УДК 004
DOI https://doi.org/10.32782/IT/2023-2-4

Борис МОРОЗ
доктор технічних наук, професор, професор кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005, moroz.boris.1948@gmail.com
***ORCID:** 0000-0002-5625-0864*
***Scopus-Author ID:** 57218242332*

Леонід КАБАК
кандидат технічних наук, доцент, доцент кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005, kabak.leo@gmail.com
***ORCID:** 0000-0001-6267-1772*
***Scopus-Author ID:** 5720222205*

Нонна ВАРЕХ
кандидат наук із соціальних комунікацій, доцент кафедри мовних та гуманітарних дисциплін, Технічний університет «Метінвест політехніка», вул. Південне шосе, 80, м. Запоріжжя, Україна, 69008, nonna.varekh@gmail.com
***ORCID:** 0000-0002-2779-9225*
***WoS ResearcherID:** ABG-3294-2021*

Дмитро МОРОЗ
викладач кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005, dmitriy@moroz.cc
***ORCID:** 0000-0003-2577-3352*
***Scopus-Author ID:** 57369936300*

Бібліографічний опис статті: Мороз, Б., Кабак, Л., Варех, Н., Мороз, Д. (2023). Система класифікації текстових документів із використанням технологій Big Data. *Information Technology: Computer Science, Software Engineering and Cyber Security,* 2, 34–40, doi: https://doi.org/10.32782/IT/2023-2-4

СИСТЕМА КЛАСИФІКАЦІЇ ТЕКСТОВИХ ДОКУМЕНТІВ ІЗ ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ BIG DATA

У роботі було розглянуто модель системи класифікації документів з використанням технології Big Data. При використанні технології Big Data на сервері накопичується великий масив документів, які потрібно попередньо обробити та завантажити у базу даних. В документах потрібно визначити ключові слова за допомогою яких їх потрібно віднести до однієї або декількох тематичних розділів. Крім того розроблена система повинна працювати швидко та передбачати автоматичне навчання.

Отже розробка моделей та методів класифікації текстових документів на дійсний час є актуальним завданням. Дуже інтенсивний розвиток цих методів спостерігається в останній час при стрімкому розвитку обчислювальної техніки, та при переході багатьох організацій на електронний документообіг. В результаті дослідження було розроблено метод та модель системи; запропоновано комбінацію підходів для навчання моделі; визначено найбільш продуктивну модель для навчання системи.

***Метою роботи** є проведення аналізу існуючих методів класифікації текстових документів та розробити модель та метод класифікації текстових документів з використанням технології MapRaduce.*

***Методологія** вирішення поставленого завдання полягає в проведенні порівняльного аналізу показників продуктивності різних конфігурацій системи, які запроваджені з урахуванням попередніх досліджень моделей систем класифікації документів, які використовують технологію Big Data.*

***Наукова новизна**. У роботі запропоноване нове рішення для виконання точної байєсовської класифікації на основі Spark. Цей класифікатор використовує велику кількість операції в пам'яті сервера, щоб класифікувати велику кількість текстових документів на основі великого навчального набору даних з використанням MapReduce. Фаза карти обчислює кількість входжень ключових слів у різних розподілах даних навчання. Після цього кілька редукторів обчислюють вірогідність віднесення документу до певних класів,*

на підставі обчислень отриманих на етапі карти. Ключовий момент цієї пропозиції полягає в управлінні набором текстових документів, зберігаючи їх в пам'яті, коли це можливо.

***Висновки**. Результати даної роботи можуть бути використані для реалізації ефективної системи класифікації текстової документації, яка використовує точний байєсовської класифікатор, з використання мови програмування Python в поєднанні з сервісом Hadoop Big Data .*

***Ключові слова:** Big Data, Hadoop, Map Reduce, Apache Spark , алгоритми машинного навчання, системи класифікації, байєсовський класифікатор.*

Borys MOROZ
Doctor of Technical Sciences, Professor, Professor at the Department of Software Engineering, Dnipro University of Technology, Dmytro Yavornytskyi ave., 19, Dnipro, Ukraine, 49005, moroz.boris.1948@gmail.com
***ORCID:** 0000-0002-5625-0864*
***Scopus-Author ID:** 57202222055*

Leonid KABAK
Candidate of Technical Sciences, Associate Professor, Associate Professor at the Department of Software Engineering, Dnipro University of Technology, Dmytro Yavornytskyi ave., 19, Dnipro, Ukraine, 49005; Associate Professor at the Department of Digital Technologies and Project-Analytical Solutions, Technical University "Metinvest Polytechnic", Pivdenne shose str., 80, Zaporizhzhia, Ukraine, 69008, kabak.leo@gmail.comm
***ORCID:** 0000-0001-6267-1772*

Nonna VAREKH
Candidate of Sciences in Social Communications, Associate Professor at the Department of Linguistic and Humanitarian Disciplines, Technical University "Metinvest Polytechnic", Pivdenne shose str., 80, Zaporizhzhia, Ukraine, 69008, nonna.varekh@gmail.com
***ORCID:** 0000-0002-2779-9225*
***WoS ResearcherID:** ABG-3294-2021*

Dmitriy MOROZ
Lecturer at the Department of Software Engineering, Dnipro University of Technology, Dmytro Yavornytskyi ave., 19, Dnipro, Ukraine, 49005, dmitriy@moroz.cc
***ORCID:** 0000-0003-2577-3352*
***Scopus-Author ID:** 57369936300*

To cite this article: Moroz, B., Kabak, L., Varekh, N., Moroz, D. (2023). Systema klasyfikatsii tekstovykh dokumentiv iz vykorystanniam tekhnolohii Big Data [Text document classification system with Big Data technologies usage]. *Information Technology: Computer Science, Software Engineering and Cyber Security,* 2, 34–40, doi: https://doi.org/10.32782/IT/2023-2-4

TEXT DOCUMENT CLASSIFICATION SYSTEM WITH BIG DATA TECHNOLOGIES USAGE

***The aim.** The paper considered a model of the document classification system using Big Data technology. When using Big Data technology, a large array of documents accumulates on the server which must be pre-processed and uploaded to the database. In the documents you need to define keywords with a help of which you need to assign them to one or more thematic sections. In addition, the developed system should operate fast and provide automatic learning.*

Therefore, the development of models and methods of classification of text documents for real time is an urgent task. A very intensive development of these methods has been observed recently with the rapid development of computer technology and with the transition of many organizations into electronic document management. As a result of the study, a method and a system model were developed; a combination of approaches for model training is proposed; the most productive model for system training is determined.

***Scientific novelty**. The paper proposes a new solution for performing accurate Bayesian classification based on Spark. This classifier uses a large number of in-memory server operations to classify a large number of text documents based on a large training dataset using MapReduce. The map phase calculates the number of occurrences of keywords in different distributions of the training data. After that, several reducers calculate the probability of assigning of a document to certain classes based on the calculations obtained at the map stage. The key point of this proposal is to manage a set of text documents keeping them in memory whenever possible.*

***Conclusions**. The results of this work could be used for implementation of an effective classification system for text documents that uses an accurate Bayesian classifier developed with the Python programming language in combination with the Hadoop Big Data service.*

***Key words:** Big Data, Hadoop, Map Reduce, Apache Spark, Machine Learning Algorithm, systems of classification, Bayes Classifier.*

Актуальність проблеми. При використанні в автоматизованих інформаційних системах технологій Big Data виникає необхідність в обробці великих масивів документів. Однією з актуальних задач в процесі обробки текстових документів є їх попередня класифікація, яка передбачає віднесення документа до одного або декількох тематичних розділів. В дійсний час розроблено багато методів що дозволяють вирішити такі завдання. Однак більшість методів розраховані на пошук документів в мережі Internet за допомогою ключових слів. Пошукова система видає користувачу безліч документів з яких користувач вибирає необхідні йому документів.

При використанні технології Big Data на сервері накопичується великий масив документів, які потрібно попередньо обробити та завантажити у базу даних. В документах потрібно визначити ключові слова за допомогою яких їх потрібно віднести до однієї або декількох тематичних розділів. Крім того розроблена система повинна працювати швидко та передбачати автоматичне навчання.

Отже розробка моделей та методів класифікації текстових документів на дійсний час є актуальним завданням. Дуже інтенсивний розвиток цих методів спостерігається в останній час при стрімкому розвитку обчислювальної техніки, та при переході багатьох організацій на електронний документообіг.

Отже, основна мета цієї роботи є розробка інформаційної системи в якій передбачається обробка та класифікація документів, які потрапляють до сервера, з використанням існуючих та з використанням нових методів класифікації.

Аналіз останніх досліджень і публікацій. Проблемою розпізнавання образів та автоматизованої класифікації текстових документів вчені займаються починаючи з 60-х років класичними роботами в цьому напряму є роботи (Gonzalez, Thomason,1974; Gonzalez, Tou, 1968; Salton, 1986) вчених Gonzalez R.C., Thomason M G. в цих роботах було досліджені різні типи класифікаторів. Класичним методом класифікації в асоціативно-статистичному підході є метод і алгоритм, що його реалізує, запропонований Дж. Солтоном в 1975 р (Salton, 1986). У 1979 р. алгоритм був уточнений і доопрацьований у Дж. Солтоном і названий TFxIDF. Цей алгоритм є на сьогоднішній день найбільш ефективним, поширеним і використовується в сучасних інформаційно-пошукових системах.

Метод, розроблений Дж. Солтоном, ґрунтується на так званій «векторній моделі тексту». У різних джерелах можна зустріти різну назву цієї моделі: «векторна», «лінійна» або «алгебраїчна».

В дійсний час існує багато методів класифікації текстових документів які використовують нейронні мережі, різноманітні методи кластеризації, методи опорних векторів. У статі пропонується розробка автоматизованої системи класифікації тестових документів з використанням платформи Apache Spark яка вбудована в систему Big Data Hadoop. Класичними роботами у цьому напряму є роботи авторів П. Сембереські та Г. Мацієвський (Semberecki, Maciejewski, 2016). У цій статті було показано, як ці послідовні кроки можна реалізувати на платформі Apache Spark, призначеній для розподіленої обробки великих даних. Авторами статі було проілюстровано запропонований метод, який є зразком класифікатора, призначеного для прогнозування тематичної категорії документа в англомовній Вікіпедії.

У роботі (Pintye, Kail, Kacsuk, Lovas) було розглянуто різноманітність технологій і протоколів. У документі зосереджено увагу на широко поширеному кластері Apache Spark із Jupyter як особливо адресованою структурою, а також на інструменті Occopus, що не залежить від хмари, для автоматизації етапів його розгортання та обслуговування. У цій статі був представлений підхід та було продемонстровано та перевірено за допомогою нової багатообіцяючої програми класифікації тексту в інфраструктурі академічних досліджень Угорщини, MTA Cloud на основі OpenStack. У статті пояснюється концепція, застосовані компоненти та ілюструється їх використання за допомогою реальних вимірювань у випадку використання.

У статі (Chaudharil, Patil, Ghorpade, 2020) розглянуті різні дисципліни мають які мають справу з великими даними, які включають велику кількість функцій. Зібрані дані можуть аналізуються для виявлення знань і використовуються для прийняття рішень. Для того щоб отримати необхідні знання використовують різні алгоритми машинного навчання. В роботі доказано тей факт що для отримання чудових результатів класифікації потрібно об'єднати методи навчання та інструментальні засоби класифікації . В роботі досліджені існуючи методи інтелектуального аналізу даних, які найбільш широко використовуються для класифікації та кластеризації даних, такі як k-Means, Support Vector Machine, Naive Baye's і k-Nearest Neighbor разом із Map Reduce, Apache Spark. Інструменти аналізу даних Map Reduce, Apache Spark, застосований до класифікації або кластеризації даних, забезпечує кращу продуктивність. У цій статі представлено екосистему Hadoop та

дослідження різні методи класифікації та кластеризації за допомогою Map Reduce і Apache Spark.

У роботі (Gopalani, Arora, 2015) обговорюється два варіанти порівняння – Hadoop Map Reduce і нещодавно представлений Apache Spark – обидва з яких забезпечують модель обробки для аналізу великих даних. Незважаючи на те, що обидва ці варіанти базуються на концепції великих даних, їх продуктивність суттєво відрізняється залежно від варіанту використання, який реалізується. Ось що робить ці два варіанти гідними аналізу з огляду на їхню мінливість і різноманітність у динамічному полі великих даних. У цій статті було досліджено ці дві системи разом із аналізом продуктивності за допомогою стандартного алгоритму машинного навчання для кластеризації (K-Means).

У статі (Maillo, Ramírez, Triguero, Herrera, 2017) досліджується k-Nearest Neighbors класифікатор, який є простим, але ефективним широко відомим методом інтелектуального аналізу даних. Фактичне застосування цієї моделі в області великих даних є неможливим через обмеження часу та пам'яті. У роботі запропоновано кілька розподілених альтернатив на основі MapReduce, щоб цей метод міг обробляти великомасштабні дані. Однак їх продуктивність можна ще більше покращити за допомогою нових конструкцій, які відповідають новим технологіям. В цій статі пропонується метод який на основі MapReduce використовувати k-Nearest Neighbors класифікатор найбільш ефективно.

Мета статті – провести аналіз існуючих методів класифікації текстових документів та розробити модель та метод класифікації текстових документів з використанням технології MapRaduce.

Виклад основного матеріалу. В класичному вигляді система класифікації працює наступним чином на сервер потрапляє безліч документів. Кожен документ є членом колекції документів D_i є членом можливих категорій S_j, таких як $\{s_1,s_2,\ldots s_j\}$, тоді класифікація тексту є операцією зіставлення кожного документа з одним або декількома класами. У роботі досліджена класифікація яка виконана на максимум п'ять категорій. Категорії освіта, спорт, культура, політика та світ. Типові кроки для застосування алгоритмів машинного навчання на основі текстових даних показано на рис. 1. Ті самі кроки використовувались і в програмі класифікації.

Система класифікації буде відносити документ який поступає на сервер до певного класу S_j. Для того щоб віднести документ до певного класу система повинна мати інформацію про документ а саме інформацію про ключові слова що входять в документ. Виходячи з цього під системою класифікації розуміється система яка відносить D_i до певних класів S_j. При прийнятті рішень про віднесення документа до певного класу система повинна робити на підставі якоїсь інформації. Ця інформація у теорії розпізнавання образів і називається вектором ознак. В якості таких ознак саме і розглядається саме факт входження певного набору ключових термінів у документ $V=\{v_1, v_2, \ldots, v_m\}$. Таким чином вектором ознак документа D_i буде $x^i=\{x_1,x_2,\ldots x_n\}$ де x приймає значення 0 якщо x_n термін не входить до документа та 1 якщо входить. Таким чином завданням системи класифікації на підставі класифікації попер D_{i-1} документів віднести D_i до одного чи декількох класів. Розглянемо роботу системи класифікації. Система

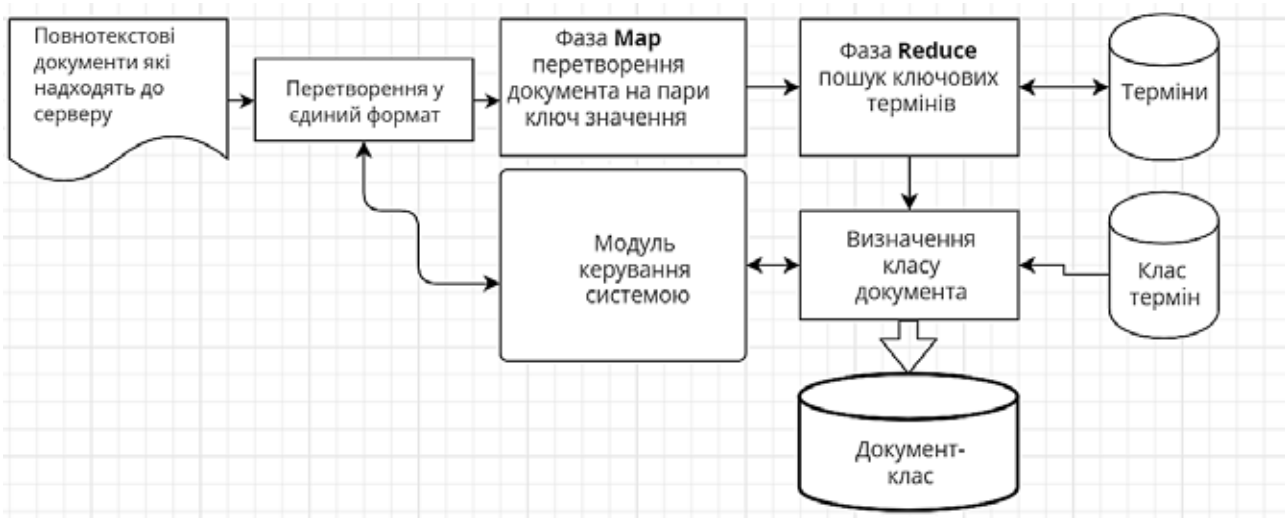

Рис. 1. Модель системи класифікації

класифікації має віднести до певного класу. Якщо система вірно віднесла документ до певного класу це вважається успіхом в зворотному випадку програшом системи, чи помилкою системи. У статичній теорії розпізнавання образів найбільш поширеним є байесовський класифікатор (Gonzalez, Thomason, 1974) який засновано на мінімізації середніх програшів системи класифікації. Під програшом системи будемо вважати невірну класифікацію документу. Будемо вважати що система віднесе документ до класу S_j з вірогідністю $P(S_j)$. Стратегією системи класифікації буде вказівка класу S_j до якого вона відносить даний документ. Зазначена гра характеризуватиметься матрицею втрат L_{il}. Припустимо, що на фхід системи класифікації потрапляють незалежні документи D_i, керуючись ймовірностями P(S_j). Тоді математичне очікування втрат, пов'язаних із віднесенням системою образу x^j до класу C_i, визначиться формулою

$$r_i\left(x^j\right) = \sum_{i=1}^{n} L_{il} P(S_i \mid x^j) . \qquad (1)$$

В цій формулі $P(S_j \mid x^j)$ – умовна ймовірність віднесення документа до класу класу S_j, коли на вхід системи було подано образ x^j.

Байєсовським називається класифікаційне правило, яке мінімізує $r_i(x^j)$ (Gonzalez, Thomason, 1974).

Найчастіше природно припустити, що за правильної класифікації рівні нулю, а інакше рівні одиниці, тобто.

$$L_{il} = 1 - \delta_{il} \qquad (2)$$

де δ_{il} – символи Кронекера.

Для матриці втрат (2) байєсовське класифікаційне правило визначатиметься за допомогою функції вигляду:

$$d_i(x^j) = P(x^j|S_i)P(S_i). \qquad (3)$$

Якщо компоненти випадкового вектора $x^j = \{x_1, x_2, \ldots, x_m\}^j$ незалежні, умовна ймовірність визначається виразом

$$P\left(x^{(j)} \mid S_i\right) = \prod_{k=1}^{m} \gamma_{ik} , \qquad (4)$$

де

$$\gamma_{ik} = \begin{cases} p_{ik}, \text{если } x_k = 1 \\ 1 - p_{ik}, \text{если } x_k = 0. \end{cases} \qquad (5)$$

Тут p_{ik} – ймовірність появи k-го терміна зі словника у документі, що відноситься до класу S_i.

Далі розглянемо алгоритм роботи системи.

Крок 1. На вхід системи потрапляє документ, система на фазі Map робить лексичний розбір документу.

Для цього користуємось додатком Zeppelin. Для прикладу у папку tmp серверу Hadoop розміщаємо документ Doc1.txt, який потребує класифікації.

Далі читаємо документ. Перетворюємо документ на пари ключ значення, та підраховуємо входження термінів.

```
%spark2.pyspark
lines = spark.sparkContext.textFile("/tmp/doc1.txt")
words = lines.flatMap(lambda line: line.split(" "))
# Видаляємо порожні слова.
wordsFiltered = words.filter(lambda w: len(w) > 0)
# підраховуємо кількість слов в документі.
wordc = wordsFiltered.count()
```

Після текстового розбору документа потрібно відшукати значущі слова. Для оцінки значущості на практиці використовують методи, які враховують частоту народження термінів і характеристики, що відображають деякі структурні властивості тексту, наприклад, спільну частоту народження (асоційованість) і щільність розподілу термінів у тексті (надфразові властивості). Використовуючи різні математичні моделі, всі ці методи мають загальне обґрунтування в рамках нейропсихологічної моделі «грубою» обробки інформації у правій півкулі мозку. Перевагою такого підходу є алгоритмічна простота, яка не вимагає точного лінгвістичного аналізу.

Статистичний аналіз тексту використовується на вирішення завдання виділення ключових слів довільного документа. В усіх документах можна назвати статистичні закономірності. Внутрішня структура тексту описується законами Дж. Зіпфа (Zipf, 1949).

Перший закон Зіпфа «ранг-частота» говорить про те, що якщо виміряти кількість входжень кожного слова в текст і взяти тільки одне значення з кожної групи, що має однакову частоту, після чого розташувати частоти в міру їх зменшення та пронумерувати (порядковий номер частоти називається рангом частоти), то слова, що найчастіше зустрічаються, матимуть ранг 1, наступні за ними – 2 і т. д. Імовірність появи довільно вибраного слова P у тексті дорівнюватиме відношенню кількості входжень цього слова K до загального числа слів у тексті N.

$$P = \frac{K}{C} . \qquad (6)$$

Дж. Зіпф виявив таку закономірність: добуток ймовірності P виявлення слова на ранг частоти R дорівнює константі C.

$$C = P \times R. \qquad (7)$$

За першим законом Зіпфа, якщо найпоширеніше слово зустрічається у мові K разів,

то наступне за частотою слово зустрічається з часткою ймовірності раз. Значення константи *C* у різних мовах по-різному, але всередині однієї мовної групи залишається незмінним (Zipf, 1949).

Другий закон Зіпфа «кількість-частота» говорить про те, що різні слова можуть входити до тексту з однаковою частотою, причому частота та кількість слів, що входять до тексту із цією частотою, теж пов'язані між собою. Відносини частоти входження слова до рангу слів представлені рис. 2 (Zipf, 1949).

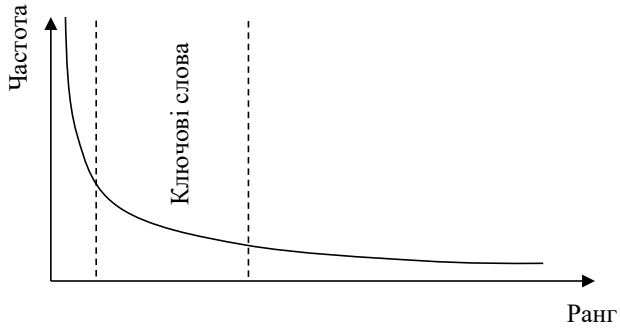

Рис. 2. Відношення частоти входження слова до рангу слів

З наведеного на рис. 2 графіка ми бачимо, що значні слова лежать у середній частині діаграми. Слова, які зустрічаються як занадто часто в тексті (в основному вони виявляються прийменниками та займенниками), так і дуже рідко, в більшості випадків не мають вирішального значення (Zipf, 1949).

Крок 2. На фазі Reduce підраховуємо кількість входження кожного слова в документ. Та за формулою (7) підраховуємо ранг слова.

Для цього знаходимо найпопулярнішими, виконавши підрахунок слів за допомогою перетворень map() і reduceByKey() для створення кортежів типу (word, count).

```
wordCounts = wordsFiltered.map(lambda word: (word, 1)).
reduceByKey(lambda a,b: a+b)
```

Далі перетворюємо пари ключ значення на дата фрейм. Та знаходимо значимі слова.

```
wordsCounts = (filteredWordCounts.map(lambda (w, c):
        Row(word=w, rang =c)) .toDF())
```

Ми можемо для прикладу переглянути значущі слова які виявила система.

```
wordsCounts.show()
```

```
+----+-------------+
|rang|         word|
+----+-------------+
|   1|     colleges|
|   1| particularly|
|   1|institutional|
|   1|   professors|
|   1| multitasking|
|   1|      through|
|   1|         cost|
|   1|   multitask |
|   1|        speed|
|   1|relationships|
|   1|        group|
|   1|         work|
```

Рис. 3. Ранг слів що входять до документу

Далі на підставі вибраних з документу значущих термінів система за допомогою байесовського правила формула (3) приймає рішення саме до якого класу віднести цей документ. На рисунку 3 наведено входження ключових слів у документ. Як ми бачимо всі ключові слова мають ранг один так як документ взятий для прикладу був не великим.

Висновки із цього дослідження і перспективи подальших розвідок у цьому напрямку. У роботі запропоноване нове рішення для виконання точної байєсовської класифікації на основі Spark. Цей класифікатор використовує велику кількість операції в пам'яті сервера, щоб класифікувати велику кількість випадків текстових документів на основі великого навчального набору даних з використанням MapReduce. Фаза карти обчислює кількість входжень ключових слів у різних розподілах даних навчання. Після цього кілька редукторів обчислюють вірогідність віднесення документу до певних класів, на підставі обчислень отриманих на етапі карти. Ключовий момент цієї пропозиції полягає в управлінні набором текстових документів, зберігаючи їх в пам'яті, коли це можливо. В іншому випадку він розбивається на мінімальну кількість частин, застосовуючи MapReduce для кожної частини, використовуючи навички кешування Spark для повторного використання попередньо розділеного навчального набору.

ЛІТЕРАТУРА:

1. Gonzalez R.C., Thomason M G. Tree Grammars and Their Application to Pattern Recognition. Tech. Rep. TR-EE/CS-74-10, Electrical Engineering Dept., Univ. of Tennessee, Knoxville. 1974. P. 364.
2. Gonzalez R C., Thomason M.G. Inference of Tree Grammars for Syntactic Pattern Recognition. Tech. Rept. TR-EE/CS-74-20, Electrical Engineering Dept., University of Tennessee, Knoxville. 1974. P. 160.
3. Gonzalez R.C., Tou J.T. Some Results in Minimum-Entropy FeaturExtraction. IEEE Convention Record. Region III. 1968.
4. Salton G. Another look at automatic text-retrieval systems. *Commun. ACM.* 1986. № 7. P. 648–656. 2000. ISBN 951-22-5145-0
5. Semberecki P., Maciejewski H. Distributed Classification of Text Documents on Apache Spark Platform. International Conference on Artificial Intelligence and Soft Computing. June 2016. P. 621–629. DOI:10.1007/978-3-319-39378-0_53 [Scopus].
6. I. Pintye, E. Kail, P. Kacsuk, R. Lovas. Big data and machine learning framework for clouds and its usage for text classification. Volume 33. Issue 19. Special Issue: Human oriented solutions for intelligent analysis, multimedia and communication systems (Human Oriented Solutions 2020). Science Gateways Special Issue (Science Gateways 2020) 10 October 2021. https://doi.org/10.1002/cpe.6164.
7. Ratna S. Chaudhari1 , Seema S. Patil , Smita J. Ghorpade. Classification and clustering methods along with Map Reduce, Apache Spark: a study. *IJRAR.* November 2020. Volume 7. Issue 4.
8. Gopalani S., Arora R. Comparing Apache Spark and Map Reduce with Performance Analysis using K-Means. *March 2015 International Journal of Computer Applications.* 113(1). P. 8–11. DOI:10.5120/19788-0531
9. Maillo J., Ramírez S., Triguero I., Herrera F. kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data. June 2016. *Knowledge-Based Systems.* 1 February 2017. Volume 117. P. 3–15. DOI:10.1016/j.knosys.2016.06.012
10. Zipf G.K. Human Behavior and the Principle of Least Effort. Cambridge, 1949. P. ix, 3, 5–8.

REFERENCES:

1. Gonzalez R.C., Thomason M G. Tree Grammars and Their Application to Pattern Recognition. Tech. Rep. TR-EE/CS-74-10, Electrical Engineering Dept., Univ. of Tennessee, Knoxville. 1974. P. 364.
2. Gonzalez R C., Thomason M.G. Inference of Tree Grammars for Syntactic Pattern Recognition. Tech. Rept. TR-EE/CS-74-20, Electrical Engineering Dept., University of Tennessee, Knoxville. 1974. P. 160.
3. Gonzalez R.C., Tou J.T. Some Results in Minimum-Entropy FeaturExtraction. IEEE Convention Record. Region III. 1968.
4. Salton G. Another look at automatic text-retrieval systems. *Commun. ACM.* 1986. № 7. P. 648–656. 2000. ISBN 951-22-5145-0
5. P. Semberecki, H. Maciejewski. Distributed Classification of Text Documents on Apache Spark Platform. International Conference on Artificial Intelligence and Soft Computing. June 2016. P. 621–629. DOI:10.1007/978-3-319-39378-0_53 [Scopus].
6. PintyeI. , Kail E., Kacsuk P., Lovas R. Big data and machine learning framework for clouds and its usage for text classification. Volume 33. Issue 19. Special Issue: Human oriented solutions for intelligent analysis, multimedia and communication systems (Human Oriented Solutions 2020). Science Gateways Special Issue (Science Gateways 2020) 10 October 2021. https://doi.org/10.1002/cpe.6164.
7. Ratna S. Chaudhari1 , Seema S. Patil , Smita J. Ghorpade. Classification and clustering methods along with Map Reduce, Apache Spark: a study // IJRAR November 2020, Volume 7, Issue 4.
8. Gopalani S., Arora R. Comparing Apache Spark and Map Reduce with Performance Analysis using K-Means. March 2015 International Journal of Computer Applications 113(1). P. 8–11. DOI:10.5120/19788-0531
9. Maillo J., Ramírez S., Triguero I., Herrera F. kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data. June 2016. Knowledge-Based Systems Volume 117, 1 February 2017, Pages 3–15. OI:10.1016/j.knosys.2016.06.012
10. Zipf G.K. Human Behavior and the Principle of Least Effort. Cambridge, 1949. P. ix, 3, 5–8.