DOI: 10.18137/RNU.V9187.21.04.P.175 УДК 004.912

Э.А. Челышев, Ш.А. Оцоков, М.В. Раскатова

АВТОМАТИЧЕСКАЯ РУБРИКАЦИЯ ТЕКСТОВ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ

Рассмотрено решение задачи автоматической рубрикации русскоязычных текстов с использованием алгоритмов машинного обучения на примере корпуса новостных статей как задачи классификации на некоторое число непересекающихся классов. Показан алгоритм подготовки текстовых данных для классификации и его практическая реализация на языке программирования Python. Проведен анализ существующих методов нормализации токенов. Представлены результаты проведенного исследования по построению ряда классификаторов для решения задачи классификации русскоязычных текстов. Обобщающая способность классификаторов оценена по ряду метрик.

Ключевые слова: классификация, токенизация, нормализация, стоп-слово, метрика.

E.A. Chelyshev, Sh.A. Otsokov, M.V. Raskatova

AUTOMATIC TEXT RUBRICATION USING MACHINE LEARNING ALGORITHMS

The article presents the solution of the problem of automatic rubrication of Russian-language texts using machine learning algorithms on the example of a corpus of news articles. This problem is considered as a classification problem for a certain number of disjoint classes. The algorithm of preparing text data for classification and its practical implementation in the Python programming language is presented. The analysis of existing methods of token normalization is carried out. The results of the research on the construction of a number of classifiers for solving the problem of classification of Russian-language texts are presented. The generalizing ability of classifiers is estimated by a number of metrics.

Keywords: classification, tokenization, normalization, stopword, metric.

Вводные замечания

Всеобъемлющее развитие информационных технологий и их внедрение в разнообразные сферы деятельности приводят к росту производимых человечеством и хранимых в различной форме данных. Так, например, прогнозируется, что к 2025 году общемировой объем данных составит более 175 зеттабайтов [13].

Столь стремительный рост накопленных данных приводит к тому, что человеку становится все сложнее ориентироваться в информационном поле, из-за чего возрастает потребность в использовании средств автоматизированной обработки информации, которая может осуществляться с применением алгоритмов машинного обучения.

В работе рассматривается задача автоматической рубрикации текстов на естественном языке с точки зрения машинного обучения по прецедентам, в терминологии которого она трактуется как задача классификации на несколько непересекающихся классов, при этом каждая отдельная рубрика рассматривается как отдельный класс [7]. В рамках данной задачи также представлен алгоритм подготовки текстовых данных.

Челышев Эдуард Артурович
магистрант Московского энергетического института (национальный исследовательский университет), Москва. Сфера научных интересов: машинное обучение по прецедентам, искусственные нейронные сети, машинная обработка текстов на естественном языке, C++. Автор 1 опубликованной научной работы.
Электронный адрес: chel.ed@yandex.ru

Оцоков Шамиль Алиевич
доктор технических наук, доцент кафедры вычислительных машин, систем и сетей. Московский энергетический институт (национальный исследовательский университет), Москва. Сфера научных интересов: машинное обучение, машинная арифметика. Автор более 30 опубликованных научных работ.
Электронный адрес: Shamil24@mail.ru

Раскатова Марина Викторовна
кандидат технических наук, доцент кафедры вычислительных машин, систем и сетей. Московский энергетический институт (национальный исследовательский университет), Москва; доцент кафедры информационных технологий и естественнонаучных дисциплин. Российский новый университет, Москва. Сфера научных интересов: разработка программного обеспечения, информационные системы. Автор более 40 опубликованных научных работ.
Электронный адрес: marina@raskatova.ru

Алгоритм подготовки данных в задаче классификации текстов

Пусть имеется некоторый набор данных, каждый объект которого содержит текстовые данные на естественном языке и метку принадлежности к определенному классу. Рассмотрим текстовые данные отдельного объекта вышеуказанного набора данных. Для построения классификатораданные необходимо подготовить, преобразовав их в числовой вид. Такая подготовка включает в себя следующие этапы:

- удаление нерелевантных символов и приведение символов к общему регистру;
- токенизация;
- нормализация;
- удаление стоп-слов;
- векторизация.

Первый этап подготовки, а именно *удаление нерелевантных символов и приведение символов к общему регистру*, позволяет исключить из текста шумовую информацию. Нерелевантными символами могут являться цифры, знаки препинания, прочие небуквенные символы.

На этапе *токенизации* проводится разбиение текстовых данных на отдельные токены, то есть отдельные текстовые единицы [3]. В рассматриваемом алгоритме в качестве токенов выступают отдельные слова, а порядок следования токенов не играет роли, поэтому без ограничения общности можно считать, что на данном этапе формируется исходное множество T_0, каждый элемент которого является отдельным токеном. Отметим также, что в контексте данной статьи понятия «слово» и «токен» в целом взаимозаменяемы.

Нормализация токенов позволяет привести различные формы слова к одному токену. Это полезно, так как одному и тому же значению соответствуют сразу несколько форм одного слова, которые в дальнейшем при машинной обработке могут восприниматься как различные токены [4]. На этапе нормализации каждый токен множества T_0 преобразуется к своей начальной форме [8]. Измененное таким образом множество токенов обозначим T_1. Более подробно задача нормализации токенов и подходы к ее решению изложены в статье далее.

Стоп-слова – это часто встречающиеся в текстах слова, которые играют большую роль в обеспечении связности предложения, однако при машинной обработке естественного языка являются шумом. К ним можно отнести частицы, предлоги, союзы и др. [4]. Обозначим множество стоп-слов S. Тогда множество токенов с удаленными из него стоп-словами обозначим $T = T_1 \setminus S$.

Векторизация – процесс, в результате которого каждому токену ставится в соответствие его векторное представление [2], то есть вектор v некоторого мерного векторного пространства $\mathbb{R}^n$, иными словами, задается отображение

$$f : T \to V, \tag{1}$$

где $V = \{v_1, v_2, ..., v_m\}$ – множество векторов-образов размерности n.

В конечном итоге текстовым данным каждого объекта ставится в соответствие вектор, определяемый как среднее арифметическое векторов, соответствующих отдельным токенам, полученным из текстовых данных этого объекта, то есть

$$v = \frac{\sum_{t \in T} f(t)}{|T|}, \tag{2}$$

где $|T|$ – мощность множества T.

Методы нормализации токенов

Для решения задачи нормализации токенов используются два метода –стемминг и лемматизация, каждый из которых имеет свои достоинства и недостатки [1].

Стемминг (англ. stemming) – метод нормализации токенов, в ходе которого от слов отсекаются префиксы, суффиксы и окончания, в результате чего выделяется основа слова [9]. Системы, осуществляющие стемминг, называются стеммерами. Наибольшее распространение получили алгоритмические стеммеры. Их недостатком является тот факт, что они могут совершать ошибки, из-за которых результат стемминга может оказаться отчасти некорректным [9].

Лемматизация – метод приведения слов к начальной форме, при котором проводится морфологический анализ слова, по результатам которого оно отождествляется с некоторой лексемой, которая называется леммой. Лемматизация гарантирует, что различные формы слова будут приведены к одной, начальной, лексеме. Однако лемматизация более требовательна к вычислительным ресурсам, нежели алгоритмический стемминг [8].

Практическая реализация алгоритма подготовки данных

Рассмотрим практическую реализацию приведенного выше алгоритма. Для подготовки текстовых данных был разработан программный модуль на языке программирования Python с использованием ряда его библиотек.

Удаление нерелевантных символов было проведено с использованием регулярных выражений. Токенизация текстов осуществлялась с использованием встроенного метода *word_tokenize* стандартной библиотеки *NLTK* [12].

В рамках данной работы для решения задачи нормализации токенов была выбран метод лемматизации, которая осуществлялась с использованием русскоязычного морфологического анализатора, реализованного в библиотеке *pymorphy2* [5; 10].

Для решения задачи удаления стоп-слов в программном модуле реализована функция *delete_stop_words.* Получая на вход список токенов, она возвращает список, очищенный от стоп-слов, поочередно сравнивая каждый поступивший на вход токен с токенами в списке стоп-слов. Такой список для русского языка, состоящий из 151 слова, имеет в своем составе библиотека NLTK.

Для получения векторного представления слов используются различные методы векторизации, которые могут как сохранять семантические, то есть смысловые, отношения слов, так и игнорировать их. В данной работе была использована предобученная модель векторизации *FastText,* сохраняющая семантические отношения, доступная для свободного использования на интернет-ресурсе [14].

С использованием разработанного программного модуля был подготовлен для задачи классификации текстов корпус статей новостного интернет-портала *LENTA.RU* [11], из которого предварительно были выделены статьи, относящиеся к девяти отдельным рубрикам.

Пример текстовых данных корпуса, прошедших все этапы подготовки (кроме векторизации), представлен на Рисунке 1.

	text	topic
0	[январь, год, всё, телеканал, оплачивать, услу...	8
1	[германский, автопромышленный, концерн, volksw...	8
2	[нераспределённый, прибыль, оао, тюменнефтегаз...	8
3	[крупный, телекоммуникационный, компания, сша,...	8
4	[оао, газ, нижегородский, банк, сбербанк, росс...	8

Рисунок 1. Фрагмент подготовленных с использованием программного модуля данных

Проведение экспериментов по построению классификаторов

Были построены четыре классификатора: на основе *наивного байесовского классификатора* (далее – НБК), *логистической регрессии*(далее – ЛР), *случайного леса решающих деревьев* (далее – СЛРД) и *искусственной нейронной сети* (далее – ИНС), архитектура которой подробно рассмотрена в [6]. Классификаторы были обучены с использованием описанного выше подготовленного корпуса статей, который предварительно был разделен на обучающую и тестовую выборки.

Гиперпараметры классификаторов, дающие наилучшую обобщающую способность, определялись с использованием алгоритма решетчатого поиска. Оценка обобщающей

способности классификаторов производилась с использованием метрик *precision* (точность) и *recall* (полнота), определяемых формулами (3) и (4) соответственно.

$$precision = \frac{G_p^+}{G_p^+ + G_p^-}; \quad (3)$$

$$recall = \frac{G_p^+}{G_p^+ + G_n^-}, \quad (4)$$

где G_p^+ – число верно классифицированных объектов; G_p^- – число объектов, которые были неверно отнесены к текущему классу; G_n^- – число объектов, которые неверно были отнесены к другим классам.

Также была использована F_1 -мера, определяемая формулой

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall}. \quad (5)$$

В Таблице приведены средние взвешенные по классам значения метрик классификации для каждого классификатора.

На Рисунке 2 представлена гистрограмма значений среднего взвешенного по классам значения F_1 -меры.

Таблица

Сводная таблица значений метрик классификации на тестовой выборке для построенных классификаторов

Классификатор	**Среднее взвешенное по классам значение точности**	**Среднее взвешенное по классам значение полноты**	**Среднее взвешенное по классам значение F_1-меры**
НБК	0,81459	0,79775	0,75367
ЛР	0,90216	0,90236	0,90222
СЛРД	0,88318	0,88310	0,88221
ИНС	0,9253	0,9250	0,9251

Заключение

Представлен, реализован и опробован алгоритм подготовки текстовых данных для задачи классификации. На подготовленных с его помощью данных был обучен ряд моделей машинного обучения, которые показали достаточно высокие значения обобщающей способности; можно сделать вывод о правильности предложенного алгоритма.

Рассматривая результаты проведенных экспериментов по построению классификаторов, можно заключить, что ИНС показала наивысшие значения метрик классификации. Классификаторы на основе логистической регрессии и случайного леса решающих деревьев показали результаты чуть хуже. Наименьшее значение продемонстрировал наивный байесовский классификатор.

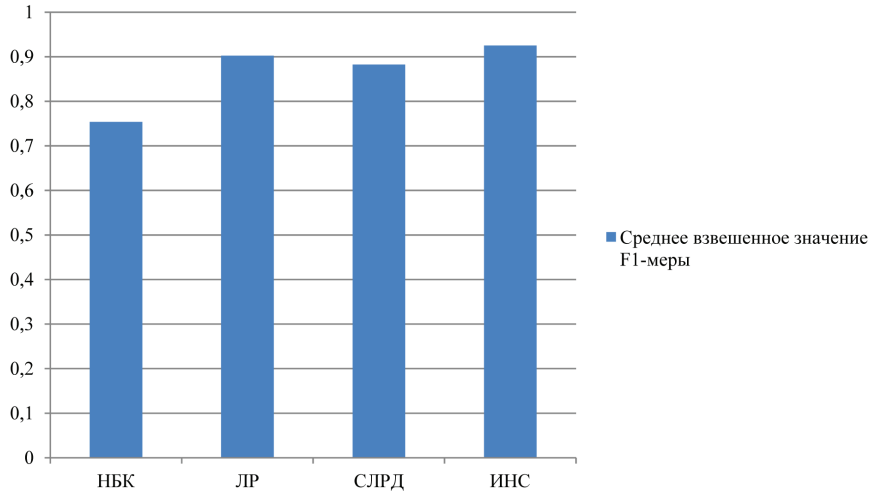

Рисунок 2. Сравнительная диаграмма значений F_1-меры для построенных моделей классификации

Литература

1. *Вершинин Е.В., Тимченко Д.К.* Исследование применения стемминга и лемматизации при разработке систем адаптивного перевода текста // Наука. Исследования. Практика: сборник избранных статей по материалам Международной научной конференции (Санкт-Петербург, 25 декабря 2019 г.). Санкт-Петербург: Гуманитарный национальный исследовательский институт «Нацразвитие», 2020. 310 с. ISBN 978-5-6043877-4-0.

2. *Жеребцова, Ю.А., Чижик А.В.* Сравнение моделей векторного представления текстов в задаче создания чат-бота. // Вестник НГУ. 2020. Т. 18.

3. *Захаров В.П., Богданова С.Ю.* Корпусная лингвистика: учебник. Иркутск: ИГЛУ. 2011.

4. *Мартынов В. А., Плотникова Н.П.* Нормализация и фильтрация текста для задачи кластеризации // XLVIII Огарёвские чтения: материалы научной конференции. В 3 ч. (Саранск, 06–13 декабря 2019 г.). Саранск: Национальный исследовательский Мордовский государственный университет имени Н.П. Огарёва, 2020. С. 448–452.

5. Морфологический анализатор pymorphy 2 [Электронный ресурс]. URL: https://pymorphy2.readthedocs.io/en/stable/ (дата обращения: 03.05.2021).

6. *Челышев Э.А., Оцоков Ш.А., Раскатова М.В.* Разработка информационной системы для автоматической рубрикации новостных текстов // Международный журнал информационных технологий и энергоэффективности. 2021. Т. 6, № 3 (21). С. 11–17.

7. *Шаграев А.Г.* Модификация, разработка и реализация методов классификации новостных текстов: дис. … канд. техн. наук. М.: МЭИ, 2014. 108 с.

8. *Якиль К.А., Рязанова Н.Ю.* Фильтрация SMS-спама // Автоматизация. Современные технологии. 2016. № 9. С. 19–24.

9. *Яцко В.А.* Алгоритмы и программы автоматической обработки текста // Вестник Иркутского государственного лингвистического университета. 2012. № 1 (17). С. 150–161.

10. *Korobov M.* (2015) Morphological Analyzer and Generator for Russian and Ukrainian Languages. Analysis of Images, Social Networks and Texts, pp. 320–332.
11. Kaggle: Your Home for Data Science. Available at: https://www.kaggle.com/yutkin/corpus-of-russian-news-articles-from-lenta (date of the application: 08.02.2021).
12. NLTK 3.6.2 documentation. Available at: https://www.nltk.org/ (date of the application: 14.04.2021).
13. *Reinsel D., Gantz J., Rydning J.* (2018) The Digitalization of the World, 28 p. Available at: https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf (date of the application: 11.03.2021).
14. Rus Vectores: semantic models for the Russian language. Available at: https://rusvectores.org/ru/ (date of the application: 14.02.2021).

References

1. Vershinin E.V., Timchenko D.K. (2020) *Issledovanie primeneniya stemming i lemmatizacii pri razrabotke system adaptivnogo perevoda teksta* [Investigation of the application of stemming and lemmatization in the development of adaptive text translation systems]. *Nauka. Issledovaniya. Praktika, Sbornik izbrannyh statej po materialam Mezhdunarodnoj nauchnoj konferencii, Sankt-Peterburg, 25 dekabrya 2019 g.* [The science. Research. Practice, Collection of selected articles based on the materials of the International Scientific Conference, St. Petersburg, December 25, 2019]. St. Petersburg, Gumanitarnyj nacional'nyj issledovatel'skij institut «NACRAZVITIE», 310 p. (in Russian). ISBN 978-5-6043877-4-0
2. Zherebcova, Yu.A., Chizhik A.V. (2020) *Sravnenie modelej vektornogo predstavleniya tekstov v zadache sozdaniya chat-bota* [Comparison of models of vector representation of texts in the task of creating a chat bot]. *Vestnik NGU*, vol. 18.
3. Zaharov V.P., Bogdanova S.Yu. (2011) *Korpusnaya lingvistika* [Corpus linguistics]. Irkutsk, IGLU Publishing (in Russian).
4. Martynov V.A., Plotnikova N.P. (2020) *Normalizaciya i fil'traciya teksta dlya zadachi klasterizacii* [Normalizing and Filtering Text for a Clustering Problem]. *XLVIII Ogaryovskie chteniya: Materialy nauchnoj konferencii, Saransk, 06–13 dekabrya 2019 g.* [HLVII Ogarev readings: Materials of a scientific conference, Saransk, December 06-13, 2019]. Saransk, Nacional'nyj issledovatel'skij Mordovskij gosudarstvennyj universitet imeni N.P. Ogaryova, pp. 448–452 (in Russian).
5. *Morfologicheskij analizator pymorphy 2* [Morphological analyzer of pomorph 2]. Available at: https://pymorphy2.readthedocs.io/en/stable/ (date of the application:03.05.2021).
6. Chelyshev E. A., Ocokov Sh.A., Raskatova M.V. (2021) *Razrabotka informacionnoj sistemy dlya avtomaticheskoj rubrikacii novostnyh tekstov* [Development of an information system for automatic rubrication of news texts]. *Mezhdunarodnyj zhurnal informacionnyh tekhnologij i energoeffektivnosti*, vol. 6, no. 3 (21), pp. 11–17 (in Russian).
7. Shagraev A.G. (2014) *Modifikaciya, razrabotka i realizaciya metodov klassifikacii novostnyh tekstov* [Modification, development and implementation of methods for the classification of news texts]: PhD thesis. Moscow, MEI, 108 p. (in Russian).

8. Yakil' K.A., Ryazanova N.Yu. (2016) *Fil'traciya SMS-spama* [Filtering SMS spam]. *Avtomatizaciya. Sovremennye tekhnologii*, no. 9, pp. 19–24 (in Russian).
9. Yacko V.A. (2012) *Algoritmy i programmy avtomaticheskoj obrabotki teksta* [Algorithms and programs for automatic word processing]. Vestnik Irkutskogo gosudarstvennogo lingvisticheskogo universiteta, no. 1 (17), pp. 150–161 (in Russian).
10. Korobov M. (2015) Morphological Analyzer and Generator for Russian and Ukrainian Languages. Analysis of Images, Social Networks and Texts, pp. 320–332.
11. Kaggle: Your Home for Data Science. Available at: https://www.kaggle.com/yutkin/corpus-of-russian-news-articles-from-lenta (date of the application: 08.02.2021).
12. NLTK 3.6.2 documentation. Available at: https://www.nltk.org/ (date of the application: 14.04.2021).
13. Reinsel D., Gantz J., Rydning J. (2018) The Digitalization of the World, 28 p. Available at: https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf (date of the application: 11.03.2021).
14. Rus Vectores: semantic models for the Russian language. Available at: https://rusvectores.org/ru/ (date of the application: 14.02.2021).