Journal of Korea Multimedia Society Vol. 22, No. 6, June 2019(pp. 695-705)
https://doi.org/10.9717/kmms.2019.22.6.695

변형된 한글 금칙어에 대한 실시간 필터링 시스템

김찬우†, 성미영††

Realtime Word Filtering System against Variations of Censored Words in Korean

ChanWoo. Kim†, Mee Young Sung††

ABSTRACT

The level of psychological damage caused by verbal abuse among cyberbully victims is very serious. It is going to introduce a system that determines the level of sanctions against chatting in real time using the automatic prohibited words filtering based on artificial neural network. In this paper, we propose a keyword filtering method that detects the modified prohibited words and determines whether the corresponding chat should be sanctioned in real time, and a real-time chatting screening system using it. The accuracy of filtering through machine learning was improved by processing data in advance through coding techniques that express consonants and vowels of similar pronunciation at close distances. After comparing and analyzing Mahalanobis-based clustering algorithms and artificial neural network-based algorithms, algorithms that utilize artificial neural networks showed high performance. If it is applied to Internet chatting, comments or online games, it is expected that it will be able to filter more effectively than the existing filtering method and that this will ease communication inconvenience due to existing indiscriminate filtering methods.

Key words: Filtering Vulgar Words, Online Chat Censorship, Korean Encoding, Natural Language Processing, Artificial Neural Network

1. 서 론

온라인 게임이나 인터넷 개인 방송과 같은 인터넷 응용에서 다른 사람과 의사소통을 하며 참여하는 것은 더욱 큰 재미를 느낄 수 있게 해주는 요소이다. 하지만 온라인 채팅은 익명성을 악용하여 폭력적으로 변할 수 있다. 실제로 한국인터넷진흥원에서 실시한 '2011년 인터넷윤리문화 실태조사'에 따르면 사이버폭력 피해 유형 중 욕설이나 비속어가 사용된 언어폭력을 경험한 사람이 국내 인터넷 이용자 중 41%에 달한다는 조사 결과가 있다[1]. 온라인 채팅에서의 언어폭력은 해당 콘텐츠의 재미요소를 반감시킬 수 있으며, 피해자는 정신적 피해를 입기도 한다. 이러한 피해를 사전에 막기 위한 노력이 필요한 시점이다.

온라인상에서의 언어폭력을 제재하기 위해 인터넷 개인방송 플랫폼에서는 매니저, 관리자와 같은 유저가 직접 채팅에 대한 관리를 수행한다. 이는 가장 정확하고 안전한 방법이지만, 인력이 소모된다는 큰 단점이 존재한다.

다른 방법으로는 금칙어 필터링을 적용하는 것이

※ Corresponding Author : Mee Young Sung, Address: (21012) Academy-ro 119, Yeonsu-gu, Incheon, Korea, TEL : +82-32-835-8496, FAX : +82-32-835-0780, E-mail : mysung@inu.ac.kr
Receipt date : Apr 29, 2019, Revision date : May 29, 2019
Approval date : May 30, 2019

† Department of Computer Science & Engineering, Graduate School, Incheon National University (E-mail : dolguso@naver.com)
†† Department of Computer Science & Engineering, Incheon National University
※ This work was supported by Incheon National University Research Grant in 2016 (No. 2016-2264).

다. 금칙어 필터링은 이미 온라인 게임, 인터넷 개인 방송 플랫폼, 인터넷 기사 댓글, 온라인 채팅 등에 적용되어 있다. 대부분의 응용에서는 금칙어 리스트에 따른 단순한 단어 매칭 방식으로 필터링을 수행하는데, 이는 두 가지의 큰 문제점을 갖고 있다. 첫 번째 문제점은 금칙어의 변형에 취약하다는 것이다. 실제로 온라인 채팅에서는 이러한 필터링을 우회하기 위해 금칙어를 변형하여 사용하고 있다. 예를 들면 '바보'가 금칙어로 지정된 경우, '바부'와 같이 변형하여 필터링을 우회할 수 있다. 또 다른 문제점은 무분별한 필터링으로 인해 정상적인 의사소통을 방해하는 것이다. '염병'이 비속어로 지정된 경우 '전염병이 돌고 있어'라는 채팅은 '전**이 돌고 있어'로 필터링 되어 정상적인 의사소통이 어려워질 수 있다. 본 연구에서는 이러한 단점을 해결하고 변형된 금칙어를 자동검출해낼 수 있는 개선된 금칙어 필터링 기법과 이를 이용하는 실시간 채팅 검열 시스템을 제안한다.

2. 연구 배경

금칙어의 변형에 취약하다거나, 무분별한 필터링으로 인해 정상적인 의사소통이 힘든 문제점을 개선하기 위해 수행된 여러 관련 연구가 있다. 한국게임산업진흥원은 국립국어원과 '게임언어 건전화를 위한 공동협력 협약'을 맺고 '게임언어 건전화 지침서'를 발간하였다[2]. 8508개의 금칙어를 폭력적, 선정적, 차별적, 사행성 유발의 네 가지 선정 기준에 따라 분류하였고, 이들 금칙어를 대표형과 변형형으로 묶어 금칙어 선정에 표준화된 기준을 제시하였다. 다른 연구로는 욕설을 변형하여 필터링을 우회하는 문제점을 개선하기 위해 변형 욕설의 표준화를 제시하고 자소 단위의 반 전역 행렬(semi-global alignment)을 이용하여 변형 욕설을 효율적으로 필터링하는 알고리즘을 제시하는 연구가 수행되었다[3]. 실험 결과에서 일반 욕설 1672개의 단어를 입력하여 1648개를 검출해내는 98.5%의 검출 성능을 보인다고 하지만, 이 방법은 모든 금칙어를 수작업으로 입력해야 하며 많은 계산이 필요하다는 단점이 있다. 서포트 벡터 머신(support vector machine)[4]을 사용한 지도 학습(supervised learning) 방식의 비속어 필터링 시스템을 제시한 연구도 있다[5]. 이 연구는 정상 문장과 비속어가 포함된 문장으로 나누어 서포트 벡터 머신 분류기를 학습시킨 후, 학습 과정에서 얻어진 자질을 기반으로 비속어 분류를 수행한다. 분류 결과, 변형되지 않은 비속어에 한해서 86%의 정확성을 보인다. 하지만 변형된 비속어에 대해서 충분히 학습되지 않았을 경우 제대로 분류하지 못하는 단점이 있다. 텍스트 마이닝과 인공신경망을 이용해 문서의 대표적인 내용을 추출하고 요약하는 연구도 진행되었다[6]. 신문 기사를 텍스트화한 비정형 데이터로부터 수치 데이터를 추출하여 문서의 중요 내용을 추출하는 방법을 활용하였으나, 정확도가 60% 이하로 실제 상황에 이용되기엔 부족한 정확도를 보였다. 욕설 문장 확인을 위한 신경망 학습에 필요한 데이터가 부족한 경우 임의로 생성한 문장을 신경망에 학습하는 연구도 진행된 바 있다[7]. 데이터 부족 및 학습 데이터의 욕설 분포 불균형을 해결하여 신경망의 성능을 좀 더 상승시켰다. 해당 연구에서는 영어 비속어를 주제로 진행되었으며, 한국어 비속어 연구가 많지 않은 상황이다. 본 논문에서는 그동안 부족했던 한국어 비속어 연구를 진행할 예정이다. 또한, 핸드폰 메시지의 문자 조합만으로 변형된 금칙어를 필터링하는 연구[8], 한의학에서 환자가 말하는 증상을 자연어 처리 및 형태소 분석을 통해 자동으로 증상을 진단하는 연구[9]와 사용자의 TV 시청 패턴을 분석하여 채널 필터링 연구[10]도 진행된 바 있다.

3. 제안하는 방법

3.1 시스템 모델

본 연구에서는 금칙어의 변형을 예측하여 검출해내고, 해당 금칙어가 정상적인 문장 일부인지 아닌지를 구별할 수 있는 개선된 필터링 시스템에 관하여 연구한다. 한글 금칙어 필터링을 위해서는 자연어에 대한 분석이 필수적이다. 자연어 중에서도 한글 인터넷 용어에 대한 분석을 위해서는 금칙어의 다양한 변형을 고려한 인접 자소 인코딩 처리 기법이 필요하다.

금칙어 변형의 대표형을 만들어내기 위해 군집화가 되어야 하며 금칙어 변형의 대표형을 군집화하려면 수치 형태의 입력 데이터가 필요하다. 따라서 금칙어를 군집화하려면 입력 데이터가 우선 수치로 변환되어야 하므로 한글을 인접 자소 인코딩 알고리즘으로 코드화하였다. 코드화한 데이터를 통해 해당 금칙어가 정상적인 문장 일부인지 악의적인 의미로 사

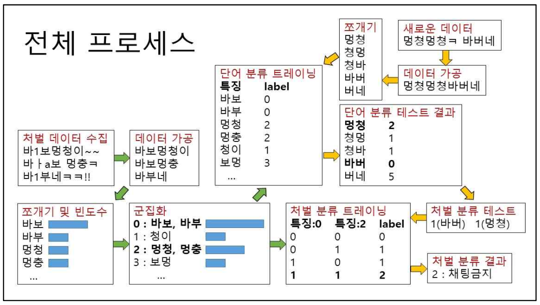

Fig. 1. Process of Simulation.

용된 것인지를 구별하고, 해당 채팅이 제재 대상인지를 결정한다. 시뮬레이션의 전체 프로세스는 Fig. 1과 같다. 처벌 데이터를 수집하여 필터링을 방해하는 무의미한 문자를 삭제한다. 그 후 단어의 빈도를 측정하고 군집화하여 같은 의미가 있는 변형 단어를 묶어준다.

3.2 데이터의 수집 및 가공

한 인터넷 방송에서 채팅 관리자가 처리하지 못한 금칙어 채팅을 시청자가 캡처하여 신고하는 게시판을 참고한다. 신고된 채팅 데이터 중 일상생활에서 흔히 사용되는 비속어가 포함된 채팅 데이터 및 채팅 데이터에 대한 처벌 분류를 수집한다. 처벌 분류 값은 0:해당 없음, 1:채팅 금지, 2:강제 퇴장으로 정의된다. Fig. 2는 수집된 채팅 데이터와 그에 해당하는 처벌 분류 값의 일부를 보여준다.

수집된 채팅 데이터 중에는 필터링을 회피하기 위한 의미 없는 문자들이 삽입되어 있거나 특수문자 등이 섞여 있어 데이터를 분석하는 데 방해가 될 수 있다. 그러므로 공백, 특수문자, 숫자, 영문자, 미완성 자음, 미완성 모음은 제거하고 완성형 한글 문자만 추출하여 가공한다. Fig. 3은 채팅 데이터에 대한 가공의 예를 보여준다.

3.3 채팅 데이터 쪼개기 및 빈도수 분석

인터넷 채팅은 Fig. 3을 통해 알 수 있듯이 문법이

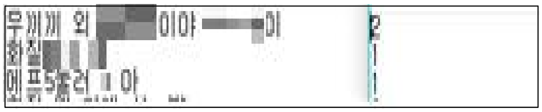

Fig. 2. Collected chat data and corresponding punishment classification.

Fig. 3. Examples of chat data processing.

잘 지켜지지 않고, 사전에 존재하지 않는 비속어, 신조어, 줄임말, 인터넷 용어 등이 자주 사용되는 특징을 갖고 있다. 필터링을 우회하기 위해 변형된 단어와 의미 없는 문자가 삽입되는 것도 확인할 수 있다. 이러한 특징으로 인해 어간 추출(stemming), 표제어 추출(lemmatization)과 같은 자연어 처리에 사용되는 기법을 사용하기 어렵다.

본 연구에서는 채팅 데이터들로부터 의미 있는 데이터를 추출하기 위해 n-그램(n-gram) 분석법[11]을 사용한다. 두 글자 단위로 쪼개기 위해 2-그램을 적용하며, 쪼개어진 단어에 대한 빈도수가 높은 상위 10개 정도의 단어들은 결과는 Fig. 4와 같다.

3.4 변형된 금칙어 군집화

변형된 금칙어들끼리 군집화하기 위해서는 일단 각 금칙어를 위치 개념으로 코드화해야 한다. 위치 개념으로 코드화해야 거리 개념으로 군집화를 할 수 있기 때문이다. 먼저 한 글자를 Fig. 6과 같이 초성, 중성, 종성에 해당하는 숫자로 치환하여 3차원 벡터로 코드화한다. 하지만 이와 같은 코드화는 ‘ㄱㄴㄷㄹㅁ...’, ‘ㅏㅐㅑㅒㅓㅔ...’ 순으로 비슷하지 않은 음을 갖는 자음, 모음끼리 가까운 거리에 있을 수 있게 되어 군집화가 제대로 수행되지 않을 가능성이 크다. 따라서 음이 비슷한 자음, 모음이 비슷한 위치에 있도록 코드화를 수행해야 한다. 아래 Table 1, 2, 3은 초성, 중성, 종성의 비슷한 발음을 가진 음운을 대표

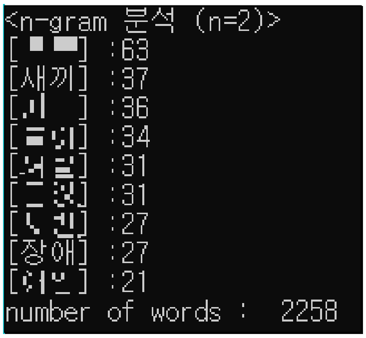

Fig. 4. The result of 2-gram analysis of the collected data.

Table 1. Rules of transforming for first syllable

First Syllable	
Existing syllable	Replaced syllable
ㄱ,ㄲ,ㅋ	ㄱ
ㄷ,ㄸ,ㅌ	ㄷ
ㅂ,ㅃ,ㅍ	ㅂ
ㅅ,ㅆ	ㅅ
ㅈ,ㅉ,ㅊ	ㅈ

Table 2. Rules of transforming for middle syllable

Middle Syllable	
Existing syllable	Replaced syllable
ㅏ,ㅑ	ㅏ
ㅐ,ㅒ,ㅔ,ㅖ,ㅙ,ㅚ,ㅞ,ㅟ	ㅐ
ㅓ,ㅕ	ㅓ
ㅗ,ㅛ	ㅗ
ㅜ,ㅠ	ㅜ
ㅢ,ㅡ,ㅣ	ㅣ

Table 3. Rules of transforming for last syllable

Last Syllable	
Existing syllable	Replaced syllable
ㄱ,ㄲ,ㅋ	ㄱ
ㄷ,ㅌ,ㅅ,ㅈ,ㅉ,ㅊ	ㄷ
ㅂ,ㅍ	ㅂ

할 하나의 음운으로 그룹화하는 기준이다.

Fig. 5와 같이 코드화를 수행하였는데, 이는 관련 논문[5]에서 제시한 비속어의 표준화를 위한 변형 규칙을 참고하여 정의하였고 그 내용은 아래 Table 4, 5와 같다. Table 4의 코드화 규칙은 Table 1, 2, 3의

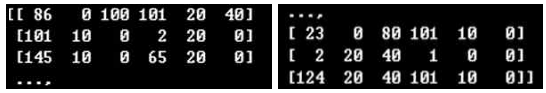

Fig. 5. Result of encoding.

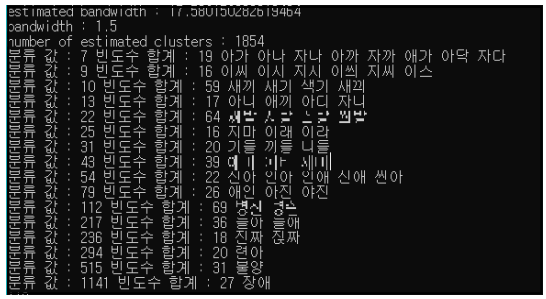

Fig. 6. Mean-shift clustering of coded results based on Table 4.

그룹화 규칙을 적용하지 않고 단순히 순서대로 음운의 고윳값을 부여한 방식이며, Table 5의 코드화 규칙은 Table 1, 2, 3의 그룹화 규칙을 적용하여 발음이 비슷한 음운에 비슷한 고윳값을 부여한 방식이다.

결과는 Fig. 5과 같으며, 두 글자의 초성, 중성, 종성을 각각 1차원 벡터씩 총 6차원 벡터로 코드화된 것을 볼 수 있다.

단순하게 한글 자모음 순으로 코드화된 6차원 벡터를 평균 이동 알고리즘(mean-shift algorithm)[12]으로 군집화한 결과는 Fig. 6과 같다. Fig. 7은 초성, 중성, 종성별로 코드화를 적용한 후 평균 이동 알고리즘으로 군집화를 수행한 결과이다. 둘을 비교했을 때, 음이 비슷한 자음, 모음이 비슷한 위치에 있도록 코드화한 결과가 실제로 군집화가 잘 수행된 것임을 확인할 수 있다.

Table 4. Encoding for consonants, vowels of a Korean syllable

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
First	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ	ㅅ	ㅆ	ㅇ	ㅈ	ㅉ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ								
Middle	ㅏ	ㅐ	ㅑ	ㅒ	ㅓ	ㅔ	ㅕ	ㅖ	ㅗ	ㅘ	ㅙ	ㅚ	ㅛ	ㅜ	ㅝ	ㅞ	ㅟ	ㅠ	ㅡ	ㅢ	ㅣ						
End	X	ㄱ	ㄲ	ㄳ	ㄴ	ㄵ	ㄶ	ㄷ	ㄹ	ㄺ	ㄻ	ㄼ	ㄽ	ㄾ	ㄿ	ㅀ	ㅁ	ㅂ	ㅄ	ㅅ	ㅆ	ㅇ	ㅈ	ㅊ	ㅋ	ㅌ	ㅍ

Table 5. Encoding consonants and vowel that similar sounds become nearby

First	ㅋ:0 ㄱ:1 ㄲ:2 ㄴ:22 ㄹ:23 ㅌ:43 ㄷ:44 ㄸ:45 ㅁ:65 ㅍ:85 ㅂ:86 ㅃ:87 ㅅ:101 ㅆ:102 ㅉ:122 ㅈ:123 ㅊ:124 ㅎ:144 ㅇ:145
Middle	ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅟ ㅝ:0 ㅐ ㅔ ㅒ ㅖ ㅙ ㅞ ㅚ ㅘ ㅡ ㅢ:10 ㅣ ㅜ ㅓ ㅟ:20
End	X:0 ㅋ ㄱ ㄲ ㄳ:20 ㄵ ㄶ ㄴ:40 ㄹ ㄺ ㄻ ㄼ ㄽ ㄾ ㄿ ㅀ:60 ㅁ ㅂ ㅄ ㅍ:80 ㅌ ㄷ ㅅ ㅆ ㅈ ㅊ ㅎ:100

Fig. 7. Mean-shift clustering of coded results based on Table 5.

3.5 채팅 데이터 처벌 분류

새로운 채팅 데이터가 입력되었을 때, 해당 채팅 데이터의 각 단어가 어떤 단어들과 비슷한지 분류하기 위해서 각 단어의 6차원 벡터값을 특징으로 정한다. 이전 단계에서 군집화한 것을 토대로 분류한 값을 목푯값으로 한다.

군집화를 토대로 분류한 목푯값을 채팅 데이터의 처벌을 분류할 때 사용할 특징으로 선택한다. 먼저 수집된 채팅 데이터들과 수집된 처벌 분류 값으로 분류기를 학습시킨다. 테스트 채팅 데이터에 해당 특징에 속하는 단어가 하나도 존재하지 않는다면 0(해당 없음)으로 분류하였고, 처벌 강도 순으로 1, 2, 3까지 분류한다.

4. 금칙어 필터링 알고리즘 고도화

4.1 필터링 알고리즘 선별

본 실험에서는 채팅 데이터에 대한 처벌 분류를 위하여 두 가지 분류 알고리즘을 적용하고 그 정확도를 비교한다. 하나는 최근접 이웃 알고리즘(k-nearest neighbor algorithm)[13]이고 다른 하나는 학습 기반의 인공신경망 알고리즘(artificial neural network algorithm)[14-15]이다. 최근접 이웃 알고리즘은 구현이 쉽고, 계산량이 적은 장점이 있어서 실험에 사용한다. 인공신경망 알고리즘은 최근 많이 활용되는 기계학습 방식이며 높은 정확도를 보이므로 채택하였다.

4.1.1 최근접 이웃 알고리즘을 이용한 단순 분류

15차원의 입력 데이터 간의 관련성을 찾기 효율적인 마할라노비스 거리(Mahalanobis distance) 기반의 최근접 이웃 알고리즘으로 군집화한다. 이에 관한

Fig. 8. Classification results for new words input.

결과로, Fig. 8과 같이 비속어가 변형되어 입력되어도 비슷한 비속어로 분류할 수 있다. 마할라노비스 거리란 평균과의 거리가 표준편차의 몇 배인지를 나타내는 값이다. 거리 계산을 위한 수식 (1)과 같다.

$$D_m(\vec{x}) = \sqrt{(\vec{x}-\vec{\mu})^T S^{-1}(\vec{x}-\vec{\mu})} \quad (1)$$

$\vec{\mu}$ = 평균 벡터
S = 공분산
$\vec{x}$ = 공분산을 갖는 벡터

Fig. 9는 탐색 노드 수에 따른 정확도의 차이를 보여주는데, 학습 데이터의 특성에 따라 최적값이 달라질 수 있음을 알 수 있다.

4.1.2 인공신경망을 이용한 채팅 데이터 처벌 분류

인공신경망은 기계학습과 인지과학에서 생물학의 신경망(동물의 중추신경계 중 특히 뇌)에서 영감을 얻은 통계학적 학습 알고리즘이다. 인공신경망은 시냅스의 결합으로 네트워크를 형성한 인공 뉴런(노드)이 학습을 통해 시냅스의 결합 세기를 변화시켜, 문제 해결 능력을 갖추는 모델 전반을 가리킨다[16]. 본 실험에서는 입력 데이터 계산에 사용되는 활성화 함수는 시그모이드 함수[17]를 활용한다. 시그모이드 함수는 아래 수식 (2)와 같다.

$$\text{Sigmoid function} = \frac{1}{1+e^{-x}} \quad (2)$$

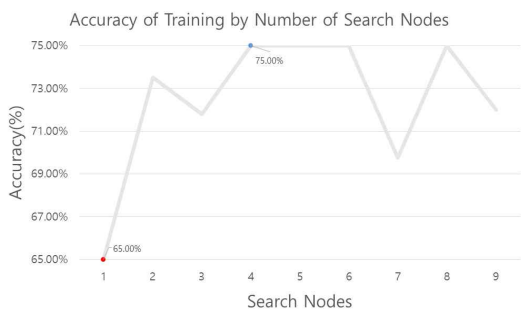

Fig. 9. Accuracy of training by number of search nodes.

Table 6. Optimal value search through variable controlling

Manipulation variable	Control variable
Node	Repeats, Learning Rate
Repeats	Node, Learning Rate
Learning Rate	Node, Repeats

각 입력 노드에는 앞서 Fig. 7의 사전 데이터 처리에서 추출된 15가지의 욕설 포함 여부를 입력한다. 각 출력 노드에는 욕설 정도에 따른 처벌 분류 값 3가지를 입력한다. 입력된 데이터를 반복 학습하는 과정에서 출력 계층의 오차 값을 은닉 계층의 가중치에 따라 나누어 학습률과 곱한 후 기존 가중치에 합산하여 갱신한다. 가중치 합산 계산식은 아래 수식 (3)과 같다.

$$\triangle w_{ij}(t+1) = \triangle w_{ij}(t) + \alpha \frac{\partial C}{\partial w_{ij}} \quad (3)$$

여기서 α는 학습률(learning rate)이며, C는 타겟값과 예측값의 차이 값으로 계산되는 비용함수이다.

4.2 최적 변숫값 탐색 실험

최근접 이웃 알고리즘이 아닌 인공신경망을 이용하여 위의 6차원 벡터값 및 목푯값을 학습하였다. 본 실험에서는 인공신경망의 적정 은닉 계층의 노드 수, 반복횟수, 학습률을 탐색하기 위하여 실험을 진행한다. 아래 Table 6은 각 변수의 최적값을 찾기 위해 결괏값에 영향을 미치는 다른 변수를 고정하여 최적의 변숫값을 찾는 기준이다.

모든 실험의 시작 조건을 입력 계층(input layer)의 노드는 15개, 은닉 계층(hidden layer) 1개, 출력 계층(output layer)의 노드는 3개로 설정한다. 은닉 계층을 2개 이상으로 설정한 경우 오히려 정확도가 감소하여 제외한다. 입력 계층의 노드에는 앞의 Fig. 7과 같이 빈도수가 높은 순으로 전체 데이터의 5% 이상이 포함한 군집만 추출하여 해당 단어의 포함 여부를 입력해주었고 15개의 군집이 추출한다. 아래

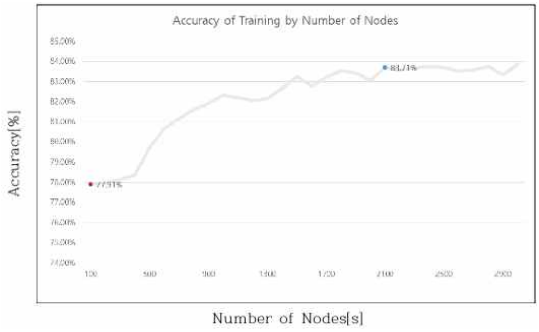

Fig. 10. Pre-test accuracy of training by number of nodes (from 100 to 3000).

Fig. 11. Pre-test accuracy of training by learning rate.

Table 7은 사전 실험에서 사용될 은닉 계층의 노드 수, 학습률, 반복횟수이다.

아래 Fig. 10은 노드 수별 정확도의 평균값을 정리한 그래프이다. 최고점은 노드가 3000개 일 때 83.36%이지만, 2100개 이후로는 큰 정확도의 차이가 없고 오히려 감소하는 예도 있다. 계산량이 많아질수록 효율이 감소하므로 뒤의 실험에서는 노드의 최대 수를 2100으로 제한한다.

아래 Fig. 11는 학습률별 정확도 평균값을 정리한 그래프이다. 최고점은 학습률이 0.3일 때 84.66%로 가장 높은 정확도를 보이지만 학습률이 0.4에서도 큰 차이를 보이지는 않기에, 뒤의 실험에서는 학습률의 최댓값을 0.4로 제한할 예정이다.

아래 Fig. 12은 반복횟수별 정확도 평균값을 정리

Table 7. Setting experimental conditions for the pre-test

Number of Nodes	100 200 300 400 500 600 700 800 900 1000 1100 1200 1300 1400 1500 1600 1700 1800 1900 2000 2100 2200 2300 2400 2500 2600 2700 2800 2900 3000
Learning Rate	0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Repeats	100 300 500 700 900

Fig. 12. Pre-test accuracy of training by number of repeats.

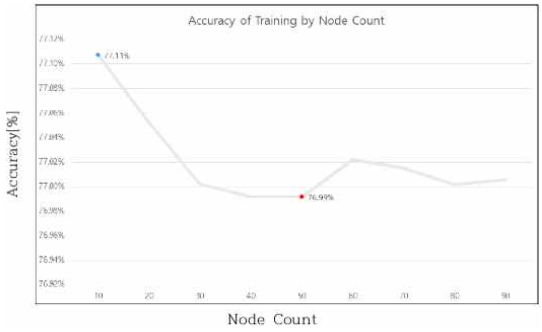

Fig. 13. Accuracy of training by node count (From 10 to 90).

한 표이다. 일반적으로 기계학습의 경우 학습 데이터가 많고 반복횟수가 많아질수록 성능이 좋아지는 특징을 갖고 있으나 반복횟수가 100일 때 오히려 가장 높은 정확도를 보인다. 본 실험에서는 정확도가 많이 감소하는 500회 이후의 반복횟수는 넣지 않고 (50, 100, 200, 300, 400) 총 5가지로 좀 더 세분화하여 실험한다.

4.3 계층 노드 수에 따른 정확도 탐색 실험

사전 실험을 통해 얻은 데이터를 기반으로 은닉계층의 노드 수, 학습률, 반복횟수를 정한다. 아래 Table 8은 본 실험에서 사용될 은닉 계층의 노드 수, 학습률, 반복횟수이다. 입력 계층의 노드 수가 15개이고, 출력 계층의 노드 수가 3개이므로 은닉 계층의 노드 수가 적은 경우도 추가한다.

첫 번째로 은닉 계층의 노드 수별 평균 정확도를 보았다. 학습률과 반복횟수는 위의 경우와 같이 각각 노드별로 200가지의 실험을 거친 결괏값의 평균이다. 입력 계층의 노드 수가 15개이고 출력 계층의 노드 수가 3개이므로 은닉 계층의 노드 수가 10~90개로 적은 경우부터 실험한다. 최고 정확도가 77.11%를 보이며 노드 수가 적은 실험의 경우 최근접 이웃 알고리즘과 정확도 차이가 거의 나지 않는 결과를 보인다. 아래 Fig. 13는 은닉 계층의 노드 수가 10~90개인 실험의 결과이다.

은닉 계층의 노드 수가 100개부터 2100개까지의 실험에서는 노드 수가 100개부터 증가할수록 정확도가 감소하다 700개인 지점을 지나는 순간부터 정확도의 상승을 보인다. 2100개일 때 가장 높은 정확도인 81.21%를 보이며, 700개일 때 가장 낮은 정확도인 76.42%를 보인다. 해당 결과는 아래 Fig. 14와 같다.

두 번째로 학습률 별 평균 정확도를 보았다. 은닉

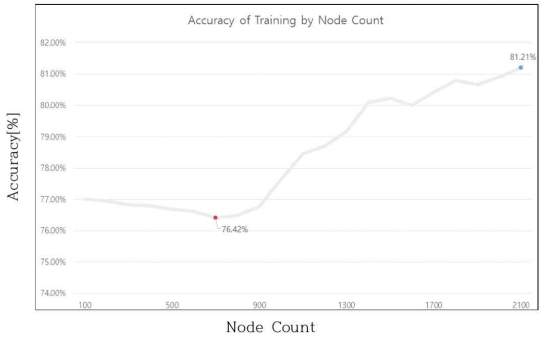

Fig. 14. Accuracy of training by node count (From 100 to 2100).

Table 8. Setting experimental conditions for the test

Lower Number of Nodes	10 20 30 40 50 60 70 80 90
Upper Number of Nodes	100 200 300 400 500 600 700 800 900 1000 1100 1200 1300 1400 1500 1600 1700 1800 1900 2000 2100
Learning Rate	0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 0.11 0.12 0.13 0.14 0.15 0.16 0.17 0.18 0.19 0.2 0.21 0.22 0.23 0.24 0.25 0.26 0.27 0.28 0.29 0.3 0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.4
Repeats	50 100 200 300 400

Table 9-1. Accuracy of Training by Learning Rate

Learning Rate	Accuracy
0.01	74.61%
0.02	74.73%
0.03	75.18%
0.04	75.55%
0.05	74.70%
0.06	74.48%
0.07	74.36%
0.08	74.72%
0.09	74.85%
0.1	75.00%
0.11	75.20%
0.12	75.76%
0.13	75.92%
0.14	76.19%
0.15	76.22%
0.16	77.01%
0.17	76.81%
0.18	77.21%
0.19	77.57%
0.2	78.05%

Table 9-2. Accuracy of Training by Learning Rate

Learning Rate	Accuracy
0.21	78.43%
0.22	78.90%
0.23	78.88%
0.24	79.31%
0.25	79.44%
0.26	79.73%
0.27	79.94%
0.28	80.44%
0.29	80.72%
0.3	80.66%
0.31	80.88%
0.32	80.91%
0.33	80.92%
0.34	81.09%
0.35	81.44%
0.36	81.25%
0.37	81.31%
0.38	81.51%
0.39	81.36%
0.4	81.65%

계층의 노드 수와 반복횟수는 위의 경우와 같이 각각 30개, 5개로 학습률 별 150가지의 실험을 거쳤다. 아래 Table 9-1과 9-2는 평균 정확도를 정리한 표이다. 학습률은 0.4일 때 가장 높은 정확도 평균을 보이며, 0.07인 경우가 제일 낮은 정확도 평균을 보인다.

아래 Fig. 15은 반복횟수 300회를 기준으로 사전 테스트의 가장 높은 정확도를 보인 학습률 0.3과 본 테스트의 가장 높은 평균 정확도를 보인 학습률 0.4

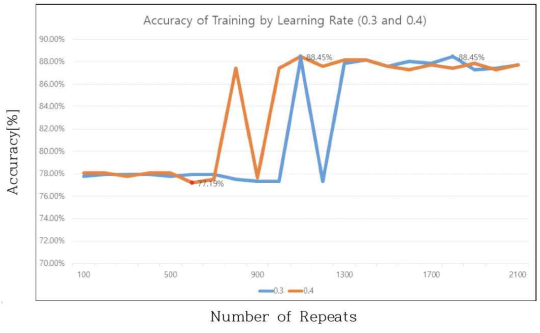

Fig. 15. Accuracy of Training by Learning Rate (0.3 and 0.4).

의 노드 수별 정확도를 정리한 그래프이다. 평균적으로는 학습률이 0.4인 경우 높은 정확도를 보이며, 두 경우 모두 88.45%의 최고 정확도를 보인다. 같은 정확도를 보이는 두 실험에서 학습률이 0.4일 때는 노드 수가 1100개이고, 학습률이 0.3일 때는 노드 수가 1800개이다. 계산시간 및 자원 사용의 효율성 측면에서 학습률이 0.4인 경우 노드 수가 적어 더 효율적이라 볼 수 있다.

좀 더 정확한 분석을 위하여 0.3부터 0.4까지의 학습률을 다시 0.1단위로 나누어 실험한다. 아래 Fig. 16은 10가지 학습률을 위 Table 8의 조건에 맞추어 실험한 결과, 가장 높은 정확도가 나온 실험의 그래프이다. 노드 수는 1300개이고 반복횟수가 300회인 경우 학습률 0.33에서 가장 높은 정확도 88.60%를 보인다.

마지막으로 인공신경망 학습 반복횟수별 각각의 가장 높은 정확도를 보았다. 은닉 계층의 노드 수와 학습률은 위의 경우와 같이 각각 30개, 40개로 반복횟수별 1200가지의 실험을 진행한다. 반복횟수는 앞

Fig. 16. Accuracy of Training by Learning Rate (from 0.3 to 0.4).

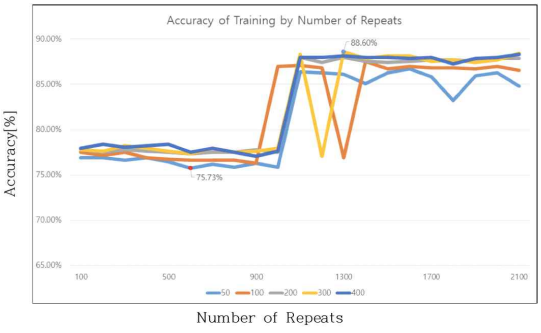

Fig. 17. Accuracy of Training by Number of Repeats.

서 실험한 은닉 계층의 노드 수와 학습률과 다르게 반복을 거듭할수록 평균 정확도가 올라가지만, 오히려 너무 과한 횟수로 반복할 경우 정확도가 떨어지는 결과를 보인다. 이는 인공신경망이 많은 데이터를 여러 번 학습 할수록 성능이 향상되지만, 너무 많은 반복을 하면 과적합(overfitting)이 될 수 있는 특성을 보여준다. 아래 Fig. 17은 본 테스트에서 가장 높은 정확도를 보인 학습률 0.33을 기준으로 반복횟수 및 노드 수별 정확도를 정리한 그래프이다. 반복횟수 300회일 때 가장 높은 정확도 88.60%를 보이며, 50회만 학습하였을 때는 가장 낮은 정확도 75.73%를 보인다.

4.4 활성화 함수와 오차 함수 실험

신경망의 완성도를 높이기 위하여 인공신경망의 다양한 활성화 함수와 오차 함수를 적용하여 실험을 진행하였다.

활성화 함수에 성능이 좋아 많이 사용되는 함수인 ReLU(rectified linear unit) 함수를 시그모이드 함수 대신 적용해보았다. 이 경우 노드 수가 900개일 때 가장 높은 평균 정확도를 보이고, 학습률이 0.3과 0.4에서 높은 정확도를 보인다. 좀 더 자세한 값을 탐색하기 위해 학습률을 0.3에서 0.4까지 0.1단위로 새로 실험을 진행한다. 그 결과 학습률이 0.35일 때 80.41%로 가장 높은 정확도를 보이며, 이는 본 실험에서 사용한 시그모이드 함수를 사용한 알고리즘보다 낮은 최고 정확도를 보인다.

오차 함수로는 단순 오차, 단순 오차의 절댓값, 단순 오차의 제곱을 적용하여 정확도를 계산해보았다. 오차값은 학습률에 직접적인 영향을 받으므로 학습률을 0.1~1.0까지 0.1 단위로 설정한다. 단순 오차를 사용한 신경망은 정확도가 88.60%로 가장 높게 나온다. 단순 오차의 절댓값을 적용했을 때는 학습률 0.6에서 정확도가 73.98%이고 단순 오차의 제곱을 적용했을 때는 학습률 0.7에서 75.44%의 정확도를 보인다.

본 실험에서는 시그모이드 함수를 활성화 함수로 사용하고 단순 오차 함수를 적용하였을 때 가장 높은 정확도를 보임을 알 수 있다. 본 실험의 알고리즘 특성상 많은 노드 수를 사용하여 정확하게 분류하는 ReLU 함수는 많은 가중치를 0 또는 무한대로 만들어 버려 시그모이드 함수를 사용한 알고리즘보다 정확도가 낮은 것으로 판단된다. 오차 함수의 경우는 다른 방법들은 양의 오차와 음의 오차를 고려하지 않아서 역전파 계산에 정확한 값을 전달해 줄 수 없으며, 이는 분류 항목별 오차가 서로 영향을 미치지 않기 때문으로 판단된다.

4.5 실험 결과 및 고찰

최근접 이웃 알고리즘과 인공신경망을 이용한 네 가지 실험을 진행하였다. 최근접 이웃 알고리즘을 이용한 실험에서는 75%의 정확도를 보인다. 인공신경망을 이용한 세 가지 실험을 거쳐 최적의 은닉 계층과 노드의 수, 반복횟수 및 학습률의 값을 찾아내었다. 이를 통해 앞선 최근접 이웃 알고리즘을 이용한 단순 필터링보다 신경망을 이용한 필터링의 정확도가 개선되었다. 신경망의 구성을 보았을 때, 최적의 은닉 계층의 노드 수는 입력 계층의 노드 수와 관련이 있음을 알 수 있다. 모든 실험 중 가장 좋은 성능을 보인 경우는 88.60%의 정확도를 보인 경우이다. 위 세 가지 평균 정확도를 통한 최적의 노드 수는 2100개, 학습률은 0.33, 반복횟수는 300회이지만 가장 높은 정확도인 88.60%를 보인 실험은 사례 1 (노드 수

: 1300, 학습률 :0.33, 반복횟수 : 300), 사례 2 (노드 수 : 1100, 학습률 : 0.32, 반복횟수 : 400) 총 두 가지의 경우이다. 이를 통해 데이터별로 평균적인 정확도를 통해 노드 수, 학습률, 반복횟수의 최적값을 도출해 낼 수 있다, 하지만 노드 수의 최적값은 학습률에 따라 달라지기에 가장 높은 정확도를 보인 실험과 세 가지 조건의 최적값은 다를 수 있다. 또한, 신경망 훈련 효율을 증대하기 위해선 너무 높거나 너무 낮지 않은 적절한 값의 학습률[18]을 찾아야 한다. 본 실험의 결과 같은 최고 정확도를 보인 두 가지 경우 중 노드 수와 반복횟수에 비례하는 학습용량(learning capacity)이 더 적은 사례 1이 더 효율적이다.

금칙어 자동 필터링을 위한 알고리즘에는 최근접 이웃 알고리즘을 이용한 실험 결과인 75%보다 신경망을 사용하는 방법이 13% 높은 88.60%의 정확도를 보이므로 인공신경망 기술이 금칙어 필터링에 더 효과적임을 확인할 수 있다.

신경망을 구성하는 은닉 계층의 노드 수에 따른 정확도는 일정 지점을 지난 후 노드 수가 증가할수록 커진다. 이는 입력 데이터를 판별하는 인자가 많아질수록 더욱 정확하게 목푯값을 지정할 수 있음을 보여준다.

채팅 데이터 필터링을 위한 해당 알고리즘의 학습률은 0.33일 때 가장 높은 정확도 평균을 보이며, 최고 정확도를 보인 두 가지 실험 결과 중 총 학습용량을 고려하여 학습률 0.33인 경우가 가장 높은 효율성을 보인다.

알고리즘 학습 반복횟수는 반복을 거듭할수록 평균 정확도가 올라가는 결과를 보인다. 이는 인공신경망이 많은 데이터를 여러 번 학습 할수록 성능이 향상함을 보여준다. 그러나 300회 이상 반복하면 오히려 정확도가 떨어졌으며 과적합 상황이 발생한 것으로 판단된다.

본 연구에서는 군집화한 분류 중 빈도수가 높은 군집을 기반으로 채팅 데이터의 처벌을 분류한다. 이 방법은 단순히 빈도수만으로 군집하였기 때문에 금칙어가 아닌 사용자 이름이나 호칭과 같은 단어가 특징에 포함될 수 있는 문제가 있다.

5. 결 론

본 연구에서는 변형된 금칙어를 자동검출해낼 수 있는 금칙어 및 변형 금칙어 자동 필터링을 이용한 실시간 자동 채팅 제재 수위 판별 시스템을 소개한다. 기계학습을 통한 자동 채팅 제재 판별을 위해 n-그램 방식을 채팅 데이터의 사전 처리에 적용하였다. 또한, 사전 처리가 완료된 데이터를 통해 기존의 처벌 방식을 최근접 이웃 알고리즘 및 인공신경망 두 가지 방식으로 학습하여 새로운 데이터의 자동 처벌 시스템의 신뢰도를 상승시켰다.

본 연구의 실험에서는 최근접 이웃 알고리즘을 이용한 실험 결과인 75%보다 신경망을 사용하는 방법이 13% 높은 88.60%의 정확도를 보였다. 학습률은 0.33일 때 가장 높은 정확도 평균을 보였으며, 0.07인 경우가 제일 낮은 정확도 평균을 보였다. 최고 정확도를 보인 두 가지 실험 결과의 학습률 (0.33, 0.32) 중 총 학습용량이 더 낮은 실험의 학습률 0.33이 높은 효율성을 보였다.

본 연구에서 제안하는 금칙어 및 변형 금칙어 필터링 기술을 인터넷 채팅, 댓글이나 온라인 게임 등에 적용하면 기존의 필터링 방식보다 더 효과적인 필터링이 가능해질 것으로 예상된다. 더 나아가서 자연어 처리 및 채팅 분석 시스템 등의 개발에도 이바지할 수 있을 것이다.

REFERENCE

[1] Korea Internet & Security Agency, *2011 Internet Ethical Culture Survey Results*, 2011.

[2] Korea Game Industry Agency, *Study of Game Language Restoration Guidelines,* 2008.

[3] T.J. Yoon and H.G. Cho, "A Filtering System for On-line Vulgar Words Using Korean Syllable Alignment," *Journal of the Korean Institute of Communication Sciences*, Vol. 36, No. 2C, pp. 194-198, 2009.

[4] M.A. Hearst and S.T. Dumais, "Support Vector Machines." *IEEE Intelligent Systems and their Applications*, Vol. 13, No. 4, pp. 18-28, 1998.

[5] K.H. Park and J.H. Lee, "Developing a Vulgarity Filtering System for Online Games Using SVM," *Journal of the Korean Institute of Communication Sciences*, Vol. 33, No. 2B, pp. 260-263, 2006.

[6] T.C. Jo, "The Comparison of Neural Network

and k - NN Algorithm for News Article Classification," *Journal of the Korean Institute of Communication Sciences*, Vol. 25, No. 2Ⅱ, pp. 363-365, 1998.

[7] S.I. Seo and S.B. Cho, "A Transfer Learning Method for Solving Imbalance Data of Abusive Sentence Classification," *Journal of Korean Institute of Information Scientists and Engineers*, Vol. 44, No. 12, pp. 1275-1281, 2017.

[8] S.B. Ou and J.W. Lee, "Implementation of a Spam Message Filtering System Using Sentence Similarity Measurements," *Korean Institute of Information Scientists and Engineers Transactions on Computing Practices*, Vol. 23, No. 1, pp. 57-64, 2017.

[9] S.B. Lee, Y.H. Son, H.C. Jang, and K.C. Lee, "The Development of the Korean Medicine Symptom Diagnosis System Using Morphological Analysis to Refine Difficult Medical Terminology," *Korean Institute of Information Scientists and Engineers Transactions on Computing Practices*, Vol. 22, No. 2, pp. 77-82, 2016.

[10] W.R. Park and T.K. Park, "Design and Implementation of Channel Filtering System based on TV Watching Patterns," *Journal of Korea Multimedia Society*, Vol. 13, No. 10, pp. 1413-1422, 2010.

[10] W.B. Cavnar and J.M. Trenkle, "N- gram-based Text Categorization," *Ann Arbor mi*, Vol. 48113, No. 2, pp. 161-175, 1994.

[11] Y.Z. Cheng, "Mean Shift, Mode Seeking, and Clustering." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 8, pp. 790-799, 1995.

[12] Anirudh Harisinghaney, Aman Dixit, Saurabh Gupta, and Anuja Arora, "Text and Image Based Spam Email Classification Using KNN, Naïve Bayes and Reverse DBSCAN Algorithm," *Proceeding of 2014 International Conference on Reli-ability Optimization and Information Technology*, pp. 153-155, 2014.

[13] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Neural Networks*, Vol. 61, pp. 85-117, 2015.

[14] Ajith Abraham, Artificial Neural Networks, *Handbook of Measuring System Design*, (John Wiley & Sons, Ltd, Hoboken, NJ, 2005)

[15] Artificial neural network - Wikipedia(2019). https://en.wikipedia.org/wiki/Artificial_neural_network (accessed Feb., 21, 2019).

[16] J. Han and C. Moraga, "The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning," *International Workshop on Artificial Neural Networks*, pp. 195-201, 1995.

[17] Setting the Learning Rate of Your Neural Network, https://www.jeremyjordan.me/nn-learning-rate/ (accessed Nov., 7, 2018).

김 찬 우

2013년~2017년 인천대학교 컴퓨터공학 학사
2017년~2019년 인천대학교 일반대학원 컴퓨터공학 석사
2019년~현재 제이엘케이 인스펙션 연구원
관심분야 : 인공지능, 게임 AI

성 미 영

1982년 서울대학교 학사
1987년 프랑스 INSA de Lyon 컴퓨터공학 석사
1990년 프랑스 INSA de Lyon 컴퓨터공학 박사
1990년~1993년 한국전자통신연구소 선임연구원
1993년~현재 인천대학교 컴퓨터공학부 교수
관심분야 : 멀티미디어, 가상현실, 햅틱스, 인공지능