비속어 표준형 변환 알고리즘을 이용한 인터넷 비속어 검출 시스템

Internet Vulgarity Detecting System by Using Algorithm to Transform Modified Vulgar words to Basic Type

윤태진
Yoon Taijin
부산대학교 컴퓨터공학과
ytj@pusan.ac.kr

ABSTRACT

한글 비속어의 경우 그 종류가 많고 변형 방법이 다양하여 일반적인 기법으로는 검출이 어렵다. 방대한 비속어와 그에 파생되는 변형어를 모두 데이터 베이스화 하는데는 큰 어려움이 있으며 지나치게 많은 데이터를 일반적인 string matching을 수행할 경우 많은 부하를 발생시켜서 클라이언트 프로그램이나 서버에 지연 현상을 초래할 수 있다. 게다가 지나치게 비속어를 막는데만 중점을 둔 나머지 정상적인 단어를 입력해도 비속어로 판정해서 사용자간의 원활한 의사소통을 방해하는 문제를 발생시키기도 한다. 이러한 문제점을 해결하기 위해서는 두가지 조건이 필요하다. 먼저 변형에 강한 검출 기법이다. 비속어는 다양한 변형을 통하여 기존의 필터링 시스템을 피해 사용되고 있으나 그 변형 방식에는 일정한 규칙이 존재한다. 이러한 규칙성을 이용하여 입력된 단어의 표준화를 통해 변형에 강한 검출이 가능하다. 둘째로 효과적인 자료 구조이다. 계층적인 자료 구조를 이용하여 표준 비속어, 파생비속어를 저장하여 검색 횟수를 줄이는 방법을 제안한다.

KEYWORDS vulgar, spam filtering, language processing

1 서 론

인터넷의 발달로 채팅과 인터넷 게시판의 유저는 남녀노소 계층을 불문하고 폭발적으로 증가하였다. 다양한 계층과 사고방식을 가진 사람들이 서로 대화를 나누게 되는 인터넷의 특성상 분쟁이 자주 발생하게 되고 그 과정에서 여과없이 사용되는 비속어는 사용자들에게 큰 불쾌감을 주고 미성년자인 사용자들에게 악영향을 주고 있다. 그래서 온라인 게임과 게시판 등에서는 비속어 필터링 기능을 제공하고 있는데 여기에는 큰 두가지 문제점이 있다.

첫째, 욕의 변형형에 대해서 올바르게 작동하지 못한다. 다양한 발음을 가지고 있는 한글의 특성상 단어에 대해서 비슷한 발음을 가지는 다른 형태로 다양한 변형이 가능하다. "바보"라는 단어를 필터링한다고 하였을때 비속어를 사용하고자 하는 사용자는 이 단어를 사용해서 필터링 기능에 제지당한다고 해도 "바버", "babo" 등 변형된 욕을 사용하여 필터링 기능을 무력화 시킨다. 특히 외계어라고 불리는 인터넷 특유의 한글 파괴현상과 맛물려 형태가 비슷한 특수문자, 한자, 외국어 등을 조합해서

비속어를 표현하면 단순한 필터링 기능으로는 제재가 불가능하다. 이러한 비속어는 사용하기 불편하다는 단점이 있으나 최근에는 필터링되지 않는 변형된 형태의 욕을 매크로에 저장해서 온라인 분쟁시 사용하는 경우도 있다.

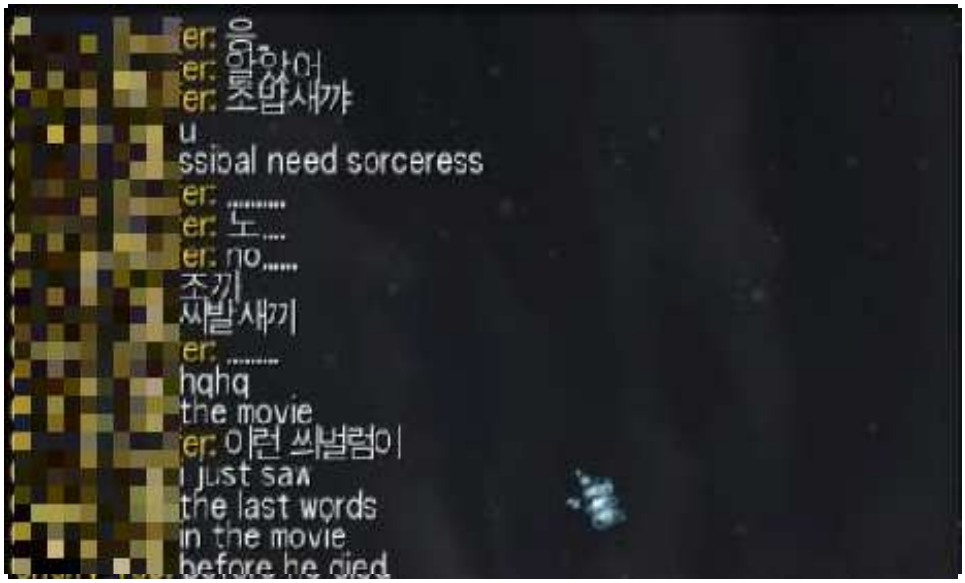

그림 1. 모 인기 게임의 채팅 장면. 인터넷에서 사용되는 비속어 문제는 심각하다. 그러나 비속어는 변형되어 사용되는 경우가 많아 필터링이 쉽지 않다.

둘째, 오히려 정상적인 단어를 필터링하는 경우가 있다. 모 온라인게임의 필터링 시스템의 경우 "상자위에 적이 있다."라는 문장을 입력할 경우 "자위"라는 단어를 필터링하여 정상적인 대화를 어렵게 한다. 그래서 원활한 대화를 위해서 변형된 단어나 은어를 사용해야 하는 모순이 발생하게 된다. 몇몇 유저들은 비속어를 제대로 검출하지 못하면서 대화에 불편함만 준다하여 이러한 비속어 필터링 시스템을 반대하기까지 한다. 올바른 국어사용을 권장하기 위해 만들어진 시스템이 오히려 한글파괴를 부추기게 되는 것이다.

기존 시스템이 가지고 있는 문제점을 해결하기 위해서 가장 중요한 것은 변형에 강한 검색 기법이다. 이것을 위한 가장 확실한 방법은 가능한 모든 변형을 포함하고 있는 비속어 사전을 만들어서 비속어 검출에 활용하는 것이다. 그러나 다양한 발음을 가지고 있는 한글의 특성상 매우 다양한 변형 방법이 존재하고 최근 특수문자와 알파벳까지 동원하는 변형 방식이 나오고 있어 현실적으로 어렵고 사전의 데이터가 지나치게 방대해지면 사전을 유지하기 위한 요구 메모리양과 검색에 필요한 연산량이 늘어나서 시스템에 큰 부하를 걸게 된다. 수많은 사용자가 접속하는 인터넷 서버의 특성상 비대한 비속어 처리 시스템을 사용할 경우 심각한 지연 현상을 초래하여 사용자에게 큰 불편을 가져 올 수

있다. 그러므로 이 문서에서는 표준형태와 최소한의 파생어만을 입력하고 변형형태의 욕에서 변형에 사용되는 규칙을 연구하여 규칙을 이용하여 표준형태로 바꿔서 대조해보는 방법을 제안한다. 그리고 비속어 필터링에 필요한 데이터를 효과적으로 검색할 수 있는 데이터 구조에 대하여 서술한다.

2 표준화를 통한 매칭 기법

일반적인 비속어 필터링 시스템에서 가장 취약한 부분은 변형된 비속어를 입력하는 경우라 할 수 있다. 특히 한글의 경우 조합 비속어는 수많은 변형 형태를 지닐 수 있고 그러한 변형된 형태로 사용될 경우 단순한 string matching으로는 검출해내기 어렵다. 가능한 변형 형태를 모두 데이터 베이스화 하는 방법이 있을 수 있겠으나 어디까지나 부가적인 모듈이 되는 비속어 필터링 기능이 지나치게 시스템이 큰 부하를 가하게 되면 시스템의 성능을 저하시켜 사용자에게 불편을 끼치게 된다. 그러나 다양한 변형형태를 지니고 있다고는 하나 변형된 형태를 살펴보면 어느정도 일관된 규칙을 찾을 수 있다.

표 1. 발음을 이용한 변형 예

기본 단어	변형단어
개새끼	개새기, 개새뀌, 개색기, 개색휘, 개색히, 개세이, 개쉐리, 개쉐이, 개색
개놈	개뇸, 개눔, 개늠, 개넘, 개너므, 게 넘, 게넘, 게놈, 게뇸, 게눔, 게늠, 궤놈
병신	뱅쉰, 뱅싄, 벵신, 벵틴, 병시인, 붱신, 붱 싄, 븅신, 븅쉰, 비융신
씨팔	쉬발, 쉬밸, 쉬뱅, 쉬벌, 쉬벧, 쉬볼, 쉬불, 쉬빌, 쉬파, 쉬팍, 쉬팔, 쉬팕, 쉬팡
불알	붕알, 부랄, 부럴, 브랄, 브라알, 블알, 뽕알, 뿡알

가장 일반적으로 사용되는 변형형태는 비슷한 발음 형태로 변화시키는 것이다. 예를 들어 욕으로 많이 사용되는 "개"라는 단어의 경우 "걔", "궤", "캐", "컈" 등의 발음으로 변형되어 사용되는 경우가 많다. 자음의 경우 된소리, 쎈소리 등으로 변화시키는 경우가 많고 모음의 경우 유사한 발음군으로 변화시켜 사용하게 된다. 이렇게 변화로 주로 사용되는 자소들을 통합하여 대표되는 발음으로 표준화시켜서 검색한다면 발음을 이용한 변형 사용을 막을 수 있다.

표 2은 본 시스템에서 사용되는 표준화 규칙을 나타낸 표이다. 모음의 경우 된소리, 쎈소리를 기본 발음으로 통합하였으며 모음의 경우 비슷한 발음이라 생각되는 것들을 하나로 통합하였다. 이 표준화만으로도 인터넷에서 사용되는 비속어 변형형의 상당수를 필터링 가능하게 되는 것을 알 수 있다. 본 표는 본인의 지식 및 경험을 통해 만들어진 표이므로 추가적인 연구를 통하여 통계학적인 근거를 마련하여 더욱 정확한 규칙을 만들 필요가 있을 것이다.

표 2. 비속어의 표준화를 위한 변형 규칙

초성		중성		종성	
원래 자소	바뀐 자소	원래 자소	바뀐 자소	원래 자소	바뀐 자소
ㄱ,ㄲ,ㅋ	ㄱ	ㅑ, ㅒ	ㅑ	ㄱ,ㄲ,ㅋ	ㄱ
ㄷ,ㄸ,ㅌ	ㄷ	ㅐ, ㅒ, ㅖ, ㅖ,ㅙ,ㅚ,ㅞ,ㅞ	ㅐ	ㄷ,ㅌ,ㅅ,ㅈ,ㅉ,ㅊ	ㄷ
ㅂ,ㅃ,ㅍ	ㅂ	ㅕ, ㅖ	ㅕ	ㅂ,ㅍ	ㅂ
ㅅ,ㅆ	ㅅ	ㅗ,ㅛ	ㅗ		
ㅈ,ㅉ,ㅊ	ㅈ	ㅜ,ㅠ	ㅜ		
		ㅟ,ㅢ, ㅣ	ㅣ		

다른 변형 방법으로는 비속어의 글자사이에 무의미한 빈칸이나 기호 등을 포함시켜 필터링을 피하는 방법이 있다. "멍청이"를 "멍 청 이"나 "멍,청,이"로 쓰게 된다면 사람은 같은 단어라는 것을 판단할 수 있으나 컴퓨터의 경우 다른 단어로 받아들여 필터링을 피해갈 수 있는 것이다. 이것을 해결하기 위하여 global alignment가 아닌 local alignment를 이용하여 단어간의 유사도를 판단하여 비속어를 판정하는 방법을 사용한다. 예를 들어 "멍청이"와 "멍-청-이"의 경우 matching score를 1.0으로 잡고 insertion gap score를 -0.31로 잡는다면 "멍", "청", "이" 세 글자가 match되고 두개의 ","이 insertion gap을 만들어서 2.38의 점수를 얻게 된다. 그러므로 global alignment가 이루어졌을 때 얻을 수 있는 점수가 3.0이므로 79.3%의 유사도를 얻을 수 있다.

더 정확한 Local Alignment의 측정을 위해서 자음, 모음을 분리한 형태로 만든다. 예를 들어 "개새끼"를 "갯샛낏"으로 변형시켜 입력할 경우 Local Alignment가 0으로 측정되게 된다. 그러나 "ㄱ ㅐㅅ ㅐㄲ ㅣ"와 "ㄱ ㅐㅅㅅ ㅐㅅㄲ ㅣㅅ"의 경우 ㅅ이 단순히 insertion gap을 만들어낼 뿐이므로 충분한 유사도가 측정될 수 있다. 비속어의 표준화 작업에도 이러한 자모 분리를 사용하는 방법이 유리하다.

표 3. 외래어와 특수문자를 이용한 변형 예

기본 단어	변형단어	기본 단어	변형단어
씨팔	cval, ㅆㅣ팔, ㅆ!발, ㅆ1바, ㅆㅣ발	개	ㄱH
개새끼	dog새끼, dog새	니기미	ㄴ1ㄱ1, nigimi
니미	ㄴ1ㅁ1, ㄴ1미	니에미	ㄴ1ㅇH미, ㄴ1에미, 니OH미
뒈진다	D질래	망할년	亡할년, 望할년
게이	gay, g@y	미친	ㅁ1친, ㅁㅣ친, ㅁㅣ친

현재 추가적으로 구현될 부분으로 유사한 형태나 발음의 알파벳이나 특수문자를 검출하기 위한

matching matrix를 완성하는 것이다. 이러한 변형의 경우 현재의 표준화 규칙으로는 올바른 표준형 추출이 어렵다. 예를 들어 "개새끼"를 "ㄱHㅅHㄲㅣ"로 입력하는 등의 방법이 있을 수 있고 이러한 변형 규칙에 대한 자료 수집 및 확립이 필요하다. 특히 이러한 변형은 시대의 흐름에 따라 발전속도가 빠르기 때문에 지속적인 보수방법에 대한 연구도 필요할 것이다.

3 효과적인 검색을 위한 계층적인 자료 구조

한글에는 다양한 비속어가 있다. 특히 인터넷의 발달로 인해 정보의 전파 속도가 기하급수적으로 빨라지면서 비속어의 발달 또한 가속화 되고 있다. 이렇듯 방대하고 빠른 속도로 증가하는 비속어를 효과적으로 저장하고 신속하게 검색하기 위해서는 계층적인 Tree 형태의 자료구조 형성이 필요하다. 데이터량의 증가에 대해 가장 유연하게 대처할 수 있는 구조이기 때문이다.

최상위 계층은 가장 중요한 영역이라고 할 수 있다. 이곳에 저장되는 단어의 숫자에 따라서 시스템의 검색성능이 크게 좌우되기 때문이다. 최상위 계층에 지나치게 많은 단어가 들어갈 경우 대부분의 단어를 비교해야하기 때문에 비속어 필터링 작업량이 크게 증가하게 된다. 이 단계에 들어가야할 비속어의 특징은 다음과 같이 정리된다.

1. 비속어의 표준형이라 할 수 있어야한다. 인터넷에서는 수많은 비속어가 사용되지만 가장 기본형이라 할 수 있는 표준형태들이 존재하기 마련이다. 최상위 계층에 올 수 있는 단어에 이러한 제한을 둠으로써 효율적인 자료구조의 형성이 가능하다.
2. 표준형이 아니더라도 alignment에서 크게 벗어난 단어를 입력한다. 예를 들어 "새끼"의 경우 표준형으로 변화시킬 수 있는 "쉐키" 같은 형태로도 쓰이지만 "스끼", "시키" 등 표준화 규칙에서 벗어난 형태의 변형도 사용된다. 지나치게 표준화 규칙을 확대할 경우 오히려 잘못된 검색이 수행될 가능성이 높으므로 이러한 형태의 비속어는 따로 등록하여 준다.

2단계에는 표준형의 파생형태를 입력한다. 앞에서 언급되었던 발음, 형태 등을 이용한 파생 형태를 입력한다. 표준화를 통한 검색도 완전하다고 할 수 없으므로 자주 사용되는 변형형태를 입력하여 필터링의 신뢰도를 높이기 위해서이다. 파생 형태를 입력할 때 중요한 점은 유사한 형태의 파생형이라 하더라도 표준형의 검색에서 matching이 이루어지는 단어만을 입력하는 것이다. 상위단계에서 matching이 이루어지지 않는다면 하위단계에서 검색될 수가 없기 때문이다.

추가적으로 구현되어야 할 부분으로 예외 단어의 입력이 있다. 단어사전을 통해 정상적인 단어를 걸러낸다고 하더라도 사용자끼리 자주 사용하는 은어나 외래어가 비속어 검색에 검출될 가능성이 있다. 특히 게임의 경우 이러한 경우가 자주 발생하는데 예를 들어 게임의 직업명으로 자주 사용되는 "성기사"의 경우 "성기"부분이 필터링에 걸려서 어려움을 겪는 경우가 많은데 "성기사"의 경우 국어사전에 등록되어 있지 않은 단어 이다. 그러므로 이 경우 사용자로부터 피드백을 받아서 예외단어 목록에 등록한 뒤에 비속어 필터링을 통해 검출되더라 하더라도 예외 단어 목록에 포함될 경우 필터링에서 제외하여야 한다.

4 시스템 구현 상황

비속어 필터링 시스템은 현재 C++를 통하여 구현되고 있으며 모듈화를 통해서 타 시스템의 채팅 및 게시판 등에 합쳐질 수 있도록 개발 중이다. 이 시스템은 크게 3개의 모듈로 구성된다.

첫째, Local Alignment를 측정하는 모듈이다. 현재 문자사이의 빈칸 및 무의미한 특수문자 삽입 등에는 효과적으로 대처하고 있으며 비교적 사용되는 string의 길이가 짧은 만큼 match, insertion gap, deletion gap 등의 스코어에 대해서 실험을 통해 효율적인 값의 추출을 진행하고 있다.

둘째, 비속어의 자소 분리 및 표준화 모듈이다. 한글 자소를 효과적으로 분리하기 위하여 유니코드를 이용해 글자의 수치를 받아와서 문자를 분리시킨다. 한글 완성형의 경우 문자에 따라 할당하는 범위가 틀려서 기계적인 자소 분리가 어려우나 유니코드의 경우 초성 19개, 중성 21개, 종성 28개가 모두 구현되어 있으므로 간단한 나누기와 나머지 연산을 통해서 자소의 분리가 가능하다. 자소 분리 함수의 경우 2가지가 존재하는데 한가지는 단순히 자소 분리만을 수행하는 함수와 표준형으로 변화시켜 자소분리를 수행하는 함수이다. 표준형 자소분리는 표제어와 Local Alignment를 측정 하는데 사용되고 일반 자소분리는 변형형으로 측정하는데 사용된다.

셋째, 비속어의 자료구조 처리 모듈이다. 비속어는 현재 표제어, 변형형 2단계로 저장되며 두개를 하나의 구조체로 묶어서 검색효율을 증가시켰다. 유사 matching처리를 위한 자료 구조는 현재 작업 중이며 이것 또한 계층적인 구조를 형성하여 작업효율을 높일 계획이다.

5 결론 및 향후 연구과제

본 보고서에서는 효과적인 비속어 필터링을 위한 검출 기법 및 자료 구조와 그 구현 상황에 대하여 서술하였다. 자모 분리를 이용한 비속어의 표준화 및 Local Alignment 측정을 통해서 한글 특유의 변형에 강한 비속어 검출 시스템을 구현하는데 성공하였으며 Tree구조를 이용한 효율적인 자료구조를 형성하여 신속한 검색이 가능하게 하였다. 그러나 해결해야 하는 문제점이 몇가지 존재한다.

1. 복합명사의 문제이다. 비속어 필터링을 하기 위해서는 단어 단위로 문장을 끊어 주어야 하는데 비속어를 다른 단어와 복합해서 사용할 경우 단어를 분리 시켜주지 않으면 올바른 필터링이 이루어지기 어렵다. 정상적인 단어라면 단어사전을 통한 검색을 통해 분리가 가능하겠지만 은어나 고유명사 등이 포함될 경우 이것을 분리할 방법이 필요하다.
2. 정상적인 단어의 예외처리 문제이다. 단어 사전에 포함되지 않은 외래어나 사용자 간에 자주 사용되는 은어의 경우 비속어로 판정될 경우 예외처리를 통해 걸러지지 않도록 해야할 필요성이 있다. 이에 대해서는 검출되는 해당 비속어에 대해 예외사전을 만들어 관리해 줄 필요가 있다.
3. 영어 및 특수문자를 이용한 변형에 대한 alignment규칙의 확립이 필요하다. 현재는 단순히 한글의 발음을 통한 변형에 대해서만 표준형 변환이 가능하나 영어의 발음을 이용한 변환이나 특수

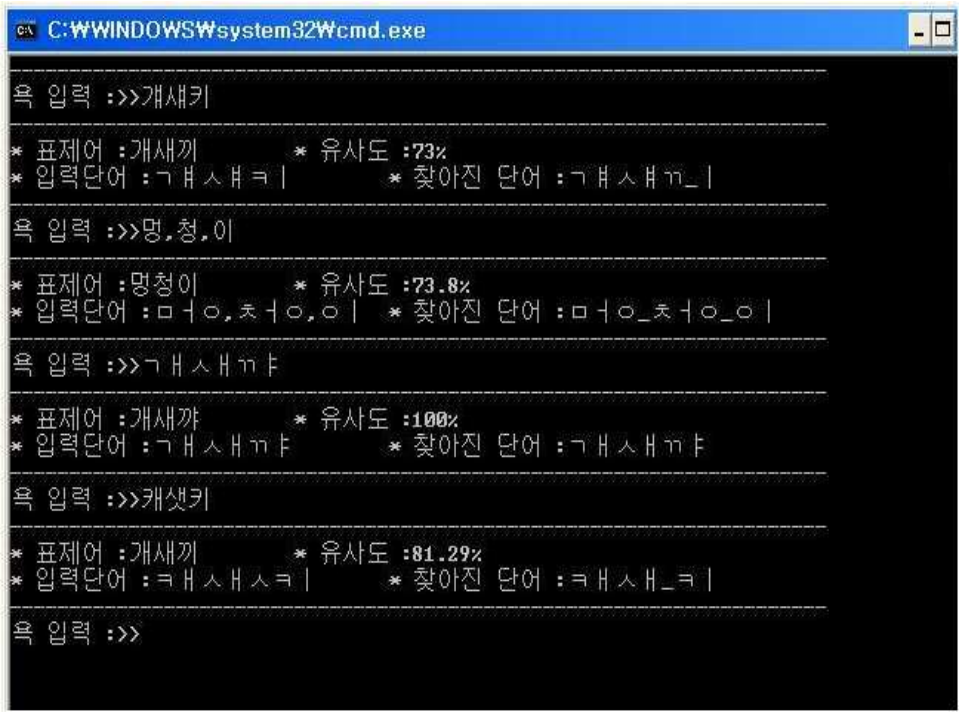

그림 2. 본 시스템의 console 실행 화면. 입력된 변형형 비속어에서 성공적으로 표준형 욕을 검색해내는 것 을 볼 수 있다. 측정된 Local Alignment 값을 단어의 자모가 분리된 단어의 길이로 나눠서 유사도를 계산한다.

문자를 이용한 형태를 통한 변형형에도 대처할 수 있도록 표준화 규칙을 추가할 필요가 있으며 이를 위해서 통계를 통한 연구를 수행해서 주로 사용되는 변형규칙의 수집이 필요할 것이다.

비속어 필터링에 대해서는 이러한 알고리즘적인 검출 방법 외에도 추가적인 연구가 필요하다. 먼저 사용자의 행동에 대한 분석이 필요하다. 게임에서 채팅시스템은 어디까지나 부가적인 요소이므로 지나치게 많은 연산이 있어서는 곤란하고 지나친 규제는 정상적인 단어를 필터링하여 이용에 어려움을 줄 수가 있다. 이를 위해서 사용자에 따라 규제 정도를 차등 적용하는 방법이 있을 수 있다. 평소에 비속어를 많이 사용하는 사용자에 대해서는 규제 등급을 높여서 전체적인 부하를 줄이고 선량한 사용의 불편을 줄이는 방법이다. 그 외에도 비속어가 한번 필터링 되었을 경우 일정시간 동안 규제 등급을 높이는 방법이 있을 수 있다. 일반적으로 사용자는 변형되지 않은 비속어를 사용한 후에 필터링될 경우 변형된 비속어를 다시 입력하는 경향이 있기 때문이다.

비속어 데이터 베이스의 자동적인 update 방법에 대한 연구도 있을 것이다. 비속어는 시대의 흐름에 따라 계속 생겨나고 있으며 유행에 따라서 자주 사용되는 비속어가 있다. 이러한 신종 비속어를 관리자가 직접 입력하는 방법도 있을 수 있으나 사용자의 채팅에서 비속어가 자주 발생하는 분쟁 상태를 발견하여 정상적인 한글 단어사전과 비속어 사전 양쪽 모두에 존재하지 않지만 자주 사용되는 단어가 존재한다면 비속어 입력을 위한 후보군으로 추출하는 방법이 있을 수 있겠다.

참고 문헌

1. 한국게임산업진흥원, "게임언어 건전화 지침서 연구," 2008.
2. Shekhar Dhupelia, "esigning a vulgarity filtering system," in *Game Programming Gems 5*. 2005, Charles River Media.
3. Stefan Kurtz, "Approximate string searching under weighted edit distance," 1996, Carlton University.
4. Gonzalo Navarro, "A guided tour to approximate string matching," in *Game Programming Gems 5*. 2005, Charles River Media.
5. Gonzalo Navarro, "A guided tour to approximate string matching," *ACM Comput. Surv*, vol. 33, no. 1, pp. 31–88, 2001.
6. journal = Know.-Based Syst volume = 20 number = 3 year = 2007 pages = 249-254 Lai C, title = An empirical study of three machine learning methods for spam filtering, ," .
7. Chen H Wei C and Cheng T, "Effective spam filtering: A single-class learning and ensemble approach," *Decis. Support Syst*, vol. 45, no. 3, pp. 491–503, 2008.
8. Feamster N Ramachandran A and Vempala S, "Filtering spam with behavioral blacklisting," *In Proceedings of the 14th ACM Conference on Computer and Communications Security (Alexandria, Virginia*, pp. 342–351, 2001.
9. Cheng V and Li C H, "Personalized spam filtering with semi-supervised classifier ensemble," *In Proceedings of the 2006 IEEE/WIC/ACM international Conference on Web intelligence*, pp. 195–201, 2001.