실시간 채팅 프로그램을 위한 변형 비속어 필터링 시스템

Coined Vulgar words Filtering System for Realtime chat program

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

ABSTRACT

채팅 프로그램은 온라인 게임의 핵심적인 부분의 하나로 사용자 간의 커뮤니케이션에 대한 편의성은 개발자가 가장 중요시 하는 부분이다. 그러나 온라인 상의 언어 폭력은 나날이 심해져서 심각한 사회 문제로 발전되어 가고 있다. 이러한 언어폭력의 피해를 최소화 하기 위해서 사전에 비속어 사용을 차단해줄 필요가 있으며 그것을 위해서는 효과적인 비속어 필터링 시스템의 개발이 시급한 문제이다. 일반적으로 일정 길이 이상의 내용을 가지는 게시판 글의 경우 문서 분류 기법을 이용하는 SPAM 필터링 방법을 응용하여 비속어 필터링이 가능하나 채팅의 경우 짧은 문장을 통해 커뮤니케이션이 이루어지기에 그러한 기법의 적용이 쉽지 않다. 사용자의 편의성을 해치지 않는 실시간 비속어 필터링 시스템을 제안하고자 한다.

KEYWORDS vulgar word, language processing, HCI

1 서 론

온라인 게임은 경쟁을 유도하여 사람들의 흥미를 유도하는 그 특성상 다툼이 자주 일어나고 언어 폭력의 피해에 늘 노출 되어 있는 곳이다. 그래서 여러 온라인게임에서는 비속어 필터링 시스템을 사용하고 유저의 신고를 받아서 비속어 사용자에게 제재를 가하는 등 피해를 막기 위해 여러가지 노력을 기울이고 있다. 그러나 비속어 필터링 시스템은 간단한 변형만으로도 쉽게 비켜나가는 것이 가능하고 사용자가 신고한다고 해서 정황을 정확히 파악할 수 없는 이상 쉽게 제재를 가하기도 어렵다. 원활한 커뮤니케이션은 온라인 게임의 중요한 핵심 부분이라고 할 수 있기 때문에 비속어 필터링 시스템의 제약을 심하게 강화하여 의사소통에 제약을 가하는 것도 바람직한 일이라고는 할 수 없다.

일반적인 온라인 BBS의 게시물 같은 경우 어느 정도의 분량이 있어서 SPAM 필터링에서 주로 사용되는 문서 분류기법이 해결책이 될 수 있다. 그러나 채팅의 경우 짧은 문장이나 단어를 통해 커뮤니케이션이 이루어지고 실시간으로 진행되기 때문에 일반적인 SPAM 필터링 기법을 그대로 적용하기는 어렵다. 특히 비속어가 필터링 되더라 하더라도 사용자가 즉시 비속어를 변형시켜서 필터링을 벗어날 수 있기 때문에 완벽한 비속어 필터링은 어렵다고 할 수 있다. 예를 들어 "개새끼"의 경우 그대로 사용할 경우 대부분의 채팅시스템에서 필터링이 이루어지지만 "개쉐이", "개샛키" 등의 변형단어가

이미 쓰이고 있고 이러한 변형 단어를 등록하더라도 사용자가 지속적으로 변형을 시도한다면 언젠가는 사용가능한 변형 비속어를 발견해내게 된다.

본인은 이미 단어 단위로 처리할 경우 변형 비속어에 대해서 어느정도 대응이 가능한 비속어 필터링 시스템을 개발하였다. 그러나 실제로 채탱에서 사용되는 문장은 띄워쓰기나 문법, 맞춤법 등을 정확히 지키지 않기 때문에 정확한 parsing이 힘들고 특히 비속어를 사용하여 다툼이 일어날 경우 이러한 경향은 특히 심해진다고 할 수 있다. Semi-Global Alignment를 이용하는 변형 비속어 필터링 시스템의 특성상 상당히 많은 연산량을 필요로 하고 적절한 parsing없이 문장을 window slide 방식으로 스캔할 경우 문장의 글자수에 따라 검사 횟수가 비례해서 늘어나기 때문에 CPU에 상당한 부하가 걸리게 된다.

정상단어의 필터링도 큰 문제라고 할 수 있다. 일부 사용자는 비속어 필터링 시스템이 만족할 만한 성능을 보여주지 못하면서 오히려 정상적인 대화를 방해할 뿐이라며 제거를 요구하기도 한다. 변형 비속어 필터링 시스템의 경우 특히 이러한 부분이 문제되는데 비속어와 유사한 일반단어가 존재한다면 비속어 필터링 시스템은 이 단어 역시 필터링해버리게 된다. 단순한 단어 단위의 필터링에서는 정상단어 사전을 통한 사전 검증을 통해서 정상단어의 필터링을 막을 수 있었으나 문장 단위의 필터링의 경우 단어간의 결합을 통해서 문제가 생길 수 있기 때문에 이러한 부분에 대한 고려가 필요할 것이다. 본 보고서에서는 단어 단위의 필터링 시스템을 문장단위의 필터링에 적용할 때의 문제점과 현재 진행된 그 해결방안 그리고 미해결된 문제점에 대해서 서술하고자 한다.

2 System Overview

Semi-global alignment를 이용한 비속어 필터링은 일반단어 필티링 문제와 속도에서 문제가 있기 때문에 그것만을 이용해서 필터링을 수행하는 것은 성능에 문제가 생기게 된다. 이러한 문제를 막기 위해서 비속어 필터링은 3개의 단계를 거쳐서 진행되게 된다.

먼저 비변형 비속어 필터링이다. 이미 알려진 비속어를 Hash 등을 이용한 단순한 검색 방법으로 검증하여 비속어 사용여부를 검증하는 것이다. 복잡한 알고리즘이 사용되지 않고 효율적인 검색방법을 사용할 수 있기 때문에 빠른 속도로 비속어를 검증할 수 있다. 더불어 일반 단어 검증을 통해서 필터링 되어 버릴 수 있는 일반단어+일반단어의 비속어를 필터링 해낼 수 있다. 해당 단계에서 비속어가 검출될 경우 이후 단계에 대한 검증이 필요 없기 때문에 전체적인 속도향상에 기여할 수 있다.

다음으로 일반단어의 검증이다. 이 단계의 의미는 두가지가 있는데 하나는 다음 단계에서 이루어질 변형 비속어 필터링에서 검사할 문자의 수를 줄이는 것이고 또다른 하나는 비속어와 유사한 문자로 이루어져 있어서 비속어와 semi-global alignment 유사도가 높게 나와서 필터링 될 수 있는 단어를 미리 제거하는 것이다. 그리고 후에 서술될 속도를 높이기 위한 parsing 기법에 사용될 수 있다.

최종적으로 변형 비속어 필터링이다. 많은 연산처리를 필요로 하지만 빈칸사용, 특수문자, 발음 변형 등을 통한 비속어 사용을 방지할 수 있다. 사용자의 대부분이 변형 비속어를 남발하는 극단적인

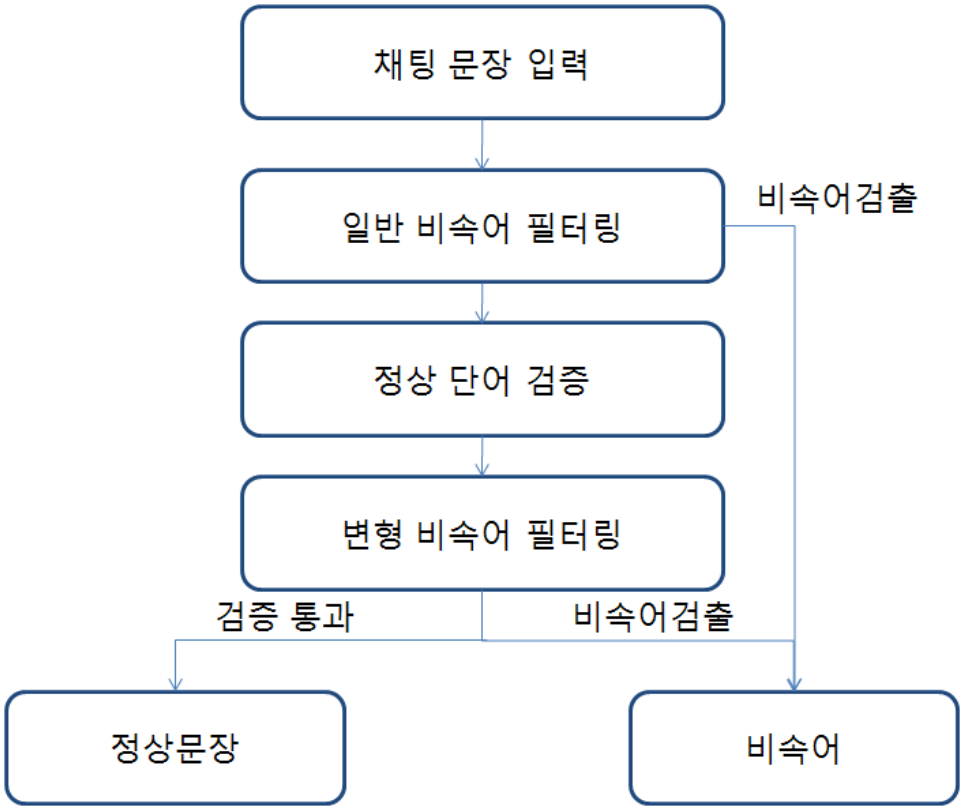

그림 1. 개발 중인 비속어 필터링 시스템의 overview. 변형 비속어 필터링 전에 비변형 비속어, 일반 단어를 검증하고 비속어를 필터링 하는 방식으로 속도와 정확성을 높일 수 있다.

상황이 아니라면 이미 앞의 단계에서 상당량의 문자를 걸러 내기 때문에 실시간 처리에 크게 무리가 없다고 할 수 있다.

3 Sentence Parsing

앞서 언급한 것처럼 채팅에 사용되는 문장은 띄워쓰기와 문법이 정확히 지켜진다는 보장이 없다. 대부분의 온라인 게임에서 자신들만의 은어와 축약어를 사용하고 이모티콘 등이 사용되기 때문에 일반적인 한글 맞춤법 등에 사용되는 Parsing 기법은 사용되기 어렵다. 그렇다고 문장 전체를 통채로 스캔해서 비속어를 찾을 경우 속도면에서 문제가 생기게 된다.

위 알고리즘은 가장 간단한 단어 검색 알고리즘이다. 이 알고리즘의 Complexity는 O(size of S × N)이다. N은 고정된 값이므로 실질적으로는 문장의 길이에 비례하여 연산량이 늘어난다고 할 수 있다. 그러나 Hash를 사용하여 단일 연산량이 적은 "일반 비속어 필터링"과 "정상 단어 검증"의 경우 큰 문제가 없으나 단일 연산량이 많은 변형 비속어 필터링의 경우 긴 문장을 처리하는데 많은 연산량이 필요하게 된다. 반면에 짧은 문장의 경우 연산량이 반이하로 줄어들게 된다. 예를 들어 문장의 길이가 N과 같을 경우 연산 회수를 M이라할때

Algorithm 1 문장 단어 검색 알고리즘

```
Input: S(chat sentence data), N(maxmum word size), W(word dictionary)
Output: T(vulgar words)
    While i is smaller than size of S
        j < − N
        While j is bigger than 1
            if S.substring(i, j) in W then T < − true
            j < − j - 1
        end While
        i < − i + 1
    end While
end procedure
```

$$M = N + (N-1) + (N-2) + ... + 1 = N(N+1)/2$$

필요 연산량이 반으로 줄어드는 것을 알 수 있다. 길이가 짧을 수록 연산량은 줄어들기 때문에 짧은 문장으로 parsing 할 수 있다면 연산량을 대폭 줄여 줄 수 있다. 이것을 위해서는 일반단어 검증 단계에서 문장을 parsing 해 줄 필요가 있다. 단순하게 생각해볼때 일반단어로 검증이 된 영역을 변형 비속어 필터링 때 검색해줄 필요가 없고 그 앞영역과 뒷영역은 정상단어를 기준으로 parsing 해 줄 수 있는 것이다.

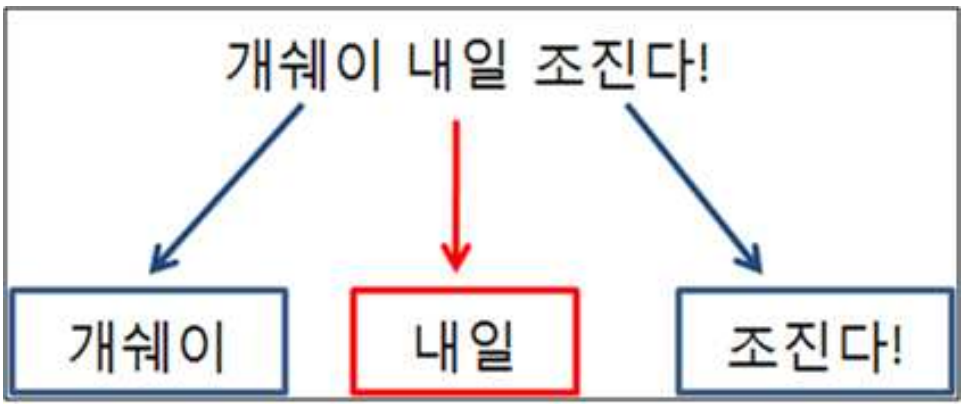

그림 2. 일반단어 검증 단계에서 이루어지는 Parsing 정상단어로 검증이 된 영역을 기준으로 문장을 parsing 해서 개개의 문장의 길이를 줄여준다.

그림 2 에서 보는 바와 같이 일반 단어를 기준으로 문장을 나눠 줌으로써 짧은 단어인 "개쉐이", "조진다!"에 대해 각각 검증하면 되므로 각각 2 회와 3 회의 검증만 이루어지게 된다. 그러나 이 방법에도 문제가 없는 것은 아니다. 정상적인 단어를 사용하지 않고 변형 비속어만으로 긴 문장을 형성한다면 일반 단어 검증에서 parsing 이 수행되지 않아 변형비속어 검증에서 문장 전체를 검색해야 하므로 많은 연산량을 필요로 하게 된다.

4 결론

본 보고서에서는 개발 중인 비속어 필터링 시스템의 문제와 그 해결방안에 대해서 제안하였다. 한글의 특성을 이용한 변형 비속어에 대해서 효과적으로 대처하기 위한 방법으로 다음과 같은 문제점과 방안이 검토되었다.

1. 정상단어의 필터링 문제 : 변형 비속어에 대응하기 위해
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 특수문자를 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 update방법에 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.