Бровкин К.Е., Раскатова М.В. Исследование методов машинного обучения для классификации неструктурированных текстовых документов // Международный журнал информационных технологий и энергоэффективности. – 2019. – Т. 4 № 2(12) с. 12–17

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 519

ИССЛЕДОВАНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ КЛАССИФИКАЦИИ НЕСТРУКТУРИРОВАННЫХ ТЕКСТОВЫХ ДОКУМЕНТОВ

Бровкин К.Е., Раскатова М.В.

Федеральное государственное бюджетное образовательное учреждение высшего образования «Национальный исследовательский университет «МЭИ», Россия (111250, г.Москва, ул. Красноказарменная, д. 14); e-mail: konstantinbrovkin@mail.ru

В статье приведены результаты исследования методов машинного обучения для автоматической многоклассовой классификации неструктурированных статей новостей, а также описаны этапы классификации: предварительная обработка данных, индексация методом мешка слов, уменьшение пространства признаков функцией TF-IDF.

Ключевые слова: классификация текстов, машинное обучение, предварительная обработка данных

RESEARCH OF MACHINE TRAINING METHODS FOR CLASSIFICATION OF UNSTRUCTURED TEXT DOCUMENTS

Brovkin K.E., Raskatova M.V.

National Research University “Moscow Power Engineering Institute”, Russia (111250, Moscow, Krasnokazarmennaya street, 14); e-mail: konstantinbrovkin@mail.ru

The paper present the result of the study of machine learning methods for automatic multi-class classification of unstructured news articles, and describes the stages of classification: preliminary data processing, method of bag-of-words indexing, and reduction of feature space by the TF-IDF function.

Keywords: text categorization, machine learning, data preprocessing

Проблема автоматической классификации текстовых документов с помощью методов машинного обучения заключается в том, что нельзя заранее определить наиболее эффективный метод для решения конкретной задачи. Поэтому, как правило, проводят несколько экспериментов с различными методами для выявления наиболее подходящего подхода под конкретную задачу. Также влияние на точность классификации будет оказывать и язык, на котором написаны документы [1].

Формально задачу классификации можно выразить так [2]:

Задано множество документов $D = \{d_1, \dots, d_{|D|}\}$ и множество различных категорий (классов) $C = \{c_1, \dots, c_{|c|}\}$. Неизвестная целевая функция $\Phi: D \cdot C \to \{0,1\}$ задается формулой:

$$\Phi(d_j, c_i) = \begin{cases} 0, & \text{если } d_j \notin c_i \\ 1, & \text{если } d_j \in c_i \end{cases} \quad (1)$$

Требуется построить такую функцию $\Phi': D \cdot C \to \{0,1\}$, называемую классификатором, которая будет максимально близка к функции Ф. Отдельно стоит отметить, что в данной статье рассматривается случай, когда один документ d_j относится только к одной категории c_i. Этот случай называют многоклассовой однозначной классификацией.

Построение классификаторов для исследования методов машинного обучения проводилось для классификации набора текстовых документов, состоящего из 30000 новостных статей на русском языке с сайта Lenta.ru, равномерно распределённых по 7 различным категориям статей, таких как:

- экономика;
- наука и техника;
- культура;
- спорт;
- политика;
- происшествия;
- авто.

Данные в наборе хорошо сбалансированы по группам (рисунок 1), то есть их количество в каждой группе почти одинаковое ($\pm 0{,}12\%$).

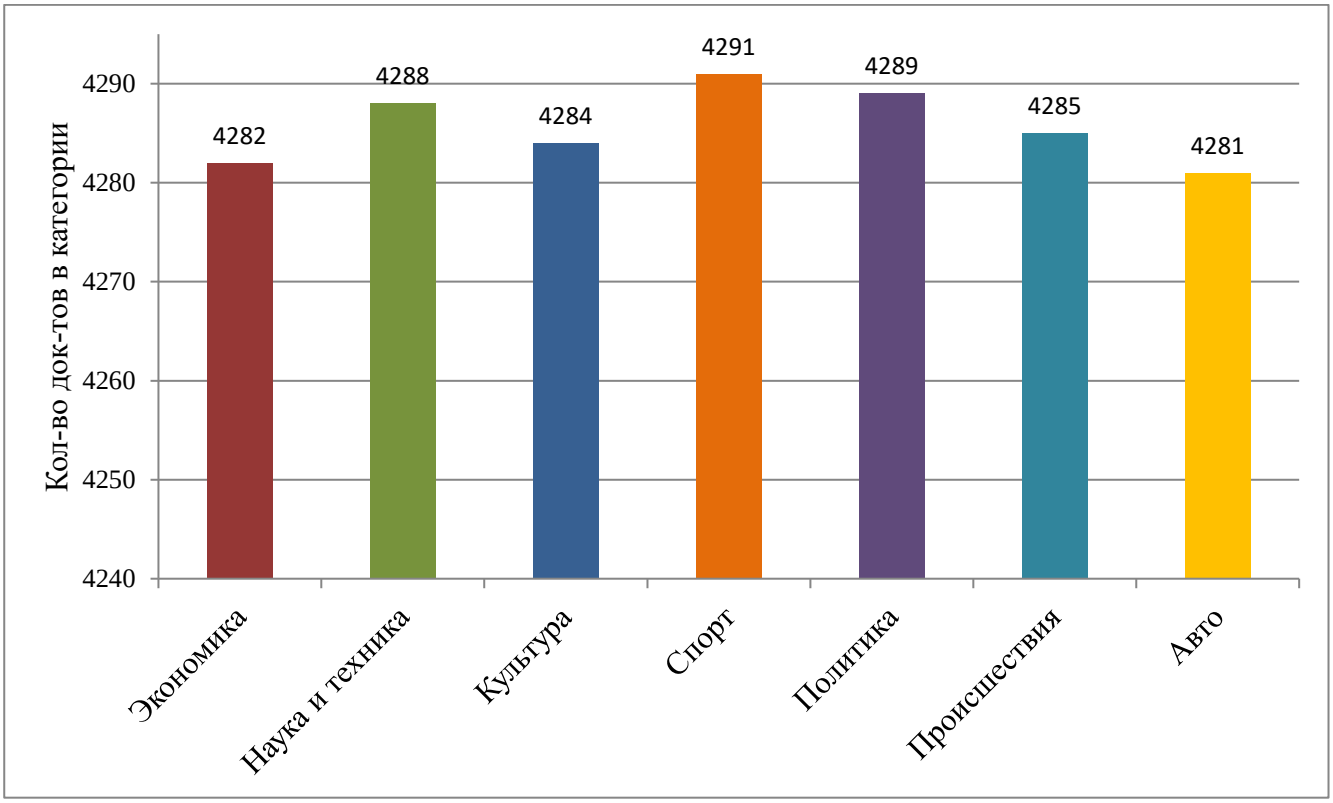


Рисунок 1 - Распределение текстовых документов по категориям

Решение задачи классификации состоит из следующих последовательных этапов [2]:

– предварительная обработка данных;
– индексация;
– уменьшение размерности пространства признаков;
– построение и обучение классификатора;
– оценка качества классификации.

Процесс предварительной обработки данных необходим для сокращения объёма документа и, как следствие, пространства признаков в массиве тестовых документов, что приводит к повышению точности классификатора. Предобработка текстовых документов заключается в следующем:

1. приведение текста к нижнему регистру;
2. удаление стоп-слов, то есть предлогов, причастий, междометий, частиц и других коротких слов, которые не несут смысловой нагрузки. Это позволяет сократить объём текста и увеличить его смысловую значимость;
3. удаление пунктуации;
4. лемматизация текста - процесс приведения слова к его нормальной форме, то есть выделение у заданного слова леммы. Позволяет избавиться от грамматической информации (падежи, род, число прилагательных, глагольные виды и времена, залоги причастий и так далее) в исходном тексте, сохранив только важную смысловую составляющую, что позволяет определять слова с одинаковыми леммами как один и тот же элемент, приводя слова с похожим значением к одному слову [3].

Для индексации использовался метод мешка слов (bag of words) [4]. Суть этого метода в том, что все текстовые файлы разбиваются на отдельные слова и подсчитывается количество вхождений каждого оригинального слова в отдельно взятый документ, и, наконец, каждому слову присваивается целочисленный идентификатор. Этот метод подходит для небольших наборов данных, таких как описанный выше набор новостных статей.

Вычислительная сложность различных методов классификации напрямую зависит от размерности пространства признаков. Поэтому для эффективной работы классификатора часто прибегают к сокращению числа используемых признаков [5].

Существуют несколько способов определения веса признаков документа. Наиболее используемый в связке с методом индексации мешка слов - вычисление функции TF-IDF [5]. Функция TF-IDF [6] рассчитывает в текстовом документе вес слов (терминов), который является статистической мерой, используемой для оценки того, насколько важно слово для документа в наборе документов. Важность слова увеличивается пропорционально тому, сколько раз оно появляется в документе, но смещается на частоту слова в наборе документов.

TF (term frequency - частота термина) - отношение числа вхождений некоторого слова к общему числу слов документа. Таким образом, оценивается важность слова t_i в пределах отдельного документа.

$$tf(t,d) = \frac{n_t}{\sum_k n_k}, \qquad (2)$$

где n_t есть число вхождений слова t в документ, а в знаменателе - общее число слов в данном документе.

IDF (inverse document frequency - обратная частота документа) - инверсия частоты, с которой некоторое слово встречается в документах коллекции. Учёт IDF уменьшает вес

широкоупотребительных слов. Для каждого уникального слова в пределах конкретной коллекции документов существует только одно значение IDF.

$$\mathrm{idf}(t, D) = \log \frac{|D|}{|\{d_i \in D \mid t \in d_i\}|}, \quad (3)$$

где $|D|$ - число документов в коллекции;

$|\{d_i \in D \mid t \in d_i\}|$ - число документов из коллекции D, в которых встречается t (когда $n_t \neq 0$).

Выбор основания логарифма в формуле не имеет значения, поскольку изменение основания приводит к изменению веса каждого слова на постоянный множитель, что не влияет на соотношение весов.

Таким образом, мера TF-IDF является произведением двух сомножителей:

$$\mathrm{tfidf}(t, d, D) = tf(t, d) * idf(t, D) \quad (4)$$

Большой вес в TF-IDF получат слова с высокой частотой в пределах конкретного документа и с низкой частотой употреблений в других документах.

Далее используя один из методов машинного обучения и набор признаков классификации, определённый на предыдущих этапах, осуществлялось обучение классификаторов. Методы машинного обучения, результаты работы которых, исследуются в данной статье:

- полиномиальный наивный Байес;
- Бернулли наивный Байес;
- метод k-ближайших соседей;
- метод деревьев решений;
- метод опорных векторов;
- нейронная сеть.

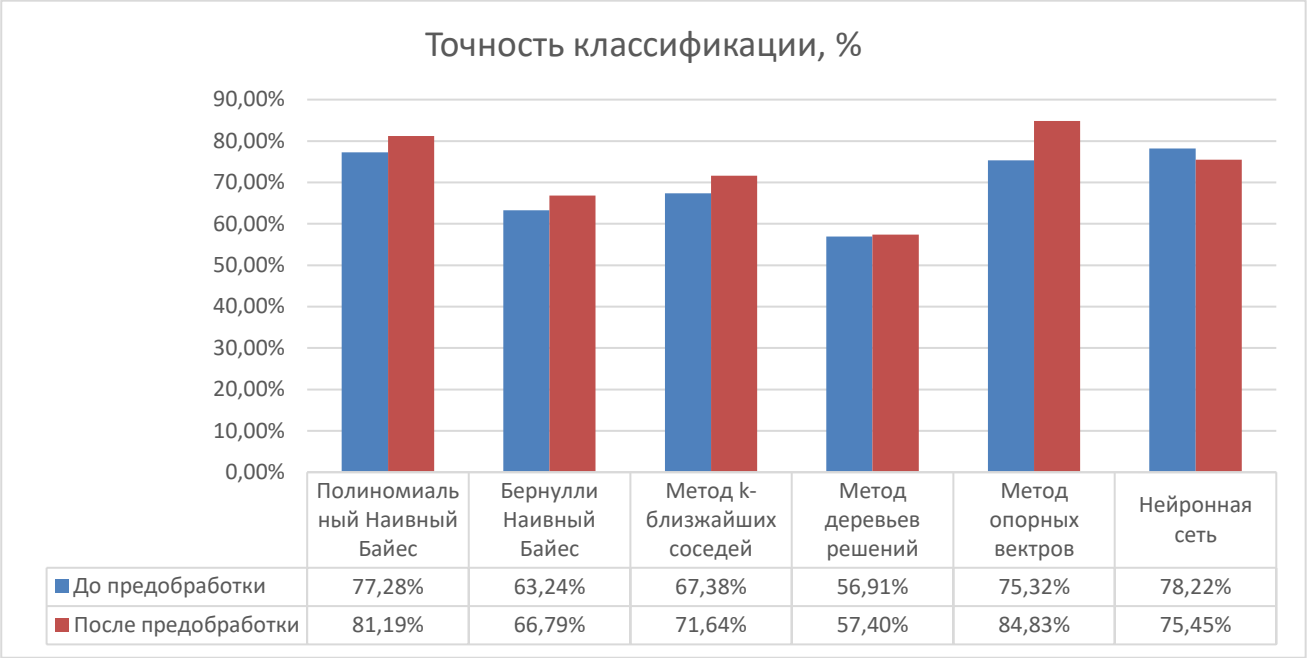
Для обучения и тестирования классификаторов всё множество документов случайным образом разделялось на два непересекающихся подмножества:

- набор данных для обучения (обучающая выборка);
- набор данных для проверки (тестирующая выборка).

На обучающем множестве строился классификатор, и определялись значения его параметров, при которых классификатор выдавал лучший результат. На тестовом наборе происходило вычисление эффективности классификатора.

Для обучения классификатора использовались 80% документов из набора новостных статей, то есть около 24000 документов. Для тестирования, соответственно, использовались оставшиеся 20% или около 6000 документов.

В итоге получены следующие экспериментальные данные (рисунок 2 и 3), на которых показаны точность и время выполнения рассматриваемых методов машинного обучения для данных текстовых документов прошедших предварительную обработку и нет.



	Полиномиальный Наивный Байес	Бернулли Наивный Байес	Метод k-ближайших соседей	Метод деревьев решений	Метод опорных вектров	Нейронная сеть
До предобработки	77,28%	63,24%	67,38%	56,91%	75,32%	78,22%
После предобработки	81,19%	66,79%	71,64%	57,40%	84,83%	75,45%

Рисунок 2 - Сравнение точности классификации при использовании различных методов машинного обучения

	Полиномиальный Наивный Байес	Бернулли Наивный Байес	Метод k-ближайших соседей	Метод деревьев решений	Метод опорных вектров	Нейронная сеть
До предобработки	24,29	22,63	25,66	39,94	26,72	53,48
После предобработки	15,61	12,76	18,12	16,38	15,64	26,21

Рисунок 3 - Сравнение времени обучения классификаторов при использовании различных методов машинного обучения

На приведённых рисунках (рисунок 2 и 3) показано, что благодаря предварительной обработке документов удалось увеличить точность, а также сократить время обучения почти для всех классификаторов. Исключение составил классификатор с методом Нейронной сети, который потерял в точности 2,77% после предобработки.

Худшие результаты показал метод деревьев решений, как до, так и после операции предобработки. Его точность составила 56,91% и 57,40% соответственно.

До предобработки себя наилучшим образом с точки зрения точности показал метод Нейронной сети (78,22%), но уже после предобработки лучшая точность была достигнута методом опорных векторов (84,83%), именно этот метод прибавил в точности больше всех, его точность увеличилась на 9,51%.

Наиболее быстрым по времени обучения классификатора среди всех методов машинного обучения оказался метод Бернулли Наивный Байес: 22,63 и 12,76 секунд до и после предобработки. Самым медленным метод Нейронной сети: 53,48 и 26,21 секунда.

Лучшие результаты, как по точности, так и по времени обучения после предобработки показали следующие два классификатора: с помощью метода Полиномиального Наивного Байеса (81,19% точности; 15,61 сек. времени обучения) и метода Опорных векторов (84,83% точности; 15,64 сек. времени обучения).

Список литературы

1. Батура Т.В. Методы автоматической классификации текстов // Программные продукты и системы. - 2017. Т. 30, № 1. - С. 85-99.
2. Fabrizio Sebastiani: Machine Learning in Automated Text Categorization, 2002.
3. Natural Language Processing with Python/Bird, Steven, Edward Loper and Ewan Klein - O’Reilly Media Inc. - 2009. - С. 107-108.
4. Jason Brownlee: Deep Learning for Natural Language Processing, 2017.
5. Zhang X. Character-level Convolutional Networks for Text Classification/Zhang X., Zhao J., LeCun Y. - Montreal: 2015.
6. Jones K.S. A statistical interpretation of term specificity and its application in retrieval // Journal of Documentation : журнал. - MCB University: MCB University Press. - 2004. Т. 60, №5 - С. 493-502.

References

1. Batura T.V. Automatic text classification methods. Programmnye produkty i sistemy [Software & Systems]. 2017, vol. 30, no. 1, pp. 85-99 (in Russ.)
2. Fabrizio Sebastiani: Machine Learning in Automated Text Categorization, 2002.
3. Natural Language Processing with Python/Bird, Steven, Edward Loper and Ewan Klein - O’Reilly Media Inc. - 2009. - p. 107-108.
4. Jason Brownlee: Deep Learning for Natural Language Processing, 2017.
5. Zhang X. Character-level Convolutional Networks for Text Classification/Zhang X., Zhao J., LeCun Y. - Montreal: 2015.
6. Jones K.S. A statistical interpretation of term specificity and its application in retrieval // Journal of Documentation : журнал. - MCB University: MCB University Press, 2004. - Vol. 60, no. 5. - p. 493-502.