Р. В. Детскина, К. П. Антоник

ПОДХОДЫ К ФИЛЬТРАЦИИ СОДЕРЖАНИЯ ДОКУМЕНТОВ В ИНТЕРНЕТЕ

В современном мире, где реклама является двигателем торговли, с развитием Интернета и средств общения, проблема нежелательной рекламы и сообщений требует интеллектуального подхода для ее решения. В настоящее время существует несколько алгоритмов фильтрации нежелательной корреспонденции. Однако современные методы борьбы со спамом, основанные на лингвистических признаках, правилах фильтрации сообщений, становятся все менее эффективными, так как требуется увеличение трудозатрат специалистов по защите от спама на поддержание этих признаков и правил в актуальном состоянии. Главными недостатками большинства существующих методов являются ложные тревоги, пропуск «спама», фиксированное количество слов, участвующих в оценке письма. Таким образом, современные методы борьбы со спамом требуют постоянного участия человека для эффективного анализа текста, они не способны самостоятельно вырабатывать эти правила.

Если рассматривать человека как средство борьбы со спамом, то можно сказать, что он обладает способностью обнаружения признаков спама, основываясь на собственном опыте и предпочтениях, знаниях о добровольных новостных и рекламных подписках, обучаемостью, его работа не сводится к шаблонам и потому более эффективна. Именно поэтому задача создания средства борьбы со спамом сводится к наделению этого средства навыками и качествами, присущими человеку: способностью к обучению, системой предпочтений и исключений, анализом контекста, системой принятия решений.

Байесовская фильтрация спама – метод для фильтрации спама, основанный на применении наивного байесовского классификатора (НБА), опирающегося на прямое использование теоремы Байеса.

Наивный байесовский алгоритм – это алгоритм классификации, основанный на теореме Байеса с допущением о независимости признаков. Другими словами, НБА предполагает, что наличие какого-либо признака в классе не связано с наличием какого-либо другого признака.

Первой известной программой, фильтрующей почту с использованием байесовского классификатора, была программа iFile Джейсона Ренни, выпущенная в 1996 году. Программа использовала сортировку почты по папкам. Первая академическая публикация по наивной байесовской фильтрации спама появилась в 1998 году. Вскоре после этой публикации была развернута работа по созданию коммерческих фильтров спама. Однако в 2002 году Пол Грэм смог значительно уменьшить число ложноположительных срабатываний до такой степени, что байесовский фильтр мог использоваться в качестве единственного фильтра спама.

Модификации основного подхода были развиты во многих исследовательских работах и внедрены в программных продуктах. Многие современные почтовые клиенты осуществляют байесовское фильтрование спама. Программное обеспечение серверов электронной почты либо включает фильтры в свою поставку, либо предоставляет API для подключения внешних модулей.

В частности, фильтр, основанный на алгоритме Байеса, имеет следующие достоинства по сравнению с более простыми методами:

- уникальный для каждой организации набор данных, что делает более сложным обход фильтра;
- просмотр полного нежелательного сообщения, а не только ключевых слов или известных подписей;
- многоязычность.

Модели на основе НБА достаточно просты и крайне полезны при работе с очень большими наборами данных. При своей простоте НБА способен превзойти даже некоторые сложные алгоритмы классификации. Наивные байесовские классификаторы были успешно применены во многих областях, в частности, в Natural Language Processing, или сокращенно NLP (обработке естественного языка). Существуют и другие альтернативы при решении проблем с NLP, такие как нейронные сети или метод опорных векторов (SVM). Однако простой дизайн наивных байесовских классификаторов делает их очень привлекательными для использования.

Нами была создана компьютерная программа, использующая наивный байесовский классификатор, который способен ответить на вопрос, к какой категории классов «спам» – «не спам» относится электронное сообщение. Данная программа написана на языке программирования Python и использует корпус СМС сообщений в формате CSV в качестве обучающего алгоритма материала.

Для обработки корпуса сообщений используются библиотеки языка Python, направленные на токенизацию и стеммирование сообщений.

```
import pandas
import matplotlib.pyplot as pyplot
from wordcloud import WordCloud
import nltk
nltk.download('punkt')
nltk.download('stopwords')
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk.corpus import stopwords
```

Метод «train» обрабатывает корпус сообщений, помеченных классами «спам» или «не спам», проводя токенизацию и стеммирование, и присваивает веса каждому слову, согласно частоте его употребления. Первый этап, этап разработки функций, сосредоточен на извлечении особенностей текста. Необходимо использовать числовые значения в качестве входных данных для классификатора. Таким образом, интуитивно понятным выбором являются частоты слов, то есть подсчет вхождения каждого слова в сообщении.

```
    spam_set = {}
    ham_set = {}
    spam_count = 0
    ham_count = 0
     def train(path, encoding):
        global spam_count
        global ham_count
        spam_words = ""
        ham_words = ""
         dataset = pandas.read_csv(path, encoding = encoding)
        dataset = dataset[['v1', 'v2']]
        dataset.columns = ['label', 'msg']
         for _, row in dataset.iterrows():
            label = row['label']
            msg = row['msg']
            msg = process_msg(msg)
             if (label == "spam"):
                for word in msg:
                    spam_words += word + " "
                    spam_set[word] = spam_set.get(word, 0) + 1
                spam_count += 1
            else:
                for word in msg:
                    ham_words += word + " "
                    ham_set[word] = ham_set.get(word, 0) + 1
                ham_count += 1
         spam_pic    =    WordCloud(width    =    1024,    height    =
1024).generate(spam_words)
        pyplot.figure(figsize = (20, 15))
        pyplot.imshow(spam_pic)
        pyplot.savefig('spam.png', bbox_inches = 'tight')
         ham_pic    =    WordCloud(width    =    1024,    height    =
1024).generate(ham_words)
        pyplot.figure(figsize = (20, 15))
        pyplot.imshow(ham_pic)
        pyplot.savefig('ham.png', bbox_inches = 'tight')
    def process_msg(msg):
        tokens = word_tokenize(msg)
        tokens = [t for t in tokens if len(t) > 2]
        stop_word = stopwords.words('english')
        tokens = [t for t in tokens if (t not in stop_word)]
        stemfx = PorterStemmer()
        tokens = [stemfx.stem(t) for t in tokens]
        return tokens
```

Метод «test» непосредственно использует наивный байесовский классификатор для определения того, является ли сообщение спамом, то есть, выражаясь формально, вычисляет вероятность того, что сообщение попадет в категорию «спам».