ChatGPT

C# 기반 Profanity Filter 라이브러리 비교 조사

ProfanityDetector (Stephen Haunts)

- **특징**: C#(.NET Standard 2.0) 기반의 욕설 감지 라이브러리 [1]. 내장된 욕설 목록을 사용하여 문자열 내 욕설 존재 여부를 검사하거나, 발견된 욕설 단어들을 추출할 수 있다. GitHub★157(리포지토리), NuGet 패키지(`Profanity.Detector`)로 제공되며, 기본 리스트 대신 사용자 정의 리스트로 교체 가능하다 [1] [2].
- **API/사용법**: `ProfanityFilter` 객체를 생성한 뒤 `IsProfanity(string)` 메서드로 단어를 검사하고, `DetectAllProfanities(string)`로 문장 내 모든 욕설을 리스트 형태로 얻을 수 있다 [3]. 또한 `CensorString(string)`을 호출하면 문장 내 욕설 단어를 `*` 등 지정 문자로 대체하는 기능을 제공한다 [4].
- **필터링 방식**: 단순 단어 매칭 기반(대소문자 구분 없음)으로 구현되어 있으며, 소위 *Scunthorpe* 문제를 완화하기 위해 허용 단어(allow list)를 지원한다. 예를 들어 "Scunthorpe" 같은 단어에 포함된 욕설 패턴은 이 단어 전체가 목록에 없으면 검출하지 않는다 [5]. 정규식보다는 사전 조회 방식이라 구현이 비교적 단순하다.
- **성능**: 공식 벤치마크는 없으나, DotnetBadWordDetector 등의 자료에 따르면 동일한 환경에서 동작 시 ML 기반 라이브러리보다 속도가 느릴 수 있다 [6]. 실제로 DotnetBadWordDetector 측정 결과 ProfanityDetector는 100회 연속 예측에 약 102.08ms가 소요되었는데 반해 해당 라이브러리는 약 4.19ms로 보고되었다 [6]. (이 결과는 학습 모델 vs 정적 리스트 비교를 위한 참고치임)
- **링크**: GitHub 리포지토리 – [stephenhaunts/ProfanityDetector](#) [1], NuGet 패키지 – [Profanity.Detector](#) [2].

.NET Bad Word Detector (FelipeLuz)

- **특징**: .NET 8 기반 최신 욕설 감지 라이브러리로, ML.NET으로 학습된 로지스틱 회귀 모델을 사용해 욕설 여부를 판별한다 [7]. 영어·포르투갈어·스페인어 등 3개 언어를 지원하며, 단순 단어 목록 대신 머신러닝 기법을 적용해 변형된 욕설(예: "h0us3" 등)도 인식할 수 있도록 설계되었다 [8].
- **API/사용법**: `ProfanityDetector` 클래스를 인스턴스화한 후 `IsProfane(string)`, `IsPhraseProfane(string)` 메서드로 욕설 감지를 수행한다. `GetProfanityProbability(string)`, `GetPhraseProfanityProbability(string)`로 욕설 확률을 얻을 수도 있으며, `MaskProfanity(string, char)`를 사용하면 탐지된 욕설을 지정 문자(예: `*`)로 치환할 수 있다 [9]. 사용 예시로, `var detector = new ProfanityDetector(allLocales: true);` `detector.IsProfane("example");` 식으로 호출한다.
- **필터링 방식**: 학습된 분류 모델을 통해 입력 문자열을 분석한다. 단순 리스트 방식과 달리, 사람이 표시한 대량의 데이터로 학습된 모델을 내장하여 단어 유사도나 패턴을 기준으로 판별하므로, 새로운 형태의 욕설도 일부 포착 가능하다 [8]. 단어 하나뿐 아니라 구 전체에 욕설이 포함됐는지 등도 판단할 수 있으며, 단일 인스턴스로 다양한 형태를 빠르게 처리할 수 있다.
- **성능**: 벤치마크 수치가 공개되어 있다. 동일 환경에서 ProfanityDetector 대비 최대 618배 빠른 속도를 보이며, 예를 들어 100회 연속 예측 시 약 4.19ms로 동작해 ProfanityDetector(약 102.08ms)보다 약 24배 빠른 것으로 보고되었다 [6]. 학습 모델 크기가 작아 메모리에 상주시켜두면 반복 검사 시 높은 처리량을 보인다. (정확도: 약 98.4%로 학습되어 있음 [6])

- **링크**: GitHub 리포지토리 – FelipeLuz/dotnet-bad-word-detector-and-filter [7], NuGet 패키지 – DotnetBadWordDetector [6].

Censored (James Montemagno)

- **특징**: 지정한 욕설 단어 리스트를 기반으로 문자열 내 욕설을 검열하는 간단한 라이브러리이다 [10]. 목록에 포함된 단어를 만날 때마다 자동으로 `*****`(혹은 지정한 문자)로 치환하며, 욕설 단어 존재 여부만을 판단하거나 실제 치환까지 손쉽게 처리할 수 있다.
- **API/사용법**: 생성자에 비속어 리스트(`List<string>`)를 전달하면 `new Censor(words)`와 같이 객체가 생성된다. 이후 `CensorText("문장")`을 호출해 해당 문장의 욕설 단어를 `*` 등으로 치환하며, `HasCensoredWord("문장")`로 욕설 포함 여부만 검사할 수도 있다 [11]. 예시: `new Censor(new List<string>{"gosh","drat"}).CensorText("Oh gosh!");` → `Oh ****!` [11].
- **필터링 방식**: 문자열 내 단순 단어 비교 및 치환 방식이다. 정규식보다는 와일드카드(`*`) 기반 간단 매칭을 지원할 뿐, 탐지된 욕설이 다른 단어에 포함돼 있거나 변형된 경우를 자동으로 배제하지 않는다. 즉, 정확히 리스트에 등록된 형태의 단어만 찾아내며, Scunthorpe 문제나 컨텍스트 분석 기능은 없다.
- **성능**: 매우 경량 구현체로 별도의 성능 벤치마크가 제공되지는 않는다. 단어 리스트 크기나 문장 길이에 비례하므로, 일반적인 채팅·로그 필터링 정도의 용도에서는 충분히 빠르게 동작할 것으로 보인다.
- **링크**: GitHub 리포지토리 – jamesmontemagno/Censored [10], NuGet 패키지 – Censored로 배포된다 [11].

기타 참고 라이브러리

- **BogaNet.BadWordFilter**: Unity용 BadWordFilter PRO를 .NET8으로 이식한 라이브러리다 [12]. 25개 언어에 대해 5,000개 이상의 욕설 정규표현식을 지원하며, URL/이메일·이모지·과도한 대문자·구두점 필터 등을 통합한 `Pacifier.Instance` 클래스로 한 번에 검사/치환할 수 있다 [13] [14]. 다국어 지원이라는 장점이 있으나 크기가 커서 무거운 편이다.
- **한국어 전용 필터**: 예를 들어 Xim-ya/korean_profanity_filter 같은 라이브러리도 있지만, 본 비교에서는 주로 글로벌 언어를 다루는 범용 라이브러리들을 중심으로 검토했다 [12].

요약: FastChatFilter의 벤치마크 대상 라이브러리로는 전통적 리스트 기반 라이브러리(ProfanityDetector), 기계학습 기반 신생 라이브러리(DotnetBadWordDetector), 간단 치환 라이브러리(Censored) 등을 선정했다. 각각의 API 방식, 필터링 로직, 성능 정보를 참고해 FastChatFilter와의 특성 비교·테스트를 진행할 수 있다.

---

[1] [3] [4] [5] GitHub - stephenhaunts/ProfanityDetector: This is a simple library for detecting profanities within a text string.
https://github.com/stephenhaunts/ProfanityDetector

[2] NuGet Gallery | Profanity.Detector 0.1.8
https://www.nuget.org/packages/Profanity.Detector

[6] [7] [8] [9] GitHub - FelipeLuz/dotnet-bad-word-detector-and-filter: .NET library that uses machine learning to detect bad words (profanity) within a string.
https://github.com/FelipeLuz/dotnet-bad-word-detector-and-filter

[10] [11] GitHub - jamesmontemagno/Censored: A .NET Profanity Censoring Library
https://github.com/jamesmontemagno/Censored

12 13 14 NuGet Gallery | BogaNet.BadWordFilter 1.4.0
https://www.nuget.org/packages/BogaNet.BadWordFilter