УДК 004.912

Л.Е. Чала[1], П.Ю. Попаденко[2]

[1, 2] ХНУРЕ, м. Харків, Україна, aspirantura@kture.kharkov.ua

МЕТОД ВИЯВЛЕННЯ НЕЧІТКИХ ДУБЛІКАТІВ ЕЛЕКТРОННИХ ТЕКСТОВИХ ДОКУМЕНТІВ

В роботі здійснено аналіз методів виявлення нечітких дублікатів текстових файлів. Показано, що існуючі методи виявлення нечітких дублікатів спрямовані на мінімізацію обчислювальної складності з одночасним збільшенням повноти і точності алгоритму. У даній роботі запропоновано та протестовано гібридний метод, який дозволяє істотно поліпшити якість виявлення нечітких дублікатів при незначному збільшенні обчислювальної складності базового алгоритму.

НЕЧІТКІ ДУБЛІКАТИ, ЧАСТОТНИЙ АЛГОРИТМ, ШИНГЛУВАННЯ, ГІБРИДНИЙ МЕТОД

Вступ

Пошук дублікатів текстових та медіа-файлів завжди був одним з ключових завдань для пошукових інформаційних систем. Як правило, такі системи аналізують всі дані, які надаються пошуковими роботами. При цьому пошукові роботи автоматично індексують будь-який сайт, на який вони переходять за посиланнями, а також кожен новий сайт, зареєстрований в базі даних пошукової системи. Найчастіше роботи можуть індексувати одні й ті ж інформаційні ресурси з певною (наперед заданою) періодичністю, внаслідок чого в базу даних пошукових систем потрапляє надлишкова інформація.

Зберігання надлишкового обсягу даних, пов'язане з дублюванням даних, вимагає додаткових часових і обчислювальних витрат. Крім того, повне або часткове дублювання документів позначається на їх порядкових номерах в пошуковій видачі, тому що впливає на ранжування інформаційних блоків, які аналізуються.

Таким чином, вдосконалення методів виявлення нечітких дублікатів є актуальним завданням, успішне вирішення якого дозволяє підвищити ефективність пошукових систем.

Основною перешкодою для успішного виявлення нечітких дублікатів є гігантський обсяг інформації, яка зберігається в сучасних базах даних, що робить практично неможливим безпосереднє виявлення нечітких дублікатів шляхом попарного порівняння текстів документів. У зв'язку з цим останнім часом значна увага приділяється розробці методів зниження обчислювальної складності алгоритмів пошуку за рахунок застосування різних евристик (наприклад, хешування певного фіксованого набору «значущих» слів або речень документу, семплювання набору підрядків тексту, використання дактилограм тощо) [1].

Існуючі методи виявлення нечітких дублікатів спрямовані на мінімізацію обчислювальної складності з одночасним збільшенням повноти і точності алгоритму.

У даній роботі представлений метод, який дозволяє істотно поліпшити якість виявлення нечітких дублікатів при незначному збільшенні обчислювальної складності базового алгоритму.

1. Загальна характеристика проблеми виявлення нечітких дублікатів

Нечіткий дублікат документа – це документ, частково змінений у змістовній частині і / або в частині форматування.

Методи виявлення нечітких дублікатів дозволяють визначити, чи є два документи нечіткими дублікатами один одного. В даному випадку дуже важливо не тільки виявити всі нечіткі дублікати, але і помилково не ідентифікувати як дублікати документи, близькі за змістом. Таким чином, вдосконалення методів виявлення нечітких дублікатів електронних документів безпосередньо пов'язане зі зниженням ймовірності виникнення помилок першого і другого роду.

Існуючі на даний час методи аналізу електронних документів, що дозволяють виявляти дублікати, можна розділити на три основні групи:

– частотні методи, для яких основним критерієм вибору значущих слів з необробленого тексту є частота;

– вагові методи, де для кожної мовної одиниці за певними формулами розраховується ваговий коефіцієнт, який є умовною мірою значущості слова в аналізованому тексті;

– методи, в основі яких лежить шинглування [2].

Усі ці методи є наближеними, оскільки дозволяють уникнути повного попарного порівняння аналізованих документів і отримати прийнятні результати при допустимій обчислювальній складності.

При застосуванні наближених методів спостерігається зменшення (іноді вельми значне) показника повноти виявлення дублів.

Одним з ключових вимог, що висуваються до якості алгоритмів детектування нечітких дублікатів, є їх стійкість до невеликих змін вихідних документів і можливість обробляти короткі документи. Для виконання цих вимог вважається доцільним розробити гібридний алгоритм, що дозволяє об'єднати позитивні властивості існуючих методів виявлення дублікатів електронних документів.

Метою роботи є розробка та програмна реалізація гібридного методу виявлення нечітких дублікатів, який би мав кращі показники точності та повноти, ніж відомі алгоритми.

Згідно з цією метою необхідно вирішити наступні задачі:

– здійснити порівняльний аналіз існуючих методів з метою виявлення їх недоліків та переваг;

– розробити гібридний метод пошуку нечітких дублікатів, який забезпечив би необхідні повноту та точність за умови прийнятних обчислювальні витрат;

– здійснити програмну реалізацію та тестування розробленого алгоритму в середовищі Visual Studio за допомогою мови програмування C#.

2. Порівняльний аналіз методів виявлення нечітких дублікатів

Розглянемо особливості застосування методів виявлення нечітких дублікатів [2-5]. Слід зазначити, що перед оцінкою тексту (незалежно від типу методу, що застосовується) його попередньо очищують від розділових знаків, букв різного регістра і слів, які не несуть смислового навантаження (часток, спілок, прийменників, слів-паразитів). В основі вагових алгоритмів лежить ідея морфологічного (або якогось іншого) аналізу тексту з метою виявлення найбільш «важливих» або «важких» слів (в даній роботі під словом будемо розуміти деяку мовну одиницю). Очевидно, що не всі слова в тексті мають однакове смислове навантаження, тому в процесі аналізу складається так званий словник документа, де виділені слова йдуть в порядку убування їх «ваги». Далі вважається, що при визначенні нечіткого дубліката достатньо оцінювати не весь текст повністю, а частину отриманого словника, яка містить найбільш «важкі» слова документа.

Частотні алгоритми (зокрема, TF, TF*IDF, TF*RIDF) в певному сенсі є окремим випадком вагових алгоритмів, де «вага» слова відповідає частоті його появи в документі (tf). Далі (за аналогією з ваговими алгоритмами) складається словник, у якому слова розташовані за спаданням значень tf, і проводиться порівняння документів із застосуванням такого словника. Порівняння документів за їх словниками може проводитися різними способами. Зазвичай обирається обмежена (до 10) кількість найбільш важких слів із словників, для якого підраховуються контрольні суми (CRC32), а далі ці суми порівнюються.

Алгоритми, засновані на шинглуванні (зокрема, Log Shingles, Megashingles), відрізняються своєю простотою і ефективністю, тому саме їх використовують пошукові системи для виявлення нечітких дублікатів. Після очищення тексту тут відбувається виділення шинглів (послідовних фрагментів тексту), які обираються «внахлест» один за одним. Далі визначаються контрольні суми кожного з шинглів і здійснюється їх порівняння. Слід зазначити, що тут відсутні попередня оцінка документа і складання словника. Це збільшує швидкодію алгоритму, а розташування шинглів «внахлест» дозволяє підвищити його точність, так як контрольні суми таких шинглів дуже чутливі до найменших змін у наданому наборі слів.

Особливий інтерес становлять алгоритми, засновані на шинглуванні, що вперше були викладені в [1], а саме, метод декомпозиції і метод «3+5». Розглянемо ці методи докладніше.

У методі декомпозиції використовуються повні набори шинглів і методи скорочення квадратичної залежності, яка виникає, коли загальну ознаку має велика кількість документів. Ідея заснована на тому спостереженні, що розмір документа в словах має значну поділяючу властивість, а також на знаходженні загальних списків документів, які мають однакову ознаку.

Робота алгоритму полягає в наступному:

– для кожного документа обчислюємо неповторювані шингли (можна не всі, а вибірково) і зберігаємо у файлі в форматі <shingle, doc_id, doc_len>;

– для однакових шинглів будуємо ланцюжки в форматі <doc_id1, doc_len1> <doc_id2, doc_len2> ..., упорядковані за зростанням doc_len;

– поділяємо ланцюжки на більш дрібні, якщо у сусідніх довжин відношення більшої довжини до меншої перевищує деякий поріг, який визначається мінімальним рівнем подібності для дублікатів (наприклад, для рівня подібності 0,85 можна практично без втрати повноти використовувати поріг 1,15);

– видаляємо дублі ланцюжків і ланцюжки, які цілком входять в інші. У результаті число ланцюжків скорочується в сотні разів, а решта ланцюжків в переважній більшості є достатньо короткими (2 – 10 елементів);

– документи всередині ланцюжка порівнюємо попарно (наприклад, шляхом використання функції Perl Similarity або ж за допомогою якихось додаткових числових характеристик документів). При цьому порівняння здійснюється не за всім ланцюжком, а тільки в межах невеликої локальної околиці, обумовленої порогом відношення довжин, тому загальне число реальних порівнянь невелике;

– для виключення дублів перевірок в різних ланцюжках списки вже оброблених пар зберігаємо в хеші.

Основним недоліком алгоритму, що знижує його продуктивність, є необхідність обчислення шинглів для документів і використання функції Similarity. Хоча, як зазначалося раніше, можна обчислювати не всі шингли, а тільки деякі, у відповідності з якою-небудь евристикою, і замінити Similarity більш простими засобами.

У методі «3+5» використовуються лише ознаки, що вимагають мінімальних обчислювальних ресурсів, і не визначаються контрольні суми всіх підрядків тексту. Крім того, беруться до уваги лише «локальні» властивості текстів, тобто не використовуються глобальні параметри колекції.

Робота алгоритму полягає в наступному:

– для кожного документа колекції формується не більше трьох записів наступного вигляду:

<ss1,id,len,num,ss1,ss2,ss3,<ws1..5>>

<ss2,id,len,num,ss1,ss2,ss3,<ws1..5>>

<ss3,id,len,num,ss1,ss2,ss3,<ws1..5>>,

де

<ss1,ss2,ss3> – сигнатури трьох найбільш довгих речень документа, впорядковані за спаданням довжини пропозицій (а при рівності довжин – за сигнатурами); <id> – ідентифікатор документа; <len> – довжина документа (кількість слів завдовжки три і більше букв); <num> – число пропозицій в документі; <ws1..5> = <ws1,ws2,ws3,ws4,ws5> – сигнатури п'яти самих довгих слів документу, впорядковані за спаданням довжини слів (а при рівності довжин – за сигнатурами);

– для коротких документів, що складаються з одного або двох речень, створюються одна або два записи виду:

<sis1,id,len,sis1,0,0,<ws1..5>>, або

<sis1,id,len,sis1,sis2,0,<ws1..5>>

<sis2,id,len,sis1,sis2,0,<ws1..5>>;

– отриманий файл сигнатур сортується, та для записів зі співпадаючим першим полем формуються ланцюжки вигляду (найперше поле відкидається):

<id1,len1,num1,ss11,ss21,ss31,<ws11..51>>

<id2,len2,num2,ss12,ss22,ss32,<ws12..52>> ...;

– впорядковані за зростанням поля <len> – довжини документа (а при рівності довжин – по <id>). При цьому відношення довжин двох сусідніх елементів ланцюжка не повинно перевищувати деякого порогу, що визначається заданим мінімальним коефіцієнтом схожості документів (наприклад, для коефіцієнта 0,85 оптимальне значення порогу довівнює 1,15);

– при перевищенні порогу формування поточного ланцюжка закінчується і починається формування нового, незважаючи на те, що перше поле може залишатися колишнім. У результаті вихідний файл сигнатур перетвориться у множину порівняно коротких ланцюжків, що містять компактні (з невеликим розкидом) послідовності довжин документів, які монотонно спадають;

– файл ланцюжків сортується і з нього виключаються дублі та ланцюжки, які цілком входять в інші, довші. В результаті такої нормалізації виходить невеликого розміру файл ланцюжків, в якому коефіцієнт надмірності (входження елементів у різні ланцюжки) становить не більше 10%;

– оскільки всередині ланцюжка елементи впорядковані за зростанням довжин документів, для кожного елемента існує локальна околиця відносно невеликого розміру, визначеного тим же пороговим значенням відношення довжин;

– переглядаємо файл ланцюжків і для кожного елемента ланцюжка знаходимо дублікати за наступними правилами:

– шукаємо тільки в межах локальної околиці;

– порівнюємо пари, тільки якщо відношення числа пропозицій в них не перевищує деякого порогу (1,20) та з п'яти сигнатур слів <ws1..5> збігається не менше двох (не важливо в якому порядку);

– два документи вважаємо дублікатами, якщо у них збігаються сигнатури найдовших речень <ss1> або (для документів, що містять більше 5 речень, а таких - переважна більшість) з трьох сигнатур <ss1, ss2, ss3> збігаються дві (неважливо в якому порядку).

Для створення гібридного методу необхідно визначити сильні і слабкі сторони розглянутих алгоритмів, щоб вибрати основу для алгоритму і спосіб її модифікації. Результати оцінки наведених вище алгоритмів наведені в таблиці 1 [1].

Таблиця 1

Результати оцінки алгоритмів нечітких дублікатів

Метод	Повнота	Точність	F-міра
«3+5»	0,96	0,95	0,95
Long Sent	0,84	0,80	0,82
TF	0,60	0,94	0,73
Opt Freq	0,59	0,94	0,73
TF*RIDF	0,59	0,95	0,73
Heavy Sent	0,62	0,86	0,72
TF*IDF	0,54	0,96	0,69
Lex Rand	0,50	0,97	0,66
Descr Words	0,44	0,77	0,56
Log Shingles	0,39	0,97	0,56
Megashingles	0,36	0,91	0,51

У таблиці наведені три основні характеристики оцінки алгоритмів: точність, повнота і F-міра. Результати впорядковані за спаданням F-міри як результуючої характеристики, пов'язаної з першими двома параметрами. Очевидно, що найбільш ефективним є алгоритм «3+5», що має дуже високі показники за повнотою та точністю. У той же час деякі алгоритми (Log Shingles, Lex Rand і TF * IDF) мають більш високу точність. У двох з них використовуються частотні характеристики слів тексту.

Оцінюючи далі результати з табл. 1, виділяємо п'ять алгоритмів, що мають найбільш високі показники повноти результатів: «3+5», Long Sent, TF, Opt Freq, TF*RIDF. При цьому в п'ятірку найбільш точних алгоритмів входять Log Shingles, Lex Rand, TF * IDF, «3 +5» і TF * RIDF.

Таким чином, можна констатувати, що використання частотних характеристик дозволяє підвищити точність алгоритмів, а оцінка за довжиною слів і речень в кінцевому підсумку збільшує їх повноту. З таблиці також випливає, що алгоритми з хорошими показниками точності зазвичай мають низьку повноту, тому що вони відкидають занадто багато документів, що є нечіткими дублікатами, при цьому пропускаючи лише малу частину документів. Найбільшу повноту дає алгоритм «3 + 5» (різниця між показниками повноти алгоритму «3 +5» та інших алгоритмів становить 12%).

Як наслідок, доцільно взяти за основу гібридного методу саме алгоритм «3 + 5», доповнивши і модифікувавши його частотними параметрами з метою істотного підвищення точності (при незначному зменшенні повноти).

3. Розробка гібридного методу

Пропонований гібридний метод виявлення нечітких дублікатів заснований на аналізі як частотних характеристик, так і довжини слів. За аналогією з методом «3 + 5», назвемо гібридний метод методом «3 + 2», оскільки тут будуть враховуватися частота і довжина в співвідношенні 3/2. Відзначимо, що методи «3 + 5» і «3 + 2» багато в чому близькі, однак істотно розрізняються за процедурами вибору пропозицій і слів у них.

Для оцінки можливого дублювання в методі «3 + 2» пропонується враховувати по 5 пропозицій текстового документа і по 5 слів з цих пропозицій.

Наведемо опис етапів реалізації методу «3 + 2».

Етап 1: Створення частотного словника документа.

Цей етап відповідає початковим етапам частотних алгоритмів. Для кожного значущого слова підраховується ступінь його зустрічальності (як і в звичайних частотних алгоритмах – наприклад, в алгоритмі TF). Далі обираються 10 слів, які найбільш часто зустрічаються в документі, вони упорядковуються за частотою входження в документ, а при рівній кількості – за довжиною (якщо і довжини дорівнюють – в алфавітному порядку).

Етап 2: Вибір 5 речень, що описують документ.

У відповідності з принципом «3 + 2» спочатку обираються три речення, в які входить найбільше число слів з частотного словника документа. Далі обираються два найбільш довгих речення. Якщо речення повторюється, воно опускається і шукається наступне за довжиною речення. Перші три речення упорядковуються за кількістю входження слів з частотного словника документа, а при рівності слів – за довжиною. Останні два речення упорядковуються за довжиною.

Етап 3: Вибір 5 слів, що описують речення.

Тут також використовується принцип 3 +2: спочатку обираються слова з частотного словника документа (не більше трьох), які доповнюються найдовшими словами документа до загальної кількості слів, рівній 5. Якщо слово повторюється, воно відкидається і береться наступне за довжиною слово. Перші слова (обрані з частотного словника документа) розташовуються в порядку, у якому вони розташовані в частотному словнику. Решта упорядковуються по довжині.

Етап 4: Визначення сигнатур слів і речень.

Сигнатура – символ або ряд символів, що утворюють унікальний ідентифікатор об'єкта, предмета чи документа.

В якості сигнатур тексту використовуємо контрольні суми, що визначаються за алгоритмом CRC32, який найбільш широко застосовується у всіх розглянутих вище методах.

Етап 5: Складання ланцюжків за принципом алгоритму «3 +5», зі зміненою кількістю ланцюжків для 1 документа (5 замість 3). При цьому ланцюжки будуть мати наступну структуру:

<ss1,id,len,num,ss1,ss2,ss3,ss4,ss5<ws1..5>>
<ss2,id,len,num,ss1,ss2,ss3,ss4,ss5<ws1..5>>
<ss3,id,len,num,ss1,ss2,ss3,ss4,ss5<ws1..5>>
<ss4,id,len,num,ss1,ss2,ss3,ss4,ss5<ws1..5>>
<ss5,id,len,num,ss1,ss2,ss3,ss4,ss5<ws1..5>>.

Якщо речень менше п'яти, то створюється кількість ланцюжків, рівна кількості речень, з таким же порядком, як і в алгоритмі «3 + 5», а частотні характеристики опускаються.

Файл ланцюжків складається так само, як і в алгоритмі «3 + 5».

Пошук здійснюється згідно з принципами алгоритму «3 + 5»: пошук проводиться тільки в межах локальної околиці; пари порівнюються, тільки якщо відношення числа пропозицій в них не перевищує деякого порогу (1.20) і з п'яти сигнатур слів <ws1..5> збігається не менше двох (неважливо в якому порядку); два документи вважаються дублікатами, якщо у них збігаються сигнатури <ss1> або (для документів, що містять більше 5 речень) з п'яти сигнатур <ss1, ss2, ss3, ss4, ss5,> співпадають N сигнатур , де N = 2 у випадку пріоритетності повноти над точністю, або N = 3 у разі пріоритетності точності над повнотою (неважливо в якому порядку).

4. Вхідні та вихідні дані

Метод проходив тестування на матеріалах, наданих електронною бібліотекою ХНУРЕ – а саме, методичних вказівках до лабораторних і курсових робіт. Виходячи зі специфіки даних матеріалів, подібні документи перевидаються практично щорічно, з незначними змінами. Ці документи були оцінені експертами і в них були виділені пари і множини нечітких дублікатів щодо кожного документа.

Вихідними даними методу є отримані в результаті роботи множини нечітких дублікатів щодо кожного документа.

5. Оцінка результатів роботи

Так як для визначення дубліката було вибрано два варіанти алгоритму (для досягнення більшої точності й для більшої повноти), нижче наведені результати його роботи в двох цих випадках.

Повнота визначається як відношення числа знайдених нечітких дублікатів до загального числа нечітких дублікатів документів в базі:

$$\mathrm{Re}\,call = \frac{|D_{dub} \cap D_{retr}|}{|D_{dub}|},$$

де D_{dub} — множина нечітких дублікатів документів

в базі, а D_{retr} — множина документів, знайдених системою.

Точність визначається як відношення числа нечітких дублікатів, знайдених методом, до загального числа знайдених документів:

$$Precision = \frac{|D_{dub} \cap D_{retr}|}{|D_{retr}|},$$

де D_{dub} — множина нечітких дублікатів документів в базі, а D_{retr} — множина документів, знайдених системою.

F-мера при однаковій вазі точності і повноти:

$$F = \frac{2PR}{R+P},$$

де R – повнота, а P – точність.

Таблиця 2

Результати роботи методу

Метод	Повнота	Точність	F-міра
«3+5» (оригінальний)	0,96	0,95	0,954
«3+2»; N=3 (орієнтованість на точність)	0,927	0,978	0,952
«3+2»; N=2 (орієнтованість на мінімальну втрату повноти)	0,956	0,963	0,959

З табл. 2 видно, що точність роботи методу «3 + 5» зростає у двох випадках, проте втрати повноти залишаються неминучими. Тим не менш, у другому випадку втрата повноти складає всього 0,4%, тоді як точність підвищується на 1,3%. Це відповідає збільшенню інтегрованого показника F-міри, а значить, можна говорити про загальне підвищення ефективності роботи алгоритму. Таким чином, результати тестування підтверджують доцільність застосування запропонованого гібридного методу для автоматичного виявлення нечітких дублікатів в електронних текстах. Можлива його подальша модернізація з перевизначенням деяких параметрів (наприклад, відношення частотної складової до складових, обраних на основі довжин), зменшенням кількості ланцюжків, що складаються при обробці тексту тощо. Перспективним видається проведення досліджень методу «3 + 2», спрямованих на збільшення показника повноти без зменшення точності

Висновки

В роботі наведено результати роботи гібридного методу пошуку нечітких дублікатів, розробленого методом модифікації алгоритму «3+5» з роботи [1] з введенням частотних характеристик слів з документа. Цей метод має на меті покращення показників точності виявлення дублікатів з найменшими можливими втратами у повноті. Роботу запропонованого гібридного методу було протестовано на виборці електронних документів (методичних вказівок, наданих електронною бібліотекою ХНУРЕ).

В ході тестування розроблений метод виявлення нечітких дублікатів показав суттєве покращення результатів за показниками точності з невеликою втратою у показниках повноти.

Перспективною здається подальша модернізація алгоритму за допомогою удосконалення процедур вибору параметрів створення ланцюгів та інших ключових параметрів.

Перспективним також є знаходження дублікатів медіа-файлів (аудіо-, відео файлів, файлів зображень) з певною модифікацією цього алгоритму та відповідною попередньою обробкою.

Список літератури: 1. *Зеленков Ю.Г.* Сравнительный анализ методов определения нечетких дубликатов для Web-документов [Текст] / Ю.Г. Зеленков, И.В. Сегалович // RCDL'2007: Сб. работ участников конкурса: Переславль-Залесский, Россия, 2007. – Том 1. – С. 166-174. **2.** *Чалая, Л. Э.* Определение нечетких дубликатов медиафайлов [Текст] / Л.Э.Чалая, П.Ю.Попаденко // Международная научно-техническая конференция, посвященная 75-летию В.В. Свиридова «Информационные системы и технологии» ИСТ-2012: матеріали першої міжнар. наук.-техн. конф., 22–29 вересня 2012 р. – С.74. **3.** *S. Robertson*. Okapi at trec-3 [Text] / S. Robertson, S. Walker S. Jones, M. Hancock-Beaulieu, M. Gatford // The Third Text REtrieval Conference (TREC-3), 1995 – P.109-127. **4.** *K. Church*. Poisson mcxtures [Text] / K. Church, W. Gale. // Natural Language Engineering, 1995, P. 163–190. **5.** *A. Broder*. On the resemblance and containment of documents, Compression and Complexity of Sequences [Electronic resource] SEQUENCES'97, IEEE Computer Society, 1998, – P. 21-29.

Надійшла до редколегії 07.12.2012

УДК 004.912

Метод обнаружения нечетких дубликатов электронных текстовых документов / Л.Э. Чалая, П.Ю. Попаденко // Бионика интеллекта: науч.-техн. журнал. – 2013. – № 1 (80). – С. 88-92.

В работе проведен анализ методов обнаружения нечетких дубликатов текстовых файлов. Показано, что существующие методы обнаружения нечетких дубликатов направлены на минимизацию вычислительной сложности с одновременным увеличением полноты и точности алгоритма. В данной работе предложен и протестирован гибридный метод, который позволяет существенно улучшить качество обнаружения нечетких дубликатов при незначительном увеличении вычислительной сложности базового алгоритма.

Табл. 2. Библиогр.: 5 назв.

УДК 004.912

Method of near-duplicate detection for electronic textual documents / L.E. Chalaya, P.Yu.Popadenko // Bionics of Intelligense: Sci. Mag. – 2013. – № 1 (80). – P. 88-92.

This paper analyzes the methods of near-duplicate detection in text files. It is shown that the existing methods of near-duplicates detection directed on minimization the computational complexity while increasing the completeness and accuracy of the algorithm. In this paper the hybrid method was proposed and tested, which can significantly improve the quality of near-duplicate detection slightly increasing the computational complexity of the basic algorithm.

Tab. 2. Ref.: 5 items.