서 형용사 자질이 성능에 도움이 되는 것을 알 수 있다.

다섯 번째 실험은 최대 자질 수는 500으로 고정한 상태에서 5개 품사 기반 자질을 모두 포함했을 때와 각각의 품사 기반 자질을 제거했을 때의 성능 비교 실험을 진행하였다.

	Accuracy	Precision	Recall	F1-score
전체 품사	0.5514	0.5646	0.5514	0.5272
명사 제거	0.5901	0.6375	0.5901	0.5515
조사 제거	0.5421	0.5513	0.5421	0.5205
동사 제거	0.6575	0.6830	0.6575	0.6452
형용사 제거	0.5823	0.6010	0.5823	0.5621
부사 제거	0.6079	0.6173	0.6079	0.5998

[표 4-11] 특정 품사 제거 시의 성능 변화 분석 (자질 수 : 500)



[그림 4-6] 특정 품사 제거 시 실험 결과

[그림 4-6]을 보면, 동사와 부사 자질을 제거 시에 성능이 올라가는 것을 알 수 있다.

본 논문의 실험 결과를 요약하면 아래 [표 4-12]와 같다. 형용사가 대부분의 실험에서 좋은 성능을 나타내는 것을 알 수 있다.

실험번호	자질 수	좋은 성능을 보이는 품사 유형 (1이 높은 정확도)				
		1	2	3	4	5
1-1	100	명사	형용사	동사	부사	조사
1-2	200	형용사	명사	동사	조사	부사
2	100 ~ 500	명사	형용사	동사	부사	조사
3	500	형용사	부사	동사	명사	조사
4-1	500	부사	형용사	동사	조사	명사
	1000	동사	조사	형용사	부사	명사
	1500	형용사	부사	동사	조사	명사
	2000	조사	형용사	명사	부사	동사
	2500	명사	형용사	부사	조사	동사
4-2	500	형용사	명사	동사	부사	조사
	1000	동사	조사	부사	명사	형용사
	1500	형용사	조사	동사	부사	명사
	2000	형용사	부사	명사	조사	동사
	2500	부사	형용사	동사	조사	명사
실험번호	자질 수	품사 제거 시 성능 순위				
5	500	동사	부사	형용사	명사	조사

[표 4-12] 실험결과

본 논문에서는 악성댓글 판별 성능 향상에 있어 품사 기반 자질 추출방법의 연구를 통해 어떠한 품사가 성능에 영향을 미치는지에 대한 실험을 진행하였다.

본 논문의 실험결과를 살펴보면, 문장에서 단어를 추출 시 명사만으로 추출했을 때보다 형용사와 조합하여 추출했을 때 더 높은 정확도를 보여주었다. 또한, 동사를 제거하는 경우에 다른 품사를 제거한 경우보다 성능이 올라가는 것을 알 수 있었다. 이를 토대로 [명사+형용사-동사] 형태로 한국어 데이터를 필터링 후 판별모델을 구축하면 기존보다 좋은 성능을 낼 수 있을 것으로 예상한다.

본 논문에서 정확하게 분석하지 못한 부분이 몇 가지 있다. 먼저, scikit-learn에서 제공하는 CountVectorizer를 사용하여 단어의 카운팅 작업을 하였는데, Count Vectorizer의 특징 중 하나인 1문자로 구성된 요소를 제외한다는 점이 실험결과에 미치는 영향을 분석하지 못하였다. 또한, 실험결과가 다른 선행 연구에 비해 낮은 성능을 보여주었는데, 이는 형태소 분석기와 CountVectorizer를 이용한 품사 기반 자질 추출 단계만 변형하며 실험했기 때문으로 보인다. TF-IDF와 SVM 부분의 최적화 작업을 거치면 보다 성능을 향상할 수 있을 것이다.

V. 결론 및 향후 연구

본 논문에서는 악성댓글 판별 성능 향상을 위한 품사 기반 자질추출방법에 대한 실험을 진행하였다. 실험방법은 Okt 형태소 분석기를 통한 형태소 분류 작업 이후에 특정 품사에 해당하는 데이터를 추출하고, CountVectorizer를 사용하여 벡터화한 후에 TF-IDF를 통한 가중치 적용 후 SVM을 사용하여 실험하였다. 10,000개의 데이터에서 10,000번 이상 나타나는 8개 형태소(명사, 조사, 동사, 구두점, 형용사, 접미사, 외국어표현, 부사) 중에서, 데이터가 무의미한 3개 형태소(구두점, 접미사, 외국어표현)를 제거하고 5개 형태소(품사)를 가지고 실험을 진행하였다.

실험 결과로는 명사만을 추출했을 때보다, 형용사를 자질 추출에 포함 시키는 경우가 성능 향상에 도움이 되는 것을 볼 수 있었으며, 동사는 제외하는 것이 판별 모델 성능 향상에 도움이 되는 것으로 나타났다.

본 논문에서 분석하지 못한 아쉬운 부분도 있다. 본 논문의 실험에서는 Count Vectorizer를 사용하여 단어의 출현빈도를 카운팅하였으나, 길이가 1인 글자는 제거되었다. 따라서 추후에 길이가 1인 글자가 성능변화에 미치는 영향을 분석할 필요가 있다. 또한, 실험결과가 다른 선행 연구에 비해 낮은 성능을 보여주었는데, 이는 형태소 분석기와 CountVectorizer를 이용한 품사 기반 자질 추출단계만 변형하며 실험했기 때문으로 보인다. TF-IDF 와 SVM 부분의 최적화 작업을 거치면 보다 성능을 향상할 수 있을 것이다.

참고 문헌

[1] D. Gillmor. 2004. “INTRODUCTION TO THE PAPERBACK EDITION” in We the Media: Grassroots Journalism by the People, for the People, xvi O’Reilly

[2] 안태형. 2013. “악성 댓글의 범위와 유형.” 우리말학회 우리말연구 32 (0): 109-131.

[3] 김진우, 조혜인, 이봉규. 2019. “인공신경망을 적용한 악성댓글 분류 모델들의 성능 비교.” 한국디지털콘텐츠학회 논문지 20 (7): 1429-1437

[4] 성대경, 이현우, 이창영, 김아영, 박성배. 2014. “확률 기반 악성댓글 판별.” 한국정보처리학회 추계학술발표대회 21 (2): 905-908

[5] 이중원. 2017. “CyberBullyWordNet:오피니언 마이닝에서의 비난 댓글 파악을 위한 방법 개발.” 석사학위, 경희대학교 대학원 경영학과.

[6] 하예람. 2020. “비속어 분포를 고려한 뉴스 댓글 필터링 방법.” 석사학위, 부산대학교 대학원 전기전자컴퓨터공학과.

[7] 김세한. 2016. “인공신경망을 이용한 인터넷 악성 댓글 탐지 기법.” 석사학위, 숭실대학교 소프트웨어특성화대학원 소프트웨어학과.

[8] 이성록, 조민제, 조수완, 김혜정 “악성댓글분류를 위한 데이터 전처리 인공신경망 모델 성능 비교.” 대한전자공학회 하계학술대회 논문집 2020 (8): 2003-2005

[9] 김유영, 송민. 2016. “영화 리뷰 감성분석을 위한 텍스트 마이닝 기반 감성 분류기 구축.” 한국지능정보시스템 학회 지능정보연구 22 (3): 71-89

[10] 김천중. 2015. “SVM을 이용한 SNS 댓글 적합성 판단.” 한국정보과학회 동계학술발표회 논문집 2015 (12): 775-777

[11] 김묘실. 2006. “SVM을 이용한 악성 댓글 판별 시스템의 설계 및 구현.” 석사학위, 국민대학교 교육대학원 전자계산 교육전공.

[12] 김현정, 윤영미, 이병문. 2011. “향상된 FFP(Feature Frequency Profile)을 활용한 악성 댓글의 판별시스템.” 한국정보기술학회논문지 9 (1): 207-216

[13] 이현상, 이희준, 오세환. 2020. “하이웨이 네트워크 기반 CNN모델링 및 사전 외 어휘 처리 기술을 활용한 악성 댓글 분류 연구” 한국정보시스템학회 정보시스템연구 29 (3): 103-117

[14] 정민철, 이지현 오하영. 2020. “앙상블 머신러닝 모델 기반 유튜브 스팸 댓글 탐

지” 한국정보통신학회 한국정보통신학회논문지 24 (5): 576-583

[15] 어동선. 2015. “빅데이터를 이용한 텍스트마이닝 기법의 성능 비교.” 석사학위, 인제대학교 일반대학원 데이터정보학과.

[16] “서포트 벡터 머신.” (2021.05.31.) 위키피디아. 2020년 7월 10일 수정, https://ko.wikipedia.org/wiki/서포트_벡터_머신