

УДК 004.89

Е. И. Бурлаева, В. Н. Павлыш
ГОУ ВПО «Донецкий национальный технический университет»
283001, г. Донецк, ул. Артема, 58

ПРОЕКТ ПОСТРОЕНИЯ АЛГОРИТМА КЛАССИФИКАЦИИ ТЕКСТОВЫХ ДОКУМЕНТОВ

E. I. Burlaeva, V. N. Pavlysh
SHEE «Donetsk National Technical University»
283001, c. Donetsk, Artyoma str., 58

PROJECT OF BUILDING THE ALGORITHM OF CLASSIFICATION OF TEXT DOCUMENTS

К. І. Бурлаєва, В. М. Павлиш
ДОУ ВПО «Донецький національний технічний університет»
283001, м. Донецьк, вул. Артема, 58

ПРОЕКТ ПОБУДОВИ АЛГОРИТМУ КЛАСИФІКАЦІЇ ТЕКСТОВИХ ДОКУМЕНТІВ

В статье рассматривается новый алгоритм машинного обучения для классификации текстовых документов, который можно использовать для более эффективной структуризации и поиска необходимой информации. Проиллюстрирован общий принцип алгоритма, применяемый к задаче улучшения качества классификации. Модель алгоритма основывается на классических методах машинного обучения, применение которых даёт возможность получить структуру алгоритма для представления знаний в интеллектуальных системах.

Ключевые слова: метод, алгоритм, текстовый документ, машинное обучение.

The new machine-learning algorithm for the classification of text documents, which may be used for more efficient structuring and retrieval of necessary information, is considered. The general principle of the algorithm applied to the problem of improving the quality of classification is illustrated. The model of the algorithm is based on the classical methods of machine learning, their application received the composition of the algorithm to represent knowledge in intelligent systems.

Key words: method, algorithm, text document, machine learning.

В статті розглянуто новий алгоритм машинного навчання для класифікації текстових документів, який можливо використовувати для більш ефективної структуризації і пошуку необхідної інформації. Проілюстровано загальний принцип алгоритму, який застосовується до задачі поліпшення якості класифікації. Модель алгоритму базується на класичних методах машинного навчання, застосування яких дає можливість отримати структуру алгоритму для представлення знань в інтелектуальних системах.

Ключові слова: метод, алгоритм, текстовий документ, машинне навчання.

Актуальность работы

Современный этап человеческой деятельности характеризуется значительным возрастанием количества информации в хранилищах данных и расширением её рубрикации, что ограничивает возможность своевременного получения требуемой информации по конкретной теме и вызывает появление большого количества неиспользуемых накопленных данных, представленных текстами в электронном виде.

Одной из технологий обработки текстовой информации является автоматическая классификация текстовых документов. Использование этой технологии позволяет сократить время на обработку электронных документов.

Для решения этого вопроса часто применяются различные тематические классификаторы рубрикаторы и т.д., которые облегчают поиск информации автоматически или вручную.

Одной из самых эффективных становится идея алгоритма объединения нескольких классификаторов в так называемую композицию. В этой связи тема работы является актуальной.

Цель работы – анализ вариантов предобработки данных и обоснование структуры алгоритма, использующего идею композиции алгоритмов системы многоклассовой текстовой классификации.

Основное содержание работы

В настоящее время достаточно часто используются системы управления знаниями, которые применяются для решения широкого круга задач. В данной работе рассмотрен подход к решению одной из таких задач – классификации документов.

Под текстовой классификацией понимается отнесение текстовой информации к одной или нескольким категориям в соответствии с определенными признаками.

Для достижения поставленной цели проведены теоретические исследования процесса классификации текста [1-3].

При построении алгоритма была использована композиция существующих алгоритмов. Предложена компоновка методов, позволяющая сократить временные затраты на обработку текстовых документов и повысить точность отнесения документа к той или иной рубрике. Для построения алгоритма использованы положительные качества методов машинного обучения и сведены к минимуму недостатки.

Алгоритм классификации текстов состоит из нескольких блоков:

- предобработка векторов;
- метод классификации.

Успешный, качественный алгоритм – это удачно подобранные все «кирпичики» алгоритма.

Анализируя результаты опубликованных экспериментов, можно увидеть, что конечный результат зависит от предобработки входных данных не в меньшей степени, чем от метода машинного обучения.

Задача выбора модели классификатора документов зависит от предметной области и характеристик рассматриваемых текстов. Реализации существующих методов машинного обучения являются универсальными и не учитывают специфики предметной области.

Интерес в данной работе представляют варианты предобработки данных и преобразование при помощи композиций алгоритмов [4], [5], в которых ошибки отдельных классических алгоритмов взаимно компенсируются.

Далее будет представлено описание алгоритма построения классификатора. Алгоритм состоит из 5 шагов, выполняемых последовательно. Первые три шага алгоритма представляют собой предварительную обработку документа. Четвертый и пятый шаг являются композицией алгоритмов.

Описываемый алгоритм имеет ряд параметров, которые могут влиять на скорость вычисления и качество (полноту, точность, размер) получаемых результатов.

Рассмотрим последовательность построения алгоритма.

Первым шагом алгоритма является составление текстового набора данных на русском языке, который будет состоять из n -го числа документов, то есть формируется своеобразная выборка, на которой будут проводиться эксперименты, так как русский язык имеет свою специфику.

Второй шаг включает в себя несколько способов предобработки текста:

- стемминг;
- стоп-слова.

Стемминг – удаление окончания; приведение слова к основе (единому корню) [6]. Русский язык относится к группе флективных синтетических языков, то есть языков, в которых преобладает словообразование с использованием аффиксов, сочетающих сразу несколько грамматических значений. Пример приведен в табл. 1.

Таблица 1 – Пример стемминга

Список словоформ	Представление после сжатия
Программа	Программа*
Программы	8ы*
Программе	8е*
Программой	8ом*
Программе	8е*
Программы	8ы*
Программ	8*
Программам	9м *
Программами	9ми*
Программах	9х*
Программировать	8ировать*
Программировал бы	8ировал бы*

Статистика показывает, что наибольшее число парадигм имеют длину основ от 6 до 10 букв. Это позволяет значительно сократить объем занимаемой словарями памяти. Используемые в системе словари русского языка из порядка 1 500 000 словоформ занимают менее 2 Мб каждый вместе с необходимыми для адресации таблицами.

Стоп-слова [7]. Словарь стоп-слов – это топ-символы, они же стоп-слова, это отнюдь не слова, на которых все останавливаются; их еще называют «шумовые слова», это слова, встречающиеся практически во всех текстах и не несущие специальной смысловой нагрузки. Обычно, в большинстве алгоритмов обработки текстов, эти слова удаляются.

В русском языке к стоп-символам относятся предлоги, суффиксы, причастия, междометия, частицы и т.п., пример в табл. 2.

Таблица 2 – Пример списка стоп-слов

-	г	их	нас	сказать	сам
а	где	к	не	со	свое
без	говорил	кажется	него	совсем	свою
более	да	как	нее	так	себе
больше	даже	какая	ней	такой	себя
будет	два	какой	нельзя	там	сегодня
будто	для	когда	нет	тебя	сейчас
бы	до	конечно	ни	тем	сказал
был	другой	которого	ни будь	теперь	сказала
была	его	которые	никогда	то	у
были	ее	кто	ним	тогда	уж
было	ей	куда	них	того	уже
быть	ему	ли	ничего	тоже	хорошо
в	если	лучше	но	только	хоть
вам	есть	между	ну	том	чего
вас	еще	меня	о	тот	человек
вдруг	ж	мне	об	три	чем
ведь	же	много	один	тут	через
во	жизнь	может	он	ты	что
вот	за	можно	она	потом	чтоб
впрочем	зачем	мой	они	потому	чтобы
все	здесь	моя	опять	почти	чуть
всегда	из	мы	от	при	эти
всего	из-за	на	перед	про	этого
всех	или	над	по	раз	этой
всю	им	надо	под	разве	этом
вы	иногда	наконец	после	с	этот
					эту
					я

Помимо указанных слов имеет смысл еще фильтровать цифры, отдельные буквы и знаки препинания.

Получившийся набор слов переводим в векторную модель с помощью *tf-idf* модели, не учитывая (не присваивая вес) стоп-словам (словам из словаря стоп-слов).

Tf-idf [8], [9] характеризует баланс между частотой появления термина в документе и частотой его появления в наборе документов.

Отдельной задачей при преобразовании текста в вектор является вычисление значений координат в пространстве Rn , соответствующих признакам, также называемых весами признаков. Выбор весов признаков существенно влияет на качество рубрицирования.

Одной из формул вычисления весов является [10], [11]:

$$tf - idf(t, d, D) = tf(t, d) \times idf(t, D) \quad (1)$$

где $tf - idf(t, d, D)$ – вес i -го слова, $tf(t, d) = \frac{n_i}{\sum_k n_k}$ отношение числа вхождений

некоторого слова к общему числу слов документа (*term frequency* – частота

слова), $idf(t, D) = \log \frac{|D|}{|\{d_i \in D | t \in d_i\}|}$ – логарифм отношения количества всех докумен-

тов в коллекции к количеству документов, в которых встречается i -е слово (*inverse document frequency* – частота обратного документа).

Такой выбор формулы можно обосновать теоретически следующими соображениями:

1. Чем чаще слово встречается в документе, тем оно важнее. Этот факт учитывает множитель $tf(t, d,)$.

2. Если слово встречается во многих или во всех документах, то это слово не может являться существенным критерием принадлежности документа рубрике и его вес следует понизить. Наоборот, если слово встречается в малом количестве документов, то его вес следует повысить. Множитель $idf(t, D)$ учитывает это и соответствует весу слова в данной коллекции документов.

Полученный вектор будем использовать на шаге 4 для классификации документа с помощью алгоритма SVM.

Метод опорных векторов (SVM), предложенный Вапником (Cortes & Вапника, 1995) [12-14], является контролируемым алгоритмом обучения, который может быть применен к классификации. Это двоичный линейный классификатор, который отделяет позитивный и негативный примеры в тестовом наборе. Метод ищет гиперплоскости, отделяющие положительные примеры от отрицательных примеров, гарантируя, что граница между ближайшими позитивами и негативами является максимальной. Эффективность SVM превосходит другие методы классификации текста. SVM делает модель, представляющую примеры обучения в качестве точек в пространстве разделенных гиперплоскостью, и она использует эту модель, чтобы предсказать, к какой стороне этой гиперплоскости принадлежит новый пример.

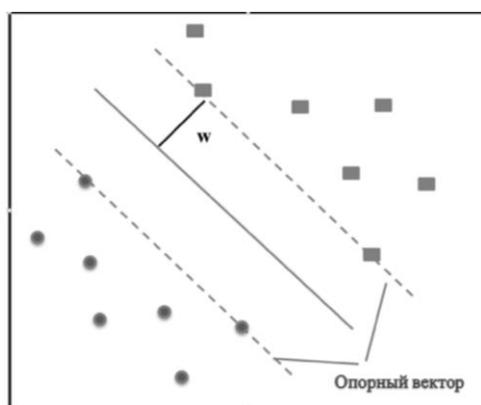


Рисунок 1 – Максимальное значение предела гиперплоскости

Часто в алгоритмах машинного обучения возникает необходимость классифицировать данные. Каждый объект данных представлен как вектор (точка) в p -мерном пространстве (последовательность p чисел). Каждая из этих точек принадлежит только одному из двух классов. Нас интересует, можем ли мы разделить точки гиперплоскостью размерности $(p-1)$. Это типичный случай линейной делимости. Таких гиперплоскостей может быть много. Поэтому вполне естественно полагать, что максимизация зазора между классами способствует более уверенной классификации. То есть, можем ли мы найти такую гиперплоскость, чтобы расстояние от неё до ближайшей точки было максимальным. Это бы означало, что расстояние между

двумя ближайшими точками, лежащими по разные стороны гиперплоскости, максимально. Если такая гиперплоскость существует, то она нас будет интересовать больше всего; она называется оптимальной разделяющей гиперплоскостью, а соответствующий ей линейный классификатор называется оптимально разделяющим классификатором.

После того как алгоритм SVM отработал, смотрим, попал ли документ в промежуток от -1 до 1 , если попал, то классифицируем этот документ с помощью алгоритма латентно-семантического анализа. Если документ не попал в этот промежуток, то относим документ к тому классу, который ему определил алгоритм SVM.

Латентно-семантический анализ (*LSA, Latent Semantic Analysis*) – это теория и метод для извлечения контекстно-зависимых значений слов при помощи статистической обработки больших наборов текстовых данных. LSA был запатентован в 1988 году [15], [16]. В области информационного поиска данный подход называют латентно-семантическим индексированием (*LSI, Latent Semantic Indexing*). LSA также работает с векторным представлением типа «мешка слов» текстовых единиц. Текстовый корпус представляется в виде числовой матрицы – слово-текст, строки которой соответствуют словам, а столбцы текстовым единицам – текстам. Существуют различные схемы определения каждого элемента данной матрицы, мы используем *tf-idf*.

Объединение слов в темы и представление текстовых единиц в пространстве тем осуществляется путем применения к данной матрице одного из матричных разложений. Например, сингулярное разложение и факторизация неотрицательных матриц. Согласно теореме о сингулярном разложении, любая вещественная прямоугольная матрица может быть разложена на произведение трех матриц: $A = USV^T$, где $A \in R^{n \times m}$, матрицы $U \in R^{n \times k}$ и $V \in R^{m \times k}$ – ортогональные, а $S \in R^{k \times k}$ – диагональная матрица, значения на диагонали которой называются сингулярными значениями матрицы A , V^T – транспонированная матрица.

Если в матрице S оставить только k наибольших сингулярных значений, а в матрицах U и V – только соответствующие этим значениям столбцы, то произведение получившихся матриц S , U и V будет наилучшим приближением исходной матрицы A к матрице A' ранга k : $A' \approx A = USV^T$. Так как в качестве матрицы A использовалась матрица слова-на-тексты, то матрица A' , содержащая только k первых линейно независимых компонент A , отражает основную структуру различных зависимостей, присутствующих в исходной матрице. Структура зависимостей определяется весовыми функциями слов.

Таким образом, каждое слово и текст представляются при помощи векторов в общем пространстве размерности k – пространстве гипотез. Сходство между любой комбинацией слов и/или текстов вычисляется при помощи скалярного произведения векторов. Как правило, выбор k зависит от поставленной задачи и подбирается эмпирически. Если выбранное значение k слишком велико, то метод теряет свою мощьность и приближается по характеристикам к стандартным векторным методам. Слишком маленькое значение k не позволяет улавливать различия между похожими словами или текстами.

Выводы

В результате выполненной работы исследована идея, лежащая в основе ансамблевого алгоритма, представляющего объединение нескольких методов машинного обучения классификации для достижения более высокой точности классификации объектов. Представлена подобная система, которая показала пошаговый процесс построения алгоритма.

- Дальнейшая работа над алгоритмом планируется по следующим направлениям:
- изучение влияния различных метрик сходства объектов (как, например, основанные на важности признаков или приросте информации) на производительность алгоритма;
 - проведение экспериментов над различными категориями данных;
 - исследование условий, при которых предпочтительно использование данного алгоритма;
 - работа над временной сложностью алгоритма.

Список литературы

1. Schutze H. A comparison of classifiers and document representations for the routing problem [Text] / H. Schutze, D. A. Hull, J. O. Pedersen // Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval, Seattle. – 1995. – P. 229–237.
2. Ageev M. Text Categorization Tasks for Large Hierarchical Systems of Categories [Text] / M. Ageev, B. Dobrov, N. Loukachevitch // SIGIR 2002 Workshop on Operational Text Classification Systems / Eds. F. Sebastiani, S. Dumas, D.D. Lewis, T. Montgomery, I. Moulinier. – Tampere : Univ. of Tampere, 2002. – P. 49–52.
3. Rose T. The Reuters Corpus Volume 1 - from Yesterday News to tomorrow's Language [Text] / T. Rose, M. Stevenson, M. Whitehead // Proc. of the Third Int. Conf. on Language Resources and Evaluation. – Las Palmas de Gran Canaria, 2002.
4. Michael J. Kearns. An Introduction to Computational Learning Theory [Text] / Michael J. Kearns and Umesh V. Vazirani. – MIT Press, Cambridge, MA, USA, 1994.
5. Anil K. Jain. Statistical pattern recognition: A review [Text] / Anil K. Jain, Robert P. W. Duin, and Jianchang Mao // IEEE Transactions on Pattern Analysis and Machine Intelligence. – Vol. 22, № 1. – 2000. – P. 4–37.
6. Аношкина Ж. Г. Морфологический процессор русского языка [Текст] / Ж. Г. Аношкина // Бюллетень машинного фонда русского языка / отв. редактор В.М. Андрищенко – М., 1996. – Вып.3. – С. 53–57.
7. Bill Frakes. Stemming algorithms [Text] / Bill Frakes // Information Retrieval: Data Structures and Algorithms. Englewood Cliffs, US. – 1992. – P. 131–160.
8. Губин М. В. Модели и методы представления текстового документа в системах информационного поиска : дис. ... к.ф.-м.н. / М. В. Губин. – СПб., 2005. – 95 с.
9. Salton G. Weighting Approaches in Automatic Text Retrieval [Text] / Salton G, Buckley C. / Information Processing and Management. – 1988. – P. 513–523.
10. Епрев А. С. Автоматическая классификация текстовых документов [Текст] / А. С. Епрев // Математические структуры и моделирование. – 2010. – № 21. – С. 5–81.
11. Коршунов А. Тематическое моделирование текстов на естественном языке [Текст] / А. Коршунов, А. Гомзин // Труды Института Системного Программирования РАН. – 2012. – Т. 23. – С. 216–243.
12. Vapnik V. The Nature of Statistical Learning Theory [Text] / V. Vapnik // Springer-Verlag. – New York, 1995.
13. Choosing Multiple Parameters for Support Vector Machines Chapelle O., Vapnik V., Bousquet O., Mukherjee S. // Machine Learning. – 2002. – Vol. 46, № 1. – P. 131–159.
14. Вьюгин В. Математические основы машинного обучения и прогнозирования [Текст] / Вьюгин В. – МЦМНО, 2014. – 304 с.
15. Kim H. Nonnegative Matrix Factorization Based on Alternating Nonnegativity Constrained Least Squares and Active Set Method [Text] / H. Kim, H. Park // SIAM Journal on Matrix Analysis and Applications, 2008. – Vol. 30, № 2. – P. 713–730.
URL: <http://www.cc.gatech.edu/~hpark/papers/simax-nmf.pdf> (дата обращения 23.10.2017).
16. Landauer T. K. An Introduction to Latent Semantic Analysis [Text] / T. K. Landauer, P. Foltz, D. Laham // Discourse Processes. – 1998. – Vol. 25. – P. 259–284.
URL : <http://lsa.colorado.edu/papers/dp1.LSAintro.pdf>. (дата обращения 23.10.2017).
17. Павлыш В. Н. Математическое моделирование процессов функционирования специализированных аппаратов конвективного типа [Текст] / В. Н. Павлыш, Е. В. Перинская // Проблемы искусственного интеллекта. – Донецк : ГУ ИПИИ. – 2015. – № 0(1). – С. 89–98.
18. Pavlysh V. N. Modification of computer methods of presentation and analysis of geotechnical information [Text] / V. N. Pavlysh, G. I. Turchanin, O. A. Tikhonova // Проблемы искусственного интеллекта. – Донецк : ГУ ИПИИ. – 2016. – № 1 (2). – С. 15–24.

References

1. Schutze H., Hull D. A., Pedersen J. O. A comparison of classifiers and document representations for the routing problem. *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval, Seattle*. 1995. P. 229-237.
2. Ageev M., Dobrov V., Loukachevitch N. Text Categorization Tasks for Large Hierarchical Systems of Categories. *SIGIR 2002 Workshop on Operational Text Classification Systems* / Eds. F. Sebastiani, S. Dumas, D.D. Lewis, T. Montgomery, I. Moulinier. Tampere: Univ. of Tampere, 2002. P. 49-52.
3. Rose T., Stevenson M., Whitehead M. The Reuters Corpus Volume 1 - from Yesterday News to tomorrow's Language. *Proc. of the Third Int. Conf. on Language Resources and Evaluation, Las Palmas de Gran Canaria*. 2002.
4. Michael J. Kearns and Umesh V. Vazirani, *An Introduction to Computational Learning Theory*, MIT Press, Cambridge, MA, USA, 1994.
5. Anil K. Jain, Robert P. W. Duin, and Jianchang Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4-37, 2000.
6. Anoshkina Zh.G. Morfologicheskij processor russkogo jazyka. *Bjulleten' mashinnogo fonda russkogo jazyka* / otv. redaktor V.M. Andrjushhenko M., 1996. Vyp.3, s.53-57.
7. Bill Frakes. Stemming algorithms. *Information Retrieval: Data Structures and Algorithms*. Englewood Cliffs, US. 1992. P. 131-160.
8. Gubin, M.V. Modeli i metody predstavlenija tekstovogo dokumenta v sistemah informacionnogo poiska: dis. ... k.f.m.n./M.V. Gubin. SPb., 2005. 95 s.
9. Salton G, Buckley C. Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*. 1988. pp. 513-523.
10. Eprev A. S. Avtomaticheskaja klassifikacija tekstovyh okumentov. *Matematicheskie struktury i modelirovanie* 2010. no. 21. pp. 5-81.
11. Korshunov A., Gomzin A. Tematicheskoe modelirovanie tekstov na estestvennom jazyke. *Trudy Instituta Sistemnog Programirovanija RAN*, 2012. T. 23. S. 216-243.
12. Vapnik V. *The Nature of Statistical Learning Theory*. Springer-Verlag; New York, 1995.
13. Chapelle O., Vapnik V., Bousquet O., Mukherjee S. Choosing Multiple Parameters for Support Vector Machines. *Machine Learning*. 2002. Vol. 46, No. 1. P. 131-159.
14. V'jugin V. Matematicheskie osnovy mashinnogo obuche-nija i prognozirovanija. MCMNO, 2014. 304 s.
15. Kim H., Park H. Nonnegative Matrix Factorization Based on Alternating Nonnegativity Constrained Least Squares and Active Set Method. *SIAM Journal on Matrix Analysis and Applications*, 2008. Vol. 30, No. 2. P. 713-730. URL: <http://www.cc.gatech.edu/~hpark/papers/simax-nmf.pdf> (data obrashhenija 23.10.2017).
16. Landauer T. K., Foltz P., Laham D. An Introduction to Latent Semantic Analysis // *Discourse Processes*. 1998. Vol. 25. P. 259-284. URL: <http://lsa.colorado.edu/papers/dp1.LSAintro.pdf>. (data obrashhenija 23.10.2017).
17. Pavlysh V. N., Perinskaya E. V. Mathematical modeling of functioning processes of special convective type apparatus. *Problems of Artificial Intelligence*, Donetsk, 2015, no. 0(1), pp. 89-98.
18. Pavlysh V. N., Turchanin G. I., Tikhonova O. A. Modification of computer methods of presentation and analysis of geotechnical information. *Problemy iskusstvennogo intellekta* [Problems of Artificial Intelligence], 2016, no. 1 (2), pp. 15-24.

RESUME

E.I. Burlaeva, V.N. Pavlysh

The design of the classification algorithm for text documents

Introduction: The current stage of human activity is characterized by a significant increase in the amount of information in data warehouses and the expansion of its classification, which limits the ability to receive timely the required information on a particular topic and causes the emergence of a large number of unused accumulated data submitted electronically. In this paper, we consider an approach to solving one of such problems - classification of documents.

A text classification is understood as the assignment of textual information to one or several categories in accordance with certain characteristics.

When constructing the algorithm, the composition of existing algorithms was used. A layout of methods is proposed, which makes it possible to reduce the time spent on processing text documents and improve the accuracy of assigning a document to this or that heading. To build the algorithm, positive qualities of machine learning methods are used and deficiencies are minimized.

Materials and methods: Currently, management systems and knowledge are often used, which are used for a wide range of tasks. In this paper, we consider an approach to solving one of such problems - classification of documents.

A text classification is understood as the assignment of textual information to one or several categories in accordance with certain characteristics.

A successful, high-quality algorithm is a well-chosen all the "bricks" of the algorithm.

Analyzing the results of the published experiments, one can see that the final result depends on the preprocessing of the input data no less than on the method of machine learning.

The task of selecting a document classifier model depends on the subject area and characteristics of the texts in question. Implementations of existing methods of machine learning are universal and do not take into account the specifics of the subject area.

The interest in this work is represented by options for preprocessing data and transformation using algorithms compositions in which the errors of individual classical algorithms are mutually compensated.

Results: In the construction of the algorithm, a composition of existing algorithms was used. A layout of methods is proposed, which makes it possible to reduce the time spent on processing text documents and improve the accuracy of assigning a document to this or that heading. To build the algorithm, positive qualities of machine learning methods are used and deficiencies are minimized.

Conclusion: The key is the idea of combining several classifiers into the so-called algorithm composition to achieve a higher accuracy of object classification. A similar system was also presented, which showed the step-by-step process of constructing the algorithm.

Статья поступила в редакцию 05.10.2017.