УДК 004.9
DOI 10.34822/1999-7604-2021-1-12-19

РЕШЕНИЕ ЗАДАЧИ КЛАССИФИКАЦИИ ДОКУМЕНТОВ ВУЗА НА ОСНОВЕ МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА

А. Л. Ткаченко
Сибирский государственный автомобильно-дорожный университет, Омск, Россия
E-mail: tanaleo@mail.ru

Рассмотрена задача классификации текстовых документов высшего учебного заведения с помощью следующих методов машинного обучения: метода Байеса, метода *k*-ближайших соседей, метода дерева решений, метода опорных векторов. Исследование проведено с использованием языка программирования Python, в качестве исходных данных был использован набор документов Сибирского государственного автомобильно-дорожного университета. Выполнена предварительная обработка документов с отнесением их к одному из четырех классов: приказ, распоряжение, письмо, извещение о вакантном месте. Проведен сравнительный анализ результатов классификации каждого метода машинного обучения по таким метрикам, как правильность алгоритма классификации, точность, полнота, *F*-мера, время работы алгоритма. Разработаны рекомендации по применению ансамбля методов, показавших наилучшие результаты, для оптимизации работы классификатора документов.

Ключевые слова: классификация текстов, обработка текстов, машинное обучение, качество классификации.

SOLVING THE PROBLEM OF UNIVERSITY DOCUMENTS CLASSIFICATION BASED ON INTELLECTUAL ANALYSIS METHODS

A. L. Tkachenko
Siberian State Automobile and Highway University, Omsk, Russia
E-mail: tanaleo@mail.ru

The article considers a problem of classification of text documents of a higher educational institution using machine learning methods such as the Bayes method, the *k*-nearest neighbors algorithm, the decision tree method, and the support-vector machines. The study is conducted on a set of documents of the Siberian State Automobile and Highway University, Omsk, using the Python programming language. Preliminary processing of documents for the study is carried out. All studied documents are divided into four classes namely the order, the instruction, the letter, and the notice of a vacant position. The process of classification of documents of a higher educational institution is presented. A comparative analysis of the classification results of each machine learning method is carried out on such metrics as the correctness of the classification algorithm, accuracy, completeness, *F*-measure, and the running time of the algorithm. As a result of the research, the author gives recommendations on the application of the considered methods for the classification of university documents. The author suggests using a combination of methods to optimize the operation of the documents classifier.

Keywords: text classification, text processing, machine learning, classification quality.

Введение

Внедрение системы электронного документооборота (СЭД) для организации и управления – актуальная задача для высших учебных заведений. В связи с большими возможностями СЭД, а также активным развитием технологий роботизации процессов RPA (robotic process automation) появилась перспектива построения СЭД, реализующих более сложные функции управления. Поэтому важными становятся разработка и внедрение усовершенствованных СЭД для максимальной автоматизации ручного труда и получения большего экономического эффекта.

Одной из задач, которую позволяет решить использование СЭД и RPA, является задача автоматической классификации документов [1]. Для уменьшения времени обработки генерируемой внутренней и внешней документации, а также сокращения количества ошибок при обработке информации применяются различные методы машинного обучения [2].

В работе рассмотрены методы машинного обучения, позволяющие классифицировать документы вуза по содержащейся в них информации. Также проведен сравнительный анализ результатов классификации документов каждым методом машинного обучения.

Материал и методы

В качестве исходных данных для сравнения методов автоматической классификации документов использован набор из 291 документа отдела организации практики и содействия трудоустройству выпускников Сибирского государственного автомобильно-дорожного университета (СибАДИ). Все рассматриваемые документы принадлежат к одной из четырех категорий: приказ, распоряжение, письмо, извещение о вакантном месте.

Документы подвергнуты предварительной обработке: все символы текстов документов переведены в нижний регистр, тексты разбиты на слова, из текстов удалены шумовые слова (слова, не обладающие семантической нагрузкой), оставшиеся слова текстов приведены к нормальной форме.

Все вычисления выполнялись на языке Python с использованием облачной платформы Google Colaboratory, включающей библиотеки машинного обучения и анализа данных, а также графические процессоры для визуализации полученных результатов.

Первоначальный набор данных содержал 10 874 уникальных словоформ. Для разбиения набора данных на отдельные слова использовался метод tokenize.RegexpTokenizer библиотеки nltk, разбивший текст на слова с помощью регулярного выражения \w+. Очистка текстов проводилась с использованием корпуса русскоязычных шумовых слов corpus. stopwords.words('russian') библиотеки nltk. Для приведения слов к нормальной форме использовался метод MorphAnalyzer.parse библиотеки pymorphy2. После предварительной обработки количество уникальных словоформ в наборе данных сократилось до 5 950. Фрагмент программы исследования представлен на рис. 1.

```
input_text = input_text.lower()

stop_words = stopwords.words('russian')

tokenizer = RegexpTokenizer(r'\w+')
tokens = tokenizer.tokenize(input_text)
tokens = [x for x in tokens if not x in stop_words]

morph = pymorphy2.MorphAnalyzer()
tokens_parse = [morph.parse(x)[0].normal_form for x in tokens]
tokens_parse = ' '.join(tokens_parse)
```

Рис. 1. Фрагмент листинга программы предварительной обработки документов вуза
Примечание: скриншот автора.

В табл. 1 представлен фрагмент результата предварительной обработки документов (номера документов, их наименования и частичное содержание). Следующий шаг после предварительной обработки – построение числовой модели текстов документов.

Таблица 1

Фрагмент данных после предварительной обработки документов

№	Наименование документа	Частичное содержание документа
1	Приказ об изменении штатного расписания	изменение, штатный, расписание, текст, приказ, связь, производственный, необходимость, приказывать, внести, изменение, …
2	Распоряжение о проведении дня открытых дверей	текст, распоряжение, связь, проведение, профориентационный, мероприятие, день, открытый, дверь, обеспечить, разместить, …

Окончание табл. 1

№	Наименование документа	Частичное содержание документа
3	Письмо	бюро, инженерный, проектирование, выражать, свой, почтение, намерение, сотрудничать, ваш, учебный, заведение, знать, сильный, …
4	Извещение о вакантном месте младшего тестировщика	младший, тестировщик, компания, который, делать, собственный, продукт, автоматизировать, сбор, хранение, анализ, …
…	…	…
288	Приказ об организации мероприятий по профилактике новой коронавирусной инфекции	текст, приказ, цель, предотвращение, распространение, новый, коронавирусный, инфекция, обеспечение, соблюдение, …
289	Распоряжение об организации проведения опроса студентов	текст, распоряжение, цель, оценка, удовлетворённость, студент, обучение, выбрать, направление, обучение, выявление, фактор, …
290	Письмо	запрос, просить, подтвердить, опровергнуть, подлинность, диплом, квалификация, инженер, строитель, специальность, автомобильный, …
291	Извещение о вакантном месте инженера	работа, требоваться, полный, занятость, полный, день, обязанность, организовывать, выполнять, работа, наладка, испытание, …

Примечание: составлено автором на основании данных, полученных в исследовании.

Постановка задачи

В рамках данного исследования рассмотрена задача классификации документов, которую в формальном виде можно представить следующим образом: пусть существует описание документа $d \in D$, где $D = \{d_1, d_2, ..., d_n\}$ – векторное пространство документов, а также фиксированный набор классов $C = \{c_1, c_2, ..., c_m\}$. Из обучающей выборки (множества документов с заранее известными классами) $D^{train} = \{< d, c > | < d, c > \in D \times C\}$ с помощью метода обучения F необходимо получить классифицирующую функцию $F(D^{train}) = g$, которая отображает документы в классы $g : D \to C$ [3]. В решаемой в рамках данного исследования задаче классами являются типы документов отдела организации практики и содействия трудоустройству выпускников СибАДИ (приказ, распоряжение, письмо, извещение о вакантном месте).

Описание процесса классификации документов

На рис. 2 представлен алгоритм классификации документов вуза и выделен цикл предварительной обработки документов, после которого идет построение числовой модели документов *bag of words*, которая представляет документ в виде вектора, состоящего из входящих в документ слов [4].

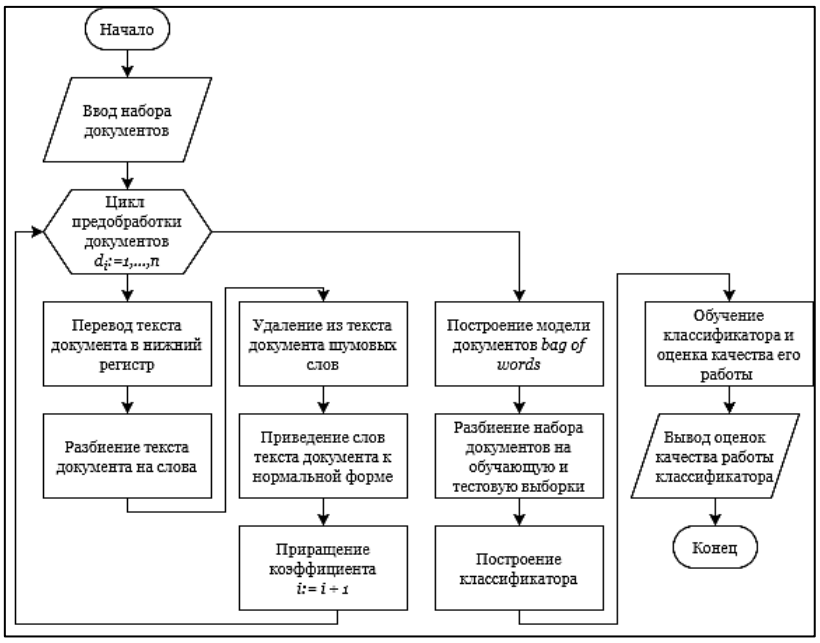

Рис. 2. Блок-схема алгоритма классификации документов вуза
Примечание: составлено автором.

С каждым словом сопоставляется его вес – количественная оценка значимости слова в документе. Для ее вычисления используется метод TF-IDF (term frequency- inverse document frequency), согласно которому слова, наиболее распространенные в конкретном документе и менее распространенные в остальных, получают больший вес [5]:

$$W_{t,d} = TF \cdot IDF, \tag{1}$$

где $TF = n_{t,d} / n_d$ – частота, с которой слово встречается в пределах одного документа, $n_{t,d}$ – число вхождений слова t в документ d; n_d – количество всех слов в документе d; $IDF = \log(|D|/D_t)$ – обратная частота документа, $|D|$ – количество документов в наборе данных; D_t – число документов из набора данных, в которых встречается слово t.

Для построения числовой модели документов использовался метод TfidfVectorizer библиотеки scikit-learn, который для вычисления весов слов использует TF-IDF. Метод был обучен на корпусе документов с помощью метода fit. Числовая модель документов была построена с помощью метода transform. Фрагмент программы исследования представлен на рис. 3.

```
vectorizer = TfidfVectorizer()
vectorizer.fit(data['text'])

X_train = vectorizer.transform(D['train']['x'])
X_test  = vectorizer.transform(D['test']['x'])
```

Рис. 3. Фрагмент листинга программы построения числовой модели документов
Примечание: скриншот автора.

После построения числовой модели текстов набор данных разбивается на обучающую и тестовую выборки, а затем происходит построение классификатора.

Построение классификатора

В работе рассмотрен байесовский классификатор (Naive Bayes Classifier), согласно которому вероятность того, что документ принадлежит классу c, имеет вид [6]:

$$P(c|d) = P(c)P(d \mid c), \tag{2}$$

где $P(c)$ – априорная вероятность принадлежности документа классу c;

$P(d|c)$ – вероятность того, что среди документов класса c встретится документ d.

Также рассмотрен метод k-ближайших соседей – kNN (k-nearest neighbors algorithm), суть которого заключается в том, что документу d присваивается тот класс c, к которому принадлежит большинство из k-ближайших соседей документа, вычисленных с помощью метрики расстояния [6]. При расчете расстояния в методе k-ближайших соседей использована косинусная мера, которая рассчитывается по формуле:

$$k(d_i, d_j) = \frac{d_i d_j}{\| d_i \| \| d_j \|}, \tag{3}$$

где d_i и d_j – векторные представления документов.

Еще одним рассмотренным в работе методом является метод дерева решений – DT (decision tree), в основе которого лежит принцип индуктивного вывода логических закономерностей [7]. Для классификации документов вуза использован алгоритм построения дерева решений CART (Classification and Regression Tree). При построении дерева используется энтропия Шеннона в качестве функционала качества, на основании которого разбивается дерево [8]:

$$H = -\sum_{j=1}^{m} \frac{N_j}{N} \log\left(\frac{N_j}{N}\right), \tag{4}$$

где m – число классов в обучающей выборке;

N_j – число документов j-го класса;

N – общее число документов в узле.

Из линейных методов классификации документов рассмотрен метод опорных векторов SVM (support-vector machines), суть которого состоит в построении гиперплоскости, максимально разделяющей набор текстов на классы [9]. В работе использован метод опорных векторов с радиальной базисной функцией Гаусса, которая вычисляется по формуле [9]:

$$K(d_i, d_j) = \exp^{-\gamma \| d_i - d_j \|^2}, \tag{5}$$

где d_i, d_j – векторные представления документов;

γ – параметр ядра.

После построения классификатора происходит его обучение с использованием обучающей выборки документов.

Оценка качества работы классификатора

Решение о приемлемости использования метода классификации принимается после оценки качества его работы, которая проводится на тестовой выборке документов вуза. Для вычисления оценок классификации используется матрица несоответствий, представленная в табл. 2.

Таблица 2

Матрица несоответствий предсказаний классификатора [10]

Класс c_j		**Принадлежность классу**	
		Положительная	**Отрицательная**
Оценка классификатора	Положительная	G_p^+	G_p^-
	Отрицательная	G_n^-	G_n^+

Примечание: G_p^+ – классификатор верно определил класс документа; G_n^+ – классификатор верно определил, что документ не относится к классу c_j; G_p^- – классификатор неверно определил класс документа; G_n^- – классификатор неверно определил, что документ не относится к классу c_j.

В исследовании оценка построенного классификатора происходит по правильности алгоритма классификации (J_a), которая показывает долю правильно классифицированных документов относительно всех документов в наборе. Правильность вычисляется по формуле [10]:

$$J_a = \frac{G_p^+ + G_n^+}{G_p^+ + G_p^- + G_n^+ + G_n^-}. \tag{6}$$

Поскольку количество документов в классах несбалансировано, для оценки также взяты метрики точности (J_p) и полноты (J_r) классификации.

Точность классификации J_p показывает долю документов, отнесенных классификатором к рассматриваемому классу, относительно всех документов этого класса [5]. Значение J_p вычисляется по формуле [10]:

$$J_p = \frac{G_p^+}{G_p^+ + G_p^-}. \tag{7}$$

Полнота классификации J_r показывает долю документов, действительно принадлежащих к рассматриваемому классу, относительно всех документов, которые классификатор отнес к этому классу [5]. Значение J_r вычисляется по формуле [10]:

$$J_r = \frac{G_p^+}{G_p^+ + G_n^-}. \quad (8)$$

Значение *F*-меры – гармонического среднего значения между точностью и полнотой классификации (J_F) – вычисляется на основе полученных значений точности и полноты по следующей формуле [10]:

$$J_F = 2 \times \frac{J_p \times J_r}{J_p + J_r}. \quad (9)$$

На основании полученных значений метрик, по которым оценивается работа классификатора, делается вывод о приемлемости использования метода классификации.

Результаты

В результате проведенного исследования были получены оценки классификации (табл. 3).

Таблица 3

Значения оценок классификации для разных методов классификации документов

Метод классификации документов		Метрика классификации				
		J_t	J_a	J_p	J_r	J_F
NB	До обработки	0,003	74,1	77,8	74,1	73,5
	После обработки	0,003	81,0	86,0	81,0	79,1
kNN	До обработки	0,009	89,7	91,9	89,7	89,5
	После обработки	0,003	93,1	94,3	93,1	93,0
DT	До обработки	0,033	86,2	87,4	86,2	86,3
	После обработки	0,019	89,7	91,8	89,7	89,4
SVM	До обработки	0,104	87,9	88,2	87,9	87,8
	После обработки	0,074	94,8	95,0	94,8	94,8

Примечание: J_t – время обучения классификатора; J_a – правильность алгоритма классификации; J_p – точность классификации; J_r – полнота классификации; J_F – *F*-мера. Составлено автором.

Результаты классификации документов до и после предварительной обработки наглядно представлены на рис. 4. Из результатов виден значительный рост всех показателей классификации после предварительной обработки документов, что объясняется высокой зашумленностью документов вуза.

Результаты показывают, что после предварительной обработки документов метод опорных векторов превосходит остальные методы по значениям метрик классификации. Худший результат показал метод NB, поскольку он не учитывает зависимость результата классификации от сочетания слов, что ведет к проблеме «нулевой частоты»: данным, не участвующим в процессе обучения, классификатор присвоит нулевую вероятность и не отнесет эти данные ни к одному из классов. Для решения задачи классификации документов вуза это может быть критично, поскольку в вузе существует большое количество типов документов и документация постоянно обновляется, что может привести к снижению качества классификации байесовским классификатором.

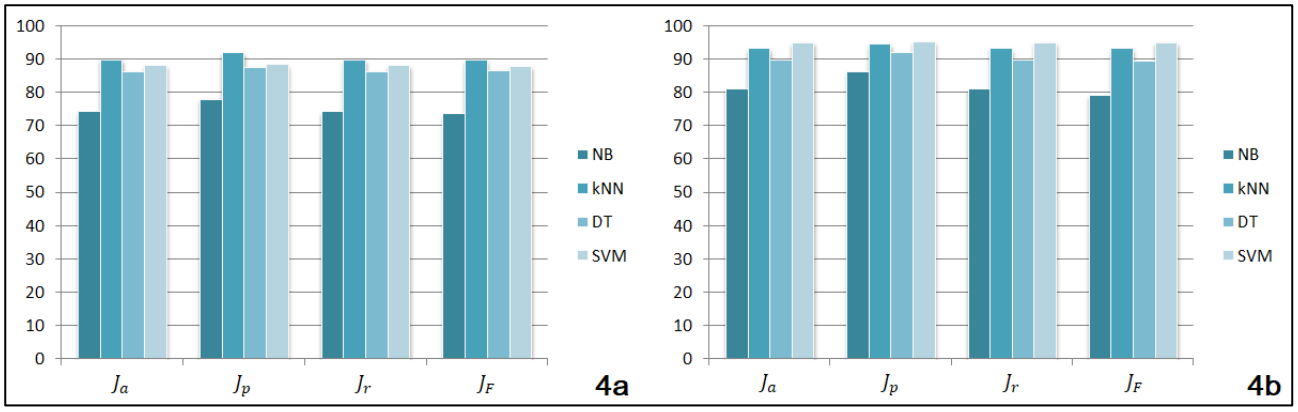

Рис. 4. Оценка метрик качества классификации для алгоритмов NB, kNN, DT и SVM:
4a – график до предварительной обработки документов;
4b – график после предварительной обработки документов
Примечание: составлено автором.

Методы kNN, DT и SVM, по сравнению с NB, показывают более высокие результаты классификации как до, так и после предварительной обработки документов. Значительный прирост значений метрик классификации методами kNN, DT и SVM объясняется их неустойчивостью к шуму.

Также при исследовании выполнена оценка времени, требуемого для обучения классификатора (рис. 5).

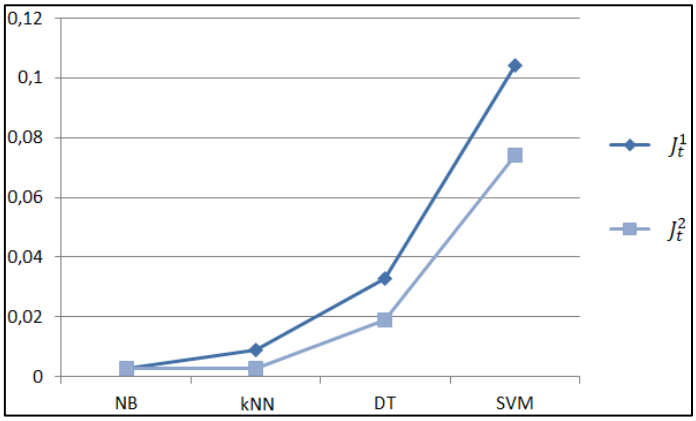

Рис. 5. Сравнение методов классификации документов вуза по времени обучения классификатора:
J_t^1 – время обучения классификатора до предварительной обработки документов;
J_t^2 – время обучения классификатора после предварительной обработки документов
Примечание: составлено автором.

Результаты показывают значительное уменьшение затрачиваемого на обучение классификатора времени для методов kNN, DT и SVM после предварительной обработки документов. Для метода kNN это объясняется тем, что вычисление соответствующих расстояний между всеми тестовыми и обучающими документами делает метод kNN вычислительно затратным при применении к зашумленным текстовым данным. Метод DT неустойчив к выбросам, поэтому для построения решающих правил при классификации предварительно необработанных документов данный метод тратит больше времени. Метод SVM, как и DT, тоже неустойчив к выбросам, поэтому на классификацию необработанных предварительно документов также тратит больше времени. Из полученных результатов видно, что самым быстрым по времени обучения является метод NB. Предварительная обработка документов

практически не повлияла на скорость его работы. Это объясняется тем, что в основе метода лежит предположение о независимости появления слова от контекста, в котором оно было употреблено, поэтому наличие шума практически не влияет на производительность метода.

Из полученных в исследовании результатов можно сделать вывод о применимости каждого из рассмотренных методов для классификации документов вуза. Так, метод NB лучше использовать для распознавания коротких документов с «сильными» ключевыми словами и прямыми отношениями между текстовыми признаками и соответствующими классами (например, для классификации извещений о вакантном месте). Для классификации больших документов вуза, таких как приказы и распоряжения, лучше использовать более точные методы классификации – kNN, DT и SVM.

Заключение

Решена задача классификации документов вуза с помощью методов машинного обучения. Приведены результаты классификации документов до и после их обработки. Показано, что после предварительной обработки документов показатели классификации: в среднем улучшились правильность алгоритма классификации, полнота классификации и *F*-мера – на 6 %, точность классификации – на 5 %, что подтверждает оправданность применения этих методов.

Проведен сравнительный анализ результатов классификации каждого метода машинного обучения, который показал, что лучшим методом классификации документов вуза по правильности, точности и полноте является метод опорных векторов. Однако этот метод показал худшее время обучения классификатора, а лучшее время показали методы NB и kNN. Поэтому представляется перспективным в дальнейшем для повышения качества работы и производительности классификатора использовать ансамбль из рассмотренных методов.

Литература

1. Ткаченко А. Л. Обзор методов интеллектуального анализа документов // Информационные технологии и автоматизация управления : материалы XI Всерос. науч.-практ. конф. Омск, 2020. С. 2018–2027.

2. Кажемский М. А., Шелухин О. И. Многоклассовая классификация сетевых атак на информационные ресурсы методами машинного обучения // Тр. учеб. заведений связи. 2019. Т. 5, № 1. С. 107–115.

3. Рубцова Ю. С. Методы и алгоритмы построения информационных систем для классификации текстов социальных сетей по тональности : дис. … канд. техн. наук. Новосибирск, 2019. 141 с.

4. Zhang X., Zhao J., LeCun Y. Character-level Convolutional Networks for Text Classification // Neural Information Processing Systems. 2015. Vol. 28. P. 649–657.

5. Батура Т. В. Методы автоматической классификации текстов // Программн. продукты и системы. 2017. Т. 30, № 1. С.85–99.

6. Бондарчук Д. В. Алгоритмы интеллектуального поиска на основе метода категориальных векторов: дис. … канд. техн. наук. Екатеринбург, 2016. 141 с.

7. Jiang L., Li C., Wang S., Zhang L. Deep Feature Weighting for Naive Bayes and its Application to Text Classification // Engineering Applications of Artificial Intelligence. 2016. Vol. 52. P. 26–39.

8. Серобабов А. С. Формирование диапазонов переменных экспертной системы на соответствие нормальному закону распределения // Проблемы и перспективы студ. науки. 2019. № 2. С. 3–6.

9. Nguyen L. Text Classification Based on Support Vector Machine // Dalat University Journal of Science. 2019. Vol. 9, Iss. 2. P. 3–19.

10. Оценка качества в задачах классификации. URL: https://neerc.ifmo.ru/wiki/index.php?title (дата обращения: 10.02.2021).