МЕТОДИКА СЕМАНТИЧЕСКОГО АНАЛИЗА ТЕКСТА С ИСПОЛЬЗОВАНИЕМ НАИВНОГО БАЙЕСОВСКОГО КЛАССИФИКАТОРА

И. Ю. Татуревич

Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», Республика Беларусь

Научный руководитель О. А. Кравченко

Семантический анализ текста – это этап в последовательности действий алгоритма автоматического понимания текстов, направленный на установление «содержательных» семантических отношений [1]. Область применения данной технологии достаточно обширна: психология, политология, филология, торговля и др. В области информационных технологий данный тип анализа используется при проектировании поисковых и SEO-алгоритмов, искусственного интеллекта и текстовых фильтров. Семантический анализ, несмотря на свою востребованность, является одной из сложнейших математических задач. Это обусловлено тем, что языки по своей природе плохо поддаются формализации. Как следствие этого, в области автоматизации семантического анализа в настоящий момент нет единых алгоритмов и подходов к решению конкретных задач. Из вышеуказанного можно сделать вывод, что исследования и разработки в данной области являются актуальными и перспективными.

Одной из задач в рамках семантического анализа текста является классификация текстов. Классификация текста необходима для решения таких задач, как борьба со спамом, распознавание эмоциональной окраски текстов, разделение сайтов по тематическим разделам, персонификация рекламы. Одним из способов решения задачи классификации является метод, использующий наивный байесовый классификатор. Наивный байесовский классификатор – один из наиболее примитивных классификаторов, основанных на теореме Байеса с условием выполнения строгой независимости вероятностных компонент. Данное допущение рассматривает каждое слово в тексте отдельно и независимо от остальных.

Одной из основных целей данной работы являлось изучение принципов работы наивного байесового классификатора в контексте семантического анализа текста, позволяющее сделать выводы о целесообразности применения данного метода для решения задач классификации текста. Для тестирования данного метода требуется разработать специальное программное обеспечение, представляющее собой библиотеку. Разработанная библиотека может быть использована в приложениях, исследующих применения байесовского классификатора в задачах семантического анализа текста. В качестве задачи классификации текста рассматривается задача по фильтрации спама.

Разработанное программное обеспечение выполняет следующие функции: принимает на вход файл текстового формата, содержащий набор текстовых сообщений, и определяет процент сообщений, которые не содержат спам. Программа написана на языке *F#* с использованием библиотеки *Accord.net*. Для тестирования разработанных алгоритмов использовалась база данных *SMS Spam Collection*, состоящая из более чем 5000 сообщений, которые принадлежат к классу «спам» и «не-спам».

В основе наивного байесовского классификатора лежит теорема Байеса, которая позволяет определить вероятность события при условии, что произошло другое взаимозависимое событие:

$$P(c|d) = \frac{P(c|d) \cdot P(c)}{P(d)},$$

где $P(c \mid d)$ – вероятность того, что сообщение принадлежит классу «не-спам»; $P(c \mid d)$ – вероятность того, что сообщение принадлежит классу «спам»; $P(c)$ – безусловная вероятность встретить сообщение класса «спам» в обучающей выборке сообщений; $P(d)$ – безусловная вероятность встретить сообщение класса «не-спам» в обучающей выборке сообщений.

Алгоритм работы классификатора состоит в следующем. Исходный массив данных разбивается на слова или предложения (токенизируется). Затем происходит преобразование полученных данных в числовую форму с использованием модели *bag-of-words*.

Следующий шаг алгоритма – обучение функции классификации тренировочным набором данных. Рассмотрим конкретный пример. Дано три сообщения: x_1, x_2, x_3. Пусть два из них x_1 и x_2 относятся к классу «спам», а x_3 – к классу «не-спам». Из сообщений выделены слова $Y_1, Y_2, Y_3,$ которые являются ключевыми характеристиками для классификации сообщений. Результаты такого извлечения представлены в табл. 1.

Таблица 1

Количество слов и классы для сообщений

Сообщение	Класс	Количество слов		
		Y_1	Y_2	Y_3
x_1	спам	2	0	1
x_2	спам	0	0	3
x_3	не-спам	1	2	0

В табл. 2 представлено, как будет выглядеть общее количество слов, входящих в класс «спам и не-спам».

Таблица 2

Общее количество вхождений слов в классы

Слова	Класс «спам»	Класс «не-спам»
Y_1	2	1
Y_2	0	2
Y_3	3	0
Итого	5	3

При появлении в системе нового сообщения x_4 из него также будут выделены слова. При этом, если данные слова не встретятся в составленной ранее таблице, то они будут исключены из классификации, и будет подсчитано общее количество оставшихся слов. Вероятность того, что сообщение будет отнесно к одному из классов, равна произведению вероятностей вхождения найденных слов в этот класс. Необходимо рассчитать значения вероятностей для обоих классов, чтобы определить, к какому из них относится сообщение x_4.

После стадии обучения в классификатор передается тестовый массив данных и вычисляется результат. Точностью вычисления алгоритма является процент корректно определенных данных. Например, если в качестве входных данных подается 100 сообщений, которые гарантированно не являются спамом, то алгоритм должен показать точность, равную 100 %. Для тестирования разработанных библиотек также применялась модификация алгоритма, учитывающая наличие часто повторяемых слов в обоих группах.

Для практического тестирования классификатора брались первые 500 строк из базы данных *SMS Spam Collection* и на их основе строилась обучающая выборка (табл. 2). Непосредственное тестирование алгоритма проводилось для следующих 100 сообщений той же базы данных. Процесс был повторен 5 раз для двух видов алгоритма: прямого и модифицированного. По результатам тестирования программы немодифицированный алгоритм отфильтровал спам с точностью до 86 %. Точность модифицированного алгоритма составила 98 %.

Таблица 3

Результат работы алгоритмов

Номер тестовой выборки	**Точность оптимизированного алгоритма, %**	**Точность прямой версии алгоритма, %**	**Разница, %**
1	98,4	86,3	12,1
2	98,0	86,9	11,1
3	97,5	87,1	10,4
4	97,9	87,0	10,9
5	98,6	86,5	12,1
Итого (*среднее*)	98,08	86,6	11,32

Применение наивного байесового классификатора показало достаточно высокую эффективность и точность результатов. По сравнению с нейронными сетями, которые также применяются для решения задач подобного рода, он имеет следующие преимущества: меньший объем данных для обучения, простота реализации, скорость работы на небольших и средних объемах данных. К недостаткам данного подхода можно отнести требования к постоянному обучению системы и изменениям алгоритма, если необходимо добиться высокой точности анализа.

Литература

1. Леонтьева, Н. Н. К теории автоматического понимания текста. / Н. Н. Леонтьева. – М. : Изд-во Моск. ун-та, 2002. – Ч. 3. Семантический компонент. Локальный семантический анализ. – С. 49.