**УДК: 004.832.32.**

***Лисенко С. М.***

# РОЗРОБКА ТА ДОСЛІДЖЕННЯ АЛГОРИТМІВ ІНТЕЛЕКТУАЛЬНОЇ ОБРОБКИ ТЕКСТІВ (TEXT MINING)

*У процесі дослідження були розроблені модифікації алгоритмів інтелектуальної обробки текстів, які дозволять здійснювати класифікацію та кластеризацію кириличних текстів для застосування у різних галузях, пов'язаних з інтернет-ресурсами. Використання модифікованих алгоритмів надало нові можливості, пов'язані з обробкою та використанням кириличних, а не латинських текстів. Розроблено інформаційно-аналітичну систему для аналізу кириличних текстів, яка дозволяє отфільтровувати спам, відносити документи до певної рубрики виконувати кластерізацію документів*

***Ключові слова:*** *Text Mining, алгоритми інтеллектуальної обробки текстів, кластеризація, класифікація, визначення спаму*

*Разработаны модификации алгоритмов интеллектуальной обработки текстов, позволяющие выполнять классификацию и кластеризацию кириличних текстов применительно к интернет-ресурсам для разных областей. Разработана информационно-аналитическая система для анализа кириличних текстов, позволяющая распознавать спам, относить документы до определенной рубрики, выполнять кластеризацию документов*

***Ключевые слова:*** *Text Mining, алгоритм интелектуальної обработки текстов, кластеризация, классификация, определение спама.*

*The text mining algorithms were developed for classification and clustering Cyrillic texts to use in various fields related to Internet resources. Using the modified algorithms presenting new opportunities associated with the processing and use Cyrillic instead of Latin texts.*

***Key words:*** *Text Mining, algorithm of intelligent text manipulation, clusterization, classification, determination of spam.*

**Вступ**

Електронна інформація грає все більшу роль в усіх сферах життя сучасного суспільства. У інформаційних сховищах, розподілених по всьому світу, зібрані терабайти текстових даних. І, як ми знаємо, розвиток інформаційних ресурсів Інтернет багаторазово посилив проблему інформаційного перевантаження. Технологія ефективного аналізу тексту Text Mining здатна виступити в ролі репетитора, який, перечитавши увесь курс, викладає лише найбільш ключову і значущу інформацію. Таким чином, користувачеві не обов'язково самому «просіювати» величезну кількість неструктурованої інформації. Розроблені на основі статистичного і лінгвістичного аналізу, а також штучного інтелекту, технології Text Mining якраз і призначені для проведення смислового аналізу, забезпечення навігації і пошуку в неструктурованих текстах.

Можливості текстової обробки відомих пакетів таких як: Intelligent Miner for Text, TextAnalyst, WebAnalyst, Text Miner, SemioMap, Oracle Text, Knowledge Server, Galaktika-ZOOM, на російській мові дуже обмежена або відсутня взагалі. Існують російські пакети (додатки до Oracle), такі як Russian Context Optimizer, але вони є комерційними продуктами. Саме тому розробка власної українсько- або російсько-мовної інформаційно-аналітичної системи придатної для інтелектуального аналізу кириличних текстів є актуальною задачею.

**Задачі дослідження**

Задачею дослідження є вибір та реалізація методів інтелектуальної обробки кириличних текстів.

Для цього необхідно:

– провести аналіз алгоритмів та методів data mining які придатні для роботи з текстами;

– провести розробку та модифікацію алгоритмів кластеризації, класифікації для інтелектуальної обробки кириличних текстів;

– створити інформаційно-аналітичну систему, яка поєднує базу даних або необхідної інформації з системою логічного висновку яка реалізує розроблені алгоритми.

Вхідними даними, тут виступають бізнес-проекти інтернет-організації, яка створена для роботи та різнопланові листи, які надходять до організації з Інтернету.

**Класифікація текстових документів**

Класифікація текстових документів, так само як і у випадку класифікації об'єктів, полягає у віднесенні документа до одного із заздалегідь відомих класів. Часто класифікацію стосовно текстових документів називають категоризацією або рубрикацією. Більшість методів класифікації текстів так чи інакше засновані на припущенні, що документи, що відносяться до однієї категорії, містять однакові ознаки (слова або словосполучення), і наявність або відсутність таких ознак в документі говорить про його приналежність або неприналежність до тієї або іншої теми.

Таким чином, для кожної категорії повинна бути множина ознак:

$$F(C) = \cup F(c_r), \tag{1}$$

*де* $F(c_r) = \{f_1,...,f_k,...f_z\}$.

Таку множину ознак часто називають словником, оскільки воно складається з лексем, які включають слова і/або словосполучення, що характеризують категорію.

Подібно до категорій кожен документ також має ознаки, по яким його можна віднести з деякою мірою вірогідності до однієї або декільком категоріям:

$$F(d_i) = \{f_1^{\,i},...,f_k^{\,i},...f_z^{\,i}\}. \tag{2}$$

Множина ознак всіх документів повинна співпадати з множиною ознак категорій, тобто:

$$F(C) = F(D) \cup F(d_i). \tag{3}$$

Необхідно відмітити, що дані набори ознак є відмітною межею класифікації текстових документів від класифікації об'єктів в Data Mining, які характеризуються набором атрибутів.

Рішення про віднесення документа до категорії, ухвалюється на підставі перетину:

$$F(d_i) = \cup F(c_r). \tag{4}$$

Завдання методів класифікації полягає в тому, щоб найкращим чином обрати такі ознаки і сформулювати правила, на основі яких ухвалюватиметься рішення про віднесення документа до рубрики.

Існує два протилежні підходи до формування множини $F(C)$ і побудови правил:

а) машинне навчання, де передбачається наявність повчальної вибірки документів, але якому будується множина $F(C)$;

б) експертний метод, який припускає, що виділення ознак – множини $F(C)$ – і складання правил проводиться експертами.

У разі машинного навчання аналізується статистика лінгвістичних шаблонів (таких як лексична близькість, повторюваність слів і т. п.) з документів повчальної вибірки. У неї повинні входити документи, які відносяться до кожної рубрики, щоб створити набір ознак (статистичну сигнатуру) для кожної рубрики, який згодом використовуватиметься для класифікації нових документів. Гідністю даною підходу є відсутність необхідності в словниках, які складно побудувати для великих наочних областей. Проте щоб уникнути невірної класифікації, потрібно забезпечити хороше представлення документів для кожної рубрики.

У другому випадку формування словника (множини $F(C)$) може бути виконане на основі набору термінів наочної області і відносин між ними (основні терміни, синоніми і споріднені терміни). Класифікація може потім визначити рубрику документа відповідно до частоти, з якою з'являються виділені в тексті терміни (ключові поняття).

Можлива і комбінація двох описаних підходів, коли виділення ознак і складання правил виконуються автоматично на основі повчальної вибірки, і в той же час правила будуються у такому вигляді, щоб експертові була зрозуміла логіка автоматичної рубрикації, і у нього була можливість уручну коректувати ці правила.

Для класифікації текстових документів успішно використовуються багато методів і алгоритми класифікації Data Mining: Naive Bayes, метод найменших квадратів, C4.5, SVM та ін.

Очевидно, що потрібна модифікація цих методів для роботи з текстовою інформацією. Як правило, адаптація алгоритмів пов'язана з тим, що поняття незалежної змінної пов'язане не з атрибутами об'єкту, а з наявністю в текстовому документі тієї або іншої ознаки.

В роботі реалізовано наївний байєсівського класифікатор до класифікації електронних листів на два класи – спам ($S$) та не-спам ($\overline{S}$) та метод Роше для віднесення листа до певної категорії.

**Метод Naive Bayes** припускає обчислення вірогідності приналежності текстового документа до кожної рубрики. Рішення про приналежність приймається по максимальній вірогідності.

Властивості наївної класифікації: використання всіх змінних і визначення всіх залежностей між ними; наявність двох припущень відносно змінних; всі змінні є однаково важливими; всі змінні є статистично незалежними, тобто значення одній змінній нічого не говорить про значення іншої; більшість інших методів класифікації припускають, що перед початком класифікації вірогідність того, що об'єкт належить тому або іншому класу однакова, але це не завжди вірно. Вважатимемо, що документи вибрані з декількох класів документів, які можуть бути представлені безліччю слів з незалежною вірогідністю, що і-те слово даного документа зустрічається в документі класу C: $p(\omega_i \mid C)$ (для цього завдання припустимо, що вірогідність зустрічі слова в документі незалежна від довжини документа і всі документи мають однакову довжину).

Тоді вірогідність для даного документа D і класу C:

$$p(D \mid C) = \prod_i p(\omega_i \mid C). \tag{5}$$

Питання, на яке ми хочемо відповісти: «яка вірогідність того, що даний документ *D* належить класу *C*?».

За теоремою Байєса:

$$p(C \mid D) = \frac{p(C)}{p(D)} p(D \mid C) \tag{6}$$

Припустимо, що ми маємо тільки два класи: $S$ і $\overline{S}$ (напр. спам и не-спам). Тоді:

$$p(\overline{S} \mid D) = \frac{p(\overline{S})}{p(D)} \prod_i p(\omega_i \mid \overline{S}). \tag{7}$$

Поділивши одне на інше отримаємо відношення правдоподібності:

$$\frac{p(S \mid D)}{p(\overline{S} \mid D)} = \frac{p(S)}{p(\overline{S})} \prod_i \frac{p(\omega_i \mid S)}{p(\omega_i \mid \overline{S})}, \tag{8}$$

або (для логарифма правдоподібності):

$$\ln \frac{p(S \mid D)}{p(\overline{S} \mid D)} = \ln \frac{p(S)}{p(\overline{S})} + \sum_i \ln \frac{p(\omega_i \mid S)}{p(\omega_i \mid \overline{S})}. \tag{9}$$

Дійсна вірогідність $p(S \mid D)$ може бути порахована з $\ln \frac{p(S \mid D)}{p(\overline{S} \mid D)}$ ґрунтуючись на спостереженні, що:

$$p(S \mid D) + p(\overline{S} \mid D) = 1. \tag{10}$$

Нарешті, документ може бути класифікований порівнянням логарифма правдоподібності з деяким порогом $h$ (наприклад $h = 0$). Перед нами спам, якщо:

$$\ln \frac{p(S \mid D)}{p(\overline{S} \mid D)} > h. \tag{11}$$

Загальний опис алгоритму:

1) Занесення вхідних даних (листа) до масивів алгоритму.

2) Створення двох класів для класифікації – Спам $S$ та Не-спам $\overline{S}$.

3) Отримання апріорної вірогідності для кожного класу $p(S \mid D)$, $C$ – клас.

4) Знаходження вірогідності приналежності листа до певного класу $p(D \mid C) = \prod_i p(\omega_i \mid C)$.

5) Знаходження відношення правдоподібності $\ln \frac{p(S \mid D)}{p(\overline{S} \mid D)}$.

6) Визначення класу документу – спам або не спам та запис до відповідної таблиці.

**Метод Роше (Rocchio method)**

Деякі класифікатори використовують так званий профайл (profile, прототип документа) для визначення категорії. Профайл – це список зважених термів, присутність (або відсутність) яких дозволяє найточніше відрізняти конкретну категорію від інших категорій. Метод, запропонований Дж. Роччио (J. Rocchio), відноситься до лінійних класифікаторів, в яких кожен документ представляється у вигляді вектора вагових значень термів. Профайл категорії розглядатимемо як вектор $\vec{c}^{(i)} = \{c_1^{(i)}, ..., c_N^{(i)}\}$ ($N$ – кількість термів у словнику), значення елементів якого $c_k^{(i)}$ при навчанні класифікатора в рамках методу Rocchio розраховується по формулі:

$$c_k^{(i)} = \frac{\alpha}{|POS_i|} \sum_{d^{(j)} \in POS_i} w_k^{(j)} - \frac{\beta}{|NEG_i|} \sum_{d^{(j)} \in NEG_i} w_k^{(j)}, \tag{12}$$

де $w_k^{(j)}$ – це вага терму $t_k$ у документі $d^{(j)}$ (розрахований, наприклад, за прин ципом *TFIDF*);

$POS_i$ – це позитивний приклад – безліч документів, що належать категорії $\vec{c}^{(i)}$, тобто

$$POS_i = \{d^{(j)} \mid \Phi(d^{(j)}, c_i) = 1\};$$

$NEG_i$ – негативний приклад – множина документів, що не належать категорії $\vec{c}^{(i)}$:

$$NEG_i = \{d^{(j)} \mid \Phi(d^{(j)}, c_i) = 0\}.$$

У цій формулі, $\alpha$ и $\beta$ – контрольні параметри, які характеризують значущість позитивних і негативних прикладів. Наприклад, якщо $\alpha = 1$ и $\beta = 0$, $c_i$ буде «центром мас» всіх документів, що відносяться до відповідної категорії.

Функція $CSV^{(i)}d$ в цьому випадку визначається по різних методиках, або як величина зворотна відстані від вектора вагових значень термів, що входять в документ $d$, до профайла категорії $c_i$, або як скалярний добуток цих векторів.

Основна особливість цього методу полягає в тому, що для кожної рубрики обчислюється зважений центроїд. Він виходить відніманням ваги кожного терму векторів ознак не відповідних рубриці документів, з вагів термів векторів ознак відповідних рубриці документів.

Хай кожен документ рубрики буде представлений у вигляді вектора ознак таким чином. Тоді рубрика буде представлена у вигляді вектора ознак. Для кожної рубрики обчислюється зважений центроїд.

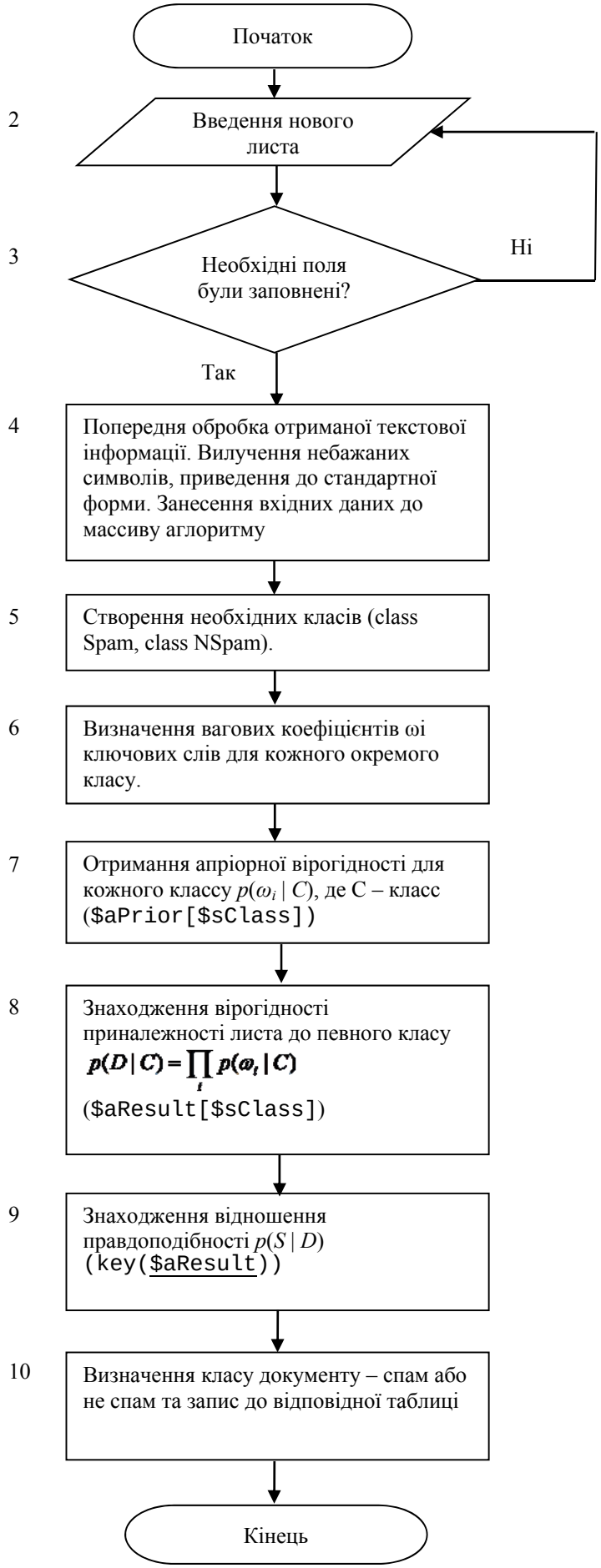


**Рис. 1.** Блок-схема алгоритму класифікації Naive Bayes

Таким чином, зважений центроїд представляє рубрику в просторі ознак. Приналежність рубрикам невідомого документа, визначається шляхом обчислення відстані між центроїдом кожної з рубрик і вектором документа, що класифікується. Якщо відстань не перевершує деякого, заздалегідь заданого порогу, документ вважається таким, що належить даній рубриці.

Практичне дослідження методу Роше показали, що даний метод володіє високою ефективністю в рішенні задачі класифікації текстів. Однією з головних його особливостей є можливість змінювати вектор зваженого центроїда рубрики, без перенавчання класифікатора. Ця властивість може бути корисною, наприклад, у випадках, коли повчальна колекція часто поповнюється новими документами, а перенавчання займає дуже багато часу. Завдяки своїй результативності і простоті метод Роше став одним з найпопулярніших в області класифікації текстових документів і часто використовується як базовий, для порівняння ефективності різних класифікаторів. Класифікатор Роше був застосований для визначення направленості листа, тому що однією з головних його особливостей є можливість змінювати вектор зваженого центроїда характеристики без перенавчання класифікатора, що корисно в нашому випадку – коли повчальна колекція часто поповнюється новими листами, а перенавчання займає дуже багато часу.

Визначення направленості письма (позитивна чи негативна), визначається шляхом обчислення відстані між центроїдом кожної з характеристик – позитивної та негативної і вектором листа, що класифікується. Якщо відстань не перевершує деякого, заздалегідь заданого порогу, лист вважається позитивним.

**Загальний опис алгоритму**

1. Занесення вхідних даних (листів та файлу позитивних наборів) до масивів алгоритму.
2. Формування навчальної вибірки позитивних значень – POS.
3. Формування вектору документа $d$, що класифікується
4. Обчислення ваги документа $w_k$ та центроїду вибірки позитивних значень $c_k$ на кожному кроці $k$.
5. Обчислення центроїду профайлу документа $\vec{c}$ .
6. Знаходження відстані між центроїдом документа та центроїдом кожного елементу з вибірки позитивних значень.
7. Підрахунок кількості позитивних відстаней.
8. Підрахунок кількості негативних відстаней як різниці між загальною кількістю та кількістю позитивних значень.

Блок-схема алгоритму класифікації Роше наведена на рис 2.

**Методи кластеризації текстових документів**

***Представлення текстових документів***

Більшість алгоритмів кластеризації потребує, щоб дані були представлені у вигляді моделі векторного простору (vector space model). Це найбільш розповсюджена модель для інформаційного пошуку. Вона концептуально проста та використовує метафору для відображення семантичної подібності як просторової близькості.

В цій моделі кожен документ уявляється в багатомірному просторі, в якому кожен вимір відповідає слову в наборі документів.

Ця модель представляє документи матрицею слів і документів:

$$M = | F | x | D |, \tag{13}$$

де $F = \{f_1,...,f_k,...f_z\}$, $D = \{d_1,...,d_i,...d_n\}$ де $d_i$– вектор в 2-мірному просторі $R^z$.

Набір ознак $F$ конструюється за допомогою виключення рідкісних слів і слів з високою частотою. Виключення слів означає, що слова розглядаються тільки як ознаки, якщо вони зустрічаються більшу кількість разів, ніж зазначений частий поріг, або меншу кількість разів, чим позначений нечастий поріг. Значення порогів визначаються експериментально.

Кожній ознаці $f_k$ в документі $d_i$, ставиться у відповідність його вага $\omega_{k,i}$, яка позначає важливість цієї ознаки для даного документа. Для обчислення ваги можуть використовуватися різні підходи, наприклад алгоритм TFIDF (Term Frequency Inverse Document Frequency). Ідея цього підходу – гарантувати, що вага ознаки знаходитиметься в діапазоні від 0 до 1. При цьому чим частіше слово з'являється в тексті, тим його вага вища, і навпаки: чим частота менша, тим вага менша. Формула, по якій обчислюється вага, має наступний вигляд:

$$\omega_{k,i} = \frac{(1+\log(N_{i,k}))(\log(|D|/N_k))}{\sqrt{\sum_{s\neq k}\left(\log(N_{i,s}+1)\right)^2}}, \tag{14}$$

де $N_{i,k}$ – кількість появ ознаки $f_k$ в документі $d_i$;

$N_k$ – кількість появ ознаки $f_k$ у всіх документах множини $D$;

$|D|$ – кількість документів (потужність множини D).

Необхідно відзначити, що в знаменнику знаходиться сума по всім документам, окрім того, що розглядається. Таким чином, вага функції нормалізується щодо всіх документів. Ця модель часто називається «Мішок слів» (bag-of-words).

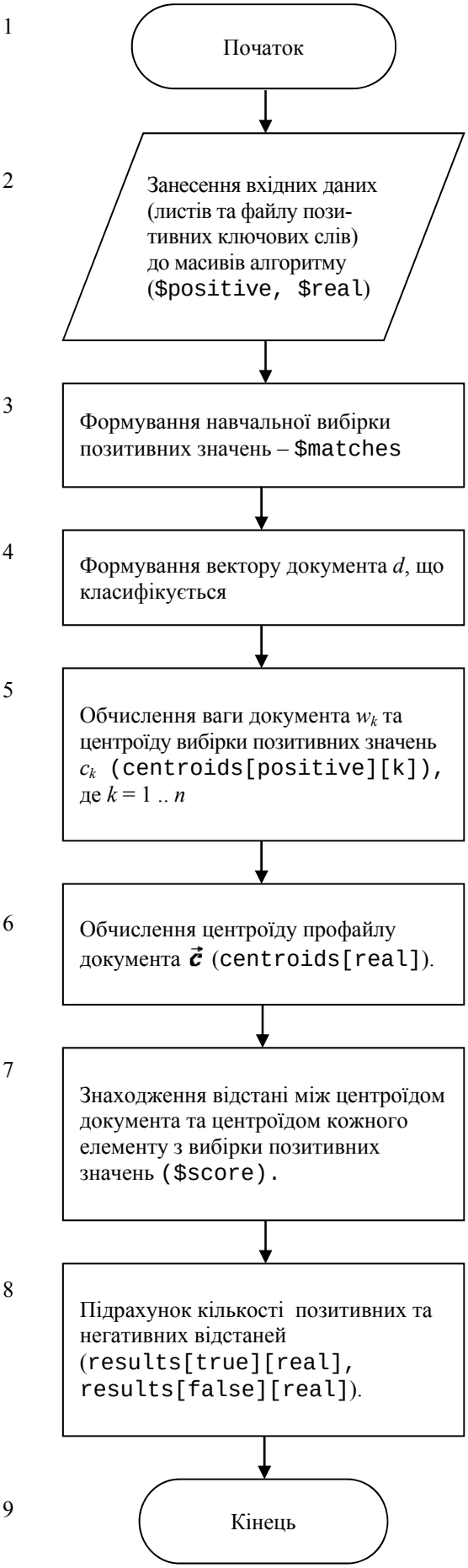


**Рис. 2.** Блок-схема алгоритму класифікації Роше

Окрім методу TFIDF для зважування термів часто використовується підхід TLTF (Term Length Term Frequency). Ідея методу TLTF базується на тому, що слова, які з'являються часто, прагнуть бути короткими. Такі слова не описують основну тему документа, тобто є стоп-словами. Навпаки, слова, які з'являються рідко, прагнуть бути довгими.

Кластери в даній моделі представляються аналогічно документам у вигляді векторів:

$$C = \{c_1,...,c_j,...c_m\},$$

де $c_j$ – вектор в 2-мірному просторі $R^z$. Вектор $c_j$, часто є центром кластера (центроїдом).

При цьому метою кластеризації є угрупування документів (представлених векторами) по кластерах відповідно до близькості їх до центрів. Близькість документа і кластера, представлених просторовими векторами, обчислюється як кут між цими векторами:

$$\cos(\vec{d}_i,\vec{c}_j) = \frac{\vec{d}_i \cdot \vec{c}_j}{\left|\vec{d}_i\right| \cdot \left|\vec{c}_j\right|} = \frac{\sum^{|F|} d_{i,k} \cdot c_{j,k}}{\sqrt{\sum^{|F|} {d_{i,k}}^2} \cdot \sqrt{\sum^{|F|} {d_{j,k}}^2}}. \quad (14)$$

Всі алгоритми кластеризації ґрунтуються на вимірюваннях схожості по різних критеріях. Деякі використовують слова, які часто з'являються разом (лексичну близькість), інші використовують вилучені особливості (такі як імена людей і т. п.). Різниця полягає також і в кластерах, що створюються.

Виділяють три основні типи методів кластеризації документів:

а) ієрархічний – створює дерево зі всіма документами в кореневому вузлі і одним документом у вузлі-листі. Проміжні вузли містять різні документи, які стають все більш і більш спеціалізованими у міру наближення до листя дерева. Цей метод корисний, коли досліджують нову колекцію документів і хочуть отримати загальне уявлення про неї;

б) бінарний – забезпечує угрупування і перегляд документальних кластерів по посиланнях подібності. У один кластер поміщаються найближчі по своїх властивостях документи. В процесі кластеризації будується базис посилань від документа до документа, заснований на вагах і сумісному вживанні визначуваних ключових слів;

в) нечіткий – включає кожен документ у всі кластери, але при цьому зв'язує з ним вагову функцію, що визначає ступінь приналежності даного документа до певного кластеру.

***Бінарна кластеризація***

У бінарній кластеризації кожен документ може міститися тільки в одному кластері, а кластери можуть зв'язуватися між собою на основі спільності характеристик, виділених екстрактором. Бінарна кластеризація створює функцію, описану локально (серед документів одного и того ж кластеру) чи глобально (крізь усі документи).

Типовим представником інтерактивних алгоритмів є алгоритм к-середніх. Він інтерактивно виконує ділення даних на к-кластерів, мінімізуючи відстані між елементами кластерів і їх центрами.

Для завдання кластеризації текстових документів він адаптується наступним чином. Є множина документів: $D = \{d_1,...,d_i,...d_n\}$, $d_i \in R^T$.

Алгоритм к-середніх створює $k$ декомпозицій так, щоб якщо $\{\vec{c}_1,\vec{c}_2,...\vec{c}_k\}$ являє $k$ центрів, то мінімізується наступна цільова функція: $j = \arg\min_j \sum_{j=1}^{k} \sum_{d_i \in D} \| \vec{d}_i - c_j \|^2$.

У данної реалізації алгоритму бінарної кластеризації кожен лист може міститися тільки в одному проекті, а проекти можуть зв'язуватися між собою на основі спільності характеристик, виділених екстрактором.

Алгоритм бінарної кластеризації створює функцію, описану локально, яка перевіряє приналежність листа до проекту створюючи кожен раз бінарну пару «лист-проект», визначаючи ваговий коефіцієнт приналежності цієї пари. Після цього перевіряється цільова функція.

***Загальний опис алгоритму***

1) Занесення вхідних даних (листа та проектів) до масивів алгоритму $\vec{d}_i$.
2) Формування бінарної пари «лист-проект» $\vec{c}_j$, $j$ – номер проекту.
3) Знаходження вагового коефіцієнту декомпозиції цієї пари.
4) Перевірка на мінімум цільової функції $\sum_{j=1}^{k} \sum_{d_i \in D} \| \vec{d}_i - c_j \|^2$.
5) Якщо так, то занесення цього листа до відповідного проекту, якщо ні – формування наступної пари.

**Дослідження алгоритму класифікації Naive Bayes**

Для навчання класифікатора були задіяні справжні листи зі спамом та звичайні листи від фірм різного направлення – юридичні, статистичні чи інші. Результати, отримані в процесі тестування системи, наведено в табл. 1.

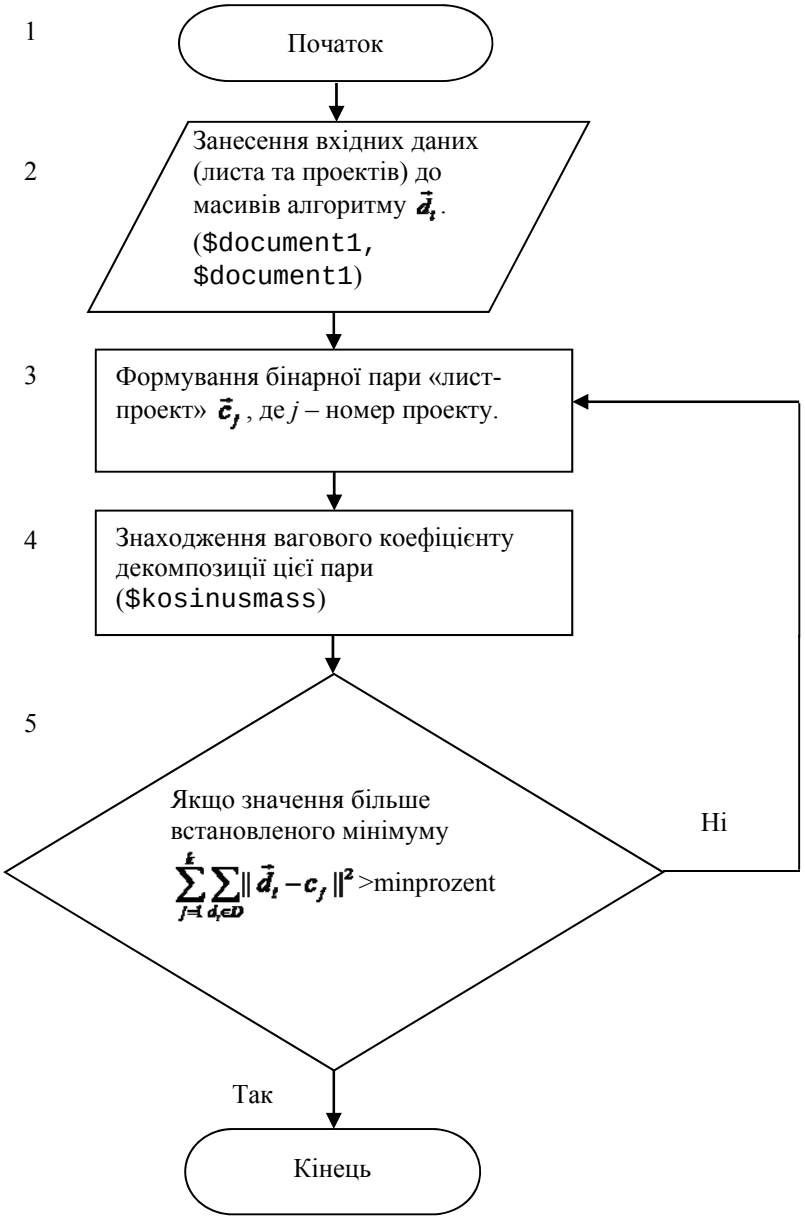


**Рис. 3.** Блок-схема алгоритму бінарної кластеризації

*Таблиця 1*

**Аналіз результатів класифікації Naive Bayes**

| Реальний тип листа | Визначення як спам | Визначення як листи з відгуками |
|---|---|---|
| Листи з інформацією | 4 % | 96 % |
| Листи зі спамом | 87 % | 13 % |

***Дослідження класифікатора Роше***

У таблиці 2 приведені аналіз результатів роботи алгоритму бінарної кластеризації на тестових даних.

*Таблиця 2*

**Аналіз результатів класифікації за алгоритмом Роше**

| Тип листа | Позитивні листи | Негативні листи |
|---|---|---|
| Направленість, визначена класифікатором | 74 % | 26 % |
| Справжня направленість | 68 % | 32 % |

***Дослідження алгоритму бінарної кластеризації***

У бінарній кластеризації кожен лист може міститися тільки в одному проекті, а проекти можуть зв'язуватися між собою на основі спільності характеристик, виділених екстрактором. Агоритм бінарної кластеризації створює функцію, описану локально, яка перевіряє приналежність листа до проекту створюючи кожен раз бінарну пару «лист-проект», визначаючи ваговий коефіцієнт приналежності цієї пари.

*Таблиця 3*

**Аналіз результатів бінарної кластеризації**

| Назва проекту | Листи 1 проекту | Листи 2 проекту | Листи 3 проекту | Листи 4 проекту | Листи 5 проекту |
|---|---|---|---|---|---|
| Сайт для банку | 90 % | 10 % | 0 % | 0 % | 2 % |
| Сайт для університету | 6 % | 88 % | 0 % | 0 % | 0 % |
| Інтернет-кафе «E-eat» | 3 % | 2 % | 93 % | 3 % | 5 % |
| Ресторан «Фараон» | 0 % | 0 % | 1 % | 97 % | 0 % |
| Інтернет-магазин «Компік» | 1 % | 0 % | 6 % | 0 % | 92 % |

**Висновки.** Доведено ефективність методів та алгоритмів інтелектуальної обробки текстової інформації Text Mining. При цьому алгоритми Text Mining можуть застосову-ватися для рішення задач класифікації неструктурованих або слабкоструктурованих документів.

При цьому реалізовано:

– алгоритмічне та програмне забезпечення для класифікації кириличних текстів за методом Байєса (Naive Bayes), у якому визначаються залежності між всіма змінними, що дозволяє легко обробляти ситуації, в яких значення деяких змінних невідомі; також даний класифікатор дозволяє уникнути проблеми перенавчання (overfitting), тобто надмірного ускладнення моделі, що є слабкою стороною багатьох методів (наприклад, дерев рішень і нейронних мереж);

– алгоритмічне та програмне забезпечення для бінарної кластеризації кириличної текстової інформації що забезпечує угрупування і перегляд документальних кластерів по посиланнях подібності. У один кластер поміщаються найближчі по своїх властивостях документи. В процесі кластеризації будується базис посилань від документа до документа, заснований на вагових коефіцієнтах і сумісному вживанні ключових слів;

– розроблено алгоритмічне та програмне забезпечення для класифікатора Роше (Rochio), однією з головних особливостей якого є можливість змінювати вектор зваженого центроїда рубрики, без перенавчання класифікатора. Ця властивість може бути корисною, наприклад, у випадках, коли навчальна колекція часто поповнюється новими документами, а перенавчання займає дуже багато часу;

– розроблено інформаційно-аналітичну систему для аналізу кириличних текстів. Було обрано середовище Zend Studio 8.0 та Adobe Dreamweaver CS5.

## *ЛІТЕРАТУРА*

1. Барсегян А. А. Анализ данных и процессов / А. А. Барсегян, М. С. Куприянов, И. И. Холод, М. Д. Тесс, С. И. Елизаров. – СПб. : БХВ-Петербург, 2009. – 512 с. – ISBN: 978-5-9775-0368-6.
2. A vector space model for automatic indexing / G. Salton, A. Wong, C. Yang // Commun. – ACM, 1995. – Vol. 18, no. 11. – P. 613–620.



*Стаття надійшла до редколегії 11.04.2011 р.*