DOI 10.37882/2223-2966.2024.3-2.07

ПОИСК ДЕСТРУКТИВНОГО КОНТЕНТА В ТЕКСТЕ

SEARCHING FOR DESTRUCTIVE CONTENT IN TEXT

A. Dzhurov

Summary. This article discusses the use of the sklearn library and the WordNet database for text classification. The principle of operation of stemming is shown and an example of its implementation in python is shown. Pipeline steps for building a model and processing data are described. Various approaches to text preprocessing were considered, including tokenization, stopword removal, and lemmatization. The use of the WordNet database made it possible to carry out semantic analysis of the text and improve the quality of text classification. Experimental results showed that combining sklearn methods and the WordNet database is an effective approach for text classification. A demonstration of the operation of the developed module with conclusions of the results is shown, and a general diagram of the operation of the module is shown with a description of its operation.

Keywords: sklearn, WordNet, Stemmer, text classification, Python, Pipeline.

Джуров Александр Андреевич
Аспирант, Донской государственный технический университет
sashaz1696@yandex.ru

Аннотация. В данной статье рассматривается использование библиотеки sklearn и базы данных WordNet для классификации текста. Показан принцип работы стемминга, а также пример реализации его в python. Расписаны шаги Pipeline для построения модели и обработки данных. Были рассмотрены различные подходы к предобработке текста, включая токенизацию, удаление стоп-слов и лемматизацию. Использование базы данных WordNet позволило провести семантический анализ текста и улучшить качество классификации текста. Результаты экспериментов показали, что комбинирование методов sklearn и базы данных WordNet являются эффективным подходом к классификации текста. Показана демонстрация работы разработанного модуля с выводами результатов и общая схема работы модуля с описанием его работы.

Ключевые слова: sklearn, WordNet, Stemmer, классификация текста, Python, Pipeline.

Актуальность

Актуальность выявления и блокировки деструктивного контента обусловлена необходимостью защиты общества от негативных последствий, которые могут возникнуть в результате распространения такого контента. Деструктивный контент может включать в себя информацию, которая призывает к насилию, пропагандирует экстремизм и терроризм, распространяет ненависть и дискриминацию, а также вводит людей в заблуждение.

Выявление и блокировка деструктивного контента являются важными мерами по защите общества от негативных последствий. Это позволяет предотвратить возможные преступления и конфликты, сохранить стабильность и безопасность в обществе. Кроме того, это способствует развитию информационной грамотности и критического мышления у граждан страны, что помогает им различать полезную информацию от вредной. Поэтому разработка модуля для выявления деструктивного контента в тексте является актуальной.

Целью является разработка модуля, для выявления деструктивного контента в тексте.

Аналоги

Sentiment Analysis API

Sentiment Analysis API — это автоматизированный процесс понимания чувств, лежащих в основе мнений, письменных или устных. Другими словами, вы можете определить, является ли мнение негативным, нейтральным или положительным.

Достоинства:
— использование API для поиска настроений;
— возможность определять настроение текста.

Недостатки:
— платная работа с API;
— классификация настроения только на английском языке;
— классификация только настроение в тексте, не позволяет определить деструктивный контент.

Антивирус

Антивирус — это программа, которая борется с вирусами. Сейчас, антивирусы также помогают ограничивать детей от опасных сайтов, на которых может использоваться деструктивный контент.

Достоинства:
— классифицированы сайты;
— большая база классифицированных сайтов;
— возможность блокировки доступа к сайту;
— ограничения по времени использования интернета.

Недостатки:
— в большинстве случаев платный функционал;
— не всегда актуальная база сайтов.

Окулус

Окулус — система автоматического поиска запрещенного контента «Окулус».

Достоинства:
— распознование изображений и символов, противоправных сцен и действий, анализирует текст в фото- и видеоматериалах;
— автоматически обнаруживает такие правонарушения, как экстремистская тематика, призывы к массовым незаконным мероприятиям, суициду, пронаркотический контент, пропаганда ЛГБТ.

Недостатки:
— система занимается лишь классификацией сайтов;
— можно легко обойти заблокированные сайты с помощью VPN.

Общие алгоритмы и модули, которые использовались

Рассмотрим разработанный модуль для обнаружения деструктивного контента в тексте на языке программирования python.

Как происходит классификация.

Для начала выполняем стемминг с помощью библиотеки Stemmer.

Stemmer — это алгоритм или программа, которая выполняет стемминг, то есть находит основу слова (стеб) и удаляет окончание (оформление) [6]. Например, стеб слова «книга» — «кни», а оформление — «а». Таким образом, Stemmer может помочь в поиске всех форм одного слова в тексте.

Принцип работы Stemmer на Python заключается в использовании словаря, который содержит все возможные слова и их основы.

1. Сначала необходимо загрузить словарь Stemmer. Для этого можно использовать модуль stemmer.
2. Затем нужно создать экземпляр класса Stemmer, который будет использоваться для стемминга слов.
3. Далее, чтобы применить стемминг к слову, нужно вызвать метод stem() экземпляра Stemmer, передав в него слово в качестве аргумента.
4. Метод stem() применяет стемминг к слову, возвращая его основу.
5. Наконец, можно сохранить результат в переменную или использовать его для дальнейшей обработки текста.

Пример использования Stemmer на Python:

```
from stemmer import Stemmer
# Загрузка словаря Stemmer
stemmer = Stemmer('russian')
# Пример использования Stemmer
word = «книга»
stemmed\_word = stemmer.stem(word)
print(stemmed\_word) # Вывод: «кни»
```

Для разработки модуля использовалась библиотека Scikit-Learn

SKLearn (Scikit-Learn) — это библиотека машинного обучения для языка программирования Python [5]. Она предоставляет набор инструментов для обработки данных, построения моделей машинного обучения и оценки их качества.

С помощью SKLearn можно решать различные задачи машинного обучения, такие как классификация, регрессия, кластеризация, анализ текстов и изображений, и многое другое. Библиотека также предоставляет множество алгоритмов для работы с данными, включая методы предобработки, методы выбора признаков и метрики для оценки моделей.

SKLearn является одной из самых популярных библиотек машинного обучения и используется во многих проектах и исследованиях. Она имеет простой и интуитивно понятный интерфейс, что делает ее доступной для начинающих пользователей машинного обучения.

А модель для обучения использовалась Pipeline.

Pipeline — это концепция в области машинного обучения, которая представляет собой последовательность шагов обработки данных и построения модели [1]. В общем смысле, pipeline представляет собой конвейер или цепочку, в которой каждый шаг выполняет определенную операцию над данными.

В контексте машинного обучения, pipeline обычно используется для решения задач классификации, регрессии или кластеризации. Он может включать в себя следующие этапы:

1. Загрузка и предобработка данных: В этом шаге данные загружаются из источника и проходят

предварительную обработку, такую как очистка, нормализация, преобразование признаков и т.д.
2. Разделение данных: Данные разделяются на обучающую и тестовую выборки. Это позволяет оценить качество модели на независимых данных.
3. Построение модели: На этом этапе выбирается и настраивается модель машинного обучения. Это может быть линейная регрессия, дерево решений, нейронная сеть и т.д.
4. Оценка модели: После построения модели, ее необходимо оценить на тестовой выборке. Это позволяет определить точность и качество модели.
5. Туннелирование модели: Если модель показывает неудовлетворительные результаты, можно провести дополнительные шаги для улучшения ее производительности, такие как настройка гиперпараметров, подбор признаков и т.д.
6. Развертывание модели: После успешного обучения и оценки модели, она может быть использована для предсказания на новых данных.

Pipeline предоставляет удобный способ организовать и автоматизировать процесс машинного обучения. Он позволяет упростить и ускорить разработку моделей, а также обеспечить их повторяемость и надежность.

Также в разработанной модели используется лексическая база данных WordNet.

WordNet — это лексическая база данных, которая содержит семантические и лингвистические связи между словами [4]. Она разработана для английского языка и представляет собой набор синонимов, гипонимов, гиперонимов и антонимов для каждого слова.

WordNet организован в виде сети синсетов (synsets), которые представляют собой группы семантически связанных слов. Каждый синсет имеет набор свойств и связей с другими синсетами.

WordNet позволяет проводить семантический поиск и анализ текста, а также предоставляет информацию о значениях и отношениях между словами. Он широко используется в области обработки естественного языка, машинного обучения и других приложениях, связанных с текстом.

WordNet доступен через библиотеку nltk (Natural Language Toolkit) для языка программирования Python, что позволяет использовать его в различных задачах обработки текста, таких как определение синонимов, поиск схожих слов и оценка тональности текста.

Текст также перед обработкой проходит процессы: *токенизации, удаления стоп-слов и лемматизацию*.

Токенизация в Python — это процесс разделения текста на отдельные токены или слова [3]. Это важная операция в обработке естественного языка и используется для подготовки текстовых данных для дальнейшего анализа или обработки.

В Python существует несколько библиотек, которые предоставляют функции для токенизации текста. Одна из самых популярных библиотек для токенизации в Python — это NLTK (Natural Language Toolkit). NLTK предоставляет различные методы токенизации, такие как разделение текста на слова или на более мелкие единицы, например, символы или подслова.

Токенизация является первым шагом в обработке текста и может быть полезна для различных задач, таких как анализ тональности, классификация текста, машинный перевод и другие.

Удаление стоп-слов в Python относится к процессу удаления общих слов, которые не несут смысловой нагрузки в тексте, таких как артикли, предлоги и местоимения. Это важная операция в предобработке текста перед его анализом или обработкой [2].

Для удаления стоп-слов в Python можно использовать библиотеку NLTK (Natural Language Toolkit). NLTK предоставляет список стоп-слов для различных языков, включая русский.

Удаление стоп-слов может помочь упростить текст и удалить шумовые слова, что может улучшить качество анализа или обработки текста.

Лемматизация в Python — это процесс приведения слов к их базовой форме (лемме) [7]. Это важная операция в предобработке текста, которая помогает упростить текст и сократить его размерность.

В Python существует несколько библиотек, которые предоставляют функции для лемматизации текста. Одна из популярных библиотек для лемматизации в Python — это pymorphy2. Pymorphy2 использует словарь русского языка и позволяет получить лемму для каждого слова.

Лемматизация помогает унифицировать слова в тексте и упростить его анализ или обработку. Это особенно полезно в задачах классификации текста или поиска информации.

Работа модуля

На рисунке 1 показана общая схема работы модуля

0 — текст не является деструктивным

1 — Текст является деструктивным

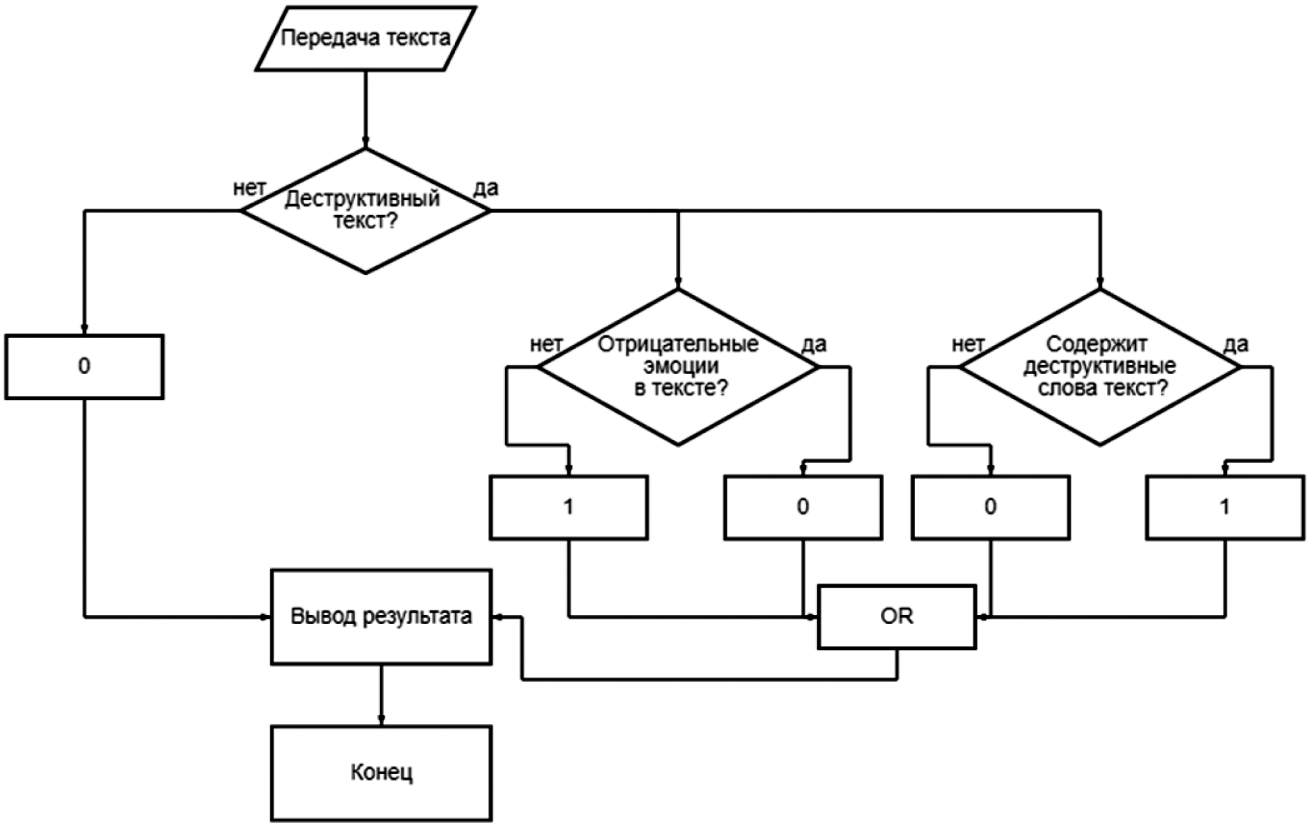

Рис. 1. Общая схема работы модуля

Принцип работы:

1. Передается текст для его анализа
2. Производится токенизация, удаление стоп-слов и лемматизация.
3. С помощью обученной нейронной сети sklearn и pipeline определяем является ли текст деструктивным? Если текст не является деструктивным, тогда программа выдает ответ, что текст не деструктивный (0), если же текст деструктивный, то требуется дополнительный анализ.
4. Проверка текста на эмоции: если эмоции отрицательные, тогда все хорошо и возвращаем 0 (говорить о плохом плохо, это хорошо), если же эмоции положительные, тогда возвращаем 1.
5. Проверка текста на деструктивные слова в тексте: если есть деструктивные слова, то возвращаем 1, если нет, то возвращаем 0.
6. Вывод результата при дополнительном анализе текста происходит по формуле 1.

Деструктивный текст = Эмоциональная оценка OR
Наличие деструктивных слов (1)

К примеру, есть предложение, в котором говорится о негативном воздействии курения на организм человека. Так как тема идет о курении, то проверка деструктивного контента может показать, что текст деструктивный и отправит его на дополнительный анализ. Так как эмоционально текст отрицательный (говорит о негативном воздействии курения) и нет деструктивных слов в тексте, то текс считается не деструктивным.

Тестирование модуля

Возьмем для примера одно из предложений о вреде курения и после его переделаем немного для того, чтобы показать работу модуля.

Первое предложение: «Ежегодно табак приводит почти к 7 миллионам случаев смерти»

Второе предложение: «Ежегодно табак приводит бл*ть почти к 7 миллионам случаев смерти»

Результат работы модуля для первого предложения изображен на рисунке 2.

С точностью 97,14 % программа считает, что предложение не является деструктивным.

Результат работы модуля для второго предложения изображен на рисунке 3.

С точностью 97,16 % программа считает, что предложение является деструктивным. Наличие деструктивного слова изменило конечный результат работы модуля.

```
Test score: 97.14 %
Введите текст для анализа: Ежегодно табак приводит почти к 7 миллионам случаев смерти
Результат: Notoxic
```

Рис. 2. Результат работы модуля для первого предложения

```
Test score: 97.16 %
Введите текст для анализа: Ежегодно табак приводит блять почти к 7 миллионам случаев смерти
Результат: toxic
```

Рис. 3. Результат работы модуля для второго предложения

В целом из 100 предложений (и деструктивных и не деструктивных), программа в 97 случаев не давала ошибок, что дает ей точность 97 %.

Выводы

В данной статье были исследованы возможности использования библиотеки sklearn и базы данных WordNet для классификации текста. Также были рассмотрены различные подходы к предобработке текста, включая токенизацию, удаление стоп-слов и лемматизацию.

Использование базы данных WordNet позволило провести семантический анализ текста, что привело к улучшению качества классификации. Результаты экспериментов подтвердили, что комбинирование методов sklearn и базы данных WordNet является эффективным подходом к классификации текста.

Однако, следует отметить, что использование WordNet может быть ограничено для некоторых языков и может потребовать дополнительной предобработки данных. Также, для достижения более высокой точности классификации, важно проводить дополнительные исследования и эксперименты с различными моделями и параметрами.

В целом, данная статья подтверждает потенциал комбинирования библиотеки sklearn и базы данных WordNet для классификации текста и предоставляет полезные рекомендации для дальнейших исследований в этой области

ЛИТЕРАТУРА

1. Алпатов А.Н., Попов К.С., Чесалин А.Н. Анализ точности моделей машинного обучения с использованием методов векторизации для задач классификации разнородных текстовых данных // International Journal of Open Information Technologies. 2022. №7.
2. Гальченко Ю.В., Нестеров С.А. Классификация текстов по тональности методами машинного обучения // SAEC. 2023. №3.
3. Герасименко Е.М., Стеценко В.В. Анализ тональности текстовых отзывов с применением тональных словарей и кардинальности нечеткого множества // Известия ЮФУ. Технические науки. 2022. №5 (229).
4. Гойло А.А., Садовский М.Е., Никифоров С.А. Интеграция лексических баз данных для решения проблемы синонимии в естественно-языковых интерфейсах // SAI. 2023. №Special Issue 3.
5. Королев И.Д., Акинфиев Д.В. Способ автоматизированного формирования обучающего набора данных для алгоритмов машинного обучения классификации электронных документов // ИВД. 2023. №10 (106).
6. Мотовских Л.В. Выделение ключевых слов для классификации текстов // Вестник Московского государственного лингвистического университета. Гуманитарные науки. 2020. №9 (838).
7. Челышев Э.А., Оцоков Ш.А., Раскатова М.В., Щёголев П. Сравнение методов классификации русскоязычных новостных текстов с использованием алгоритмов машинного обучения // ВК. 2022. №1 (45).

Журнал «Современная наука: актуальные проблемы теории и практики»