УДК 81:004.93

О.В. Годич, Ю.С. Наконечний, Ю.М. Щербина
Львівський національний університет ім. Івана Франка,
кафедра дискретного аналізу та інтелектуальних систем

КАТЕГОРИЗАЦІЯ ЕЛЕКТРОННИХ ДОКУМЕНТІВ

Нині існує багато алгоритмів кластеризації документів, кожен з яких має певні переваги та недоліки. Запропонований у цій статті алгоритм є спробою компромісу між якістю категоризації документів і вимогами до обчислювальних ресурсів, залишаючись незалежним від мови документів. Алгоритм використовує техніку пошуку ключових слів документів для формування вхідних даних та карти Кохонена в поєднанні з ієрархічною кластеризацією для категоризації та візуалізації документів.

Ключові слова: карти Кохонена, зменшення розмірності, категоризація документів, кластеризація, візуалізація даних.

Nowadays, a number of document clustering algorithms are available with number of benefits and trade-offs. The proposed in this article algorithm is an attempt to provide a compromise between quality and complexity, while remaining independent of the language. It uses keyword-based dimensionality reduction technique to create an input space, and then applies Self-Organizing Map for clustering and visualization.

Keywords: Self-Organizing Map, dimensionality reduction, document categorization, clustering, data visualization.

Постановка проблеми

Зі швидким розвитком комп'ютерних технологій дедалі більше інформації починає зберігатись в цифровому вигляді. Левову частку цієї інформації становлять книги, документи, статті, журнали тощо. Причиною цього є зручність передавання, копіювання та обробки цієї інформації порівняно із друкованою. Проте здебільшого нам відомо дуже мало про зміст електронних документів, які потрапляють до нас з Інтернету чи просто з інших носіїв інформації. Назви файлів, зазвичай, нічого нам не кажуть про їхній вміст, тому визначити зміст окремого тексту можливо, лише переглянувши його. У разі малих обсягів документів це зробити насправді не важко, проте зі зростанням кількості електронних документів людина мусить переглядати кожен, щоб пригадати, про що в ньому йшлося. Виникає потреба в автоматизації процесу категоризації документів, яка здійснюється без втручання людини.

Математична модель задачі категоризації документів

Нехай $D = \{d_1, d_2, \cdots, d_n\}$ – скінченна множина документів. Цю множину потрібно розділити на скінченну кількість категорій так, щоб документи, що належать до однієї категорії, були ближчими за змістом, ніж документи з різних категорій. Тобто мають виконуватись умови:

$$D_1, D_2, \cdots, D_m \subset D, m < n$$

$$D_1 \cup D_2 \cup \cdots \cup D_m = D$$

$$\forall i, j \le m, i \ne j : D_i \cap D_j = \varnothing$$

Нехай $\rho : D \times D \to \mathsf{R}_+$ – функція подібності документів. Умову приналежності документів до певної категорії можна записати так: $\forall k \le m, \forall i, j, s : d_i, d_j \in D_k, d_s \notin D_k$, $\rho(d_i, d_j) < \rho(d_i, d_s)$. Функцією подібності може бути довільна функція, для якої виконуються умови:

$$\forall a, b \in D, \rho(a,b) = 0 \Leftrightarrow a = b$$

$$\forall a, b \in D, \rho(a,b) = \rho(b,a)$$

$$\forall x, y, z \in D, \rho(x,z) \le \rho(x,y) + \rho(y,z)$$

Проблема подання документів. Для розв'язування задачі категоризації необхідно подати електронні документи у вигляді математичної абстракції. Серед цілої низки способів подання документів вибрано найбільш інтуїтивний та тривіальний – подання документів у вигляді вектора частот появи слів у документі [1]. Кожен документ із множини $D = \{d_1, d_2, \ldots, d_n\}$ подаватиметься у вигляді вектора $d = (\nu_1, \nu_2, \ldots, \nu_s)$, де $\nu_i (i = 1,2,\ldots,s)$ частота появи i-го слова в документі, s – кількість слів документа. Частота появи i-го слова в документі визначається як $\nu_i = \frac{w_i}{\sum_{j=1}^{s} w_j}$, де w_i – кількість появ i слова у документі, $\sum_{j=1}^{s} w_j$ – кількість слів у документі. Очевидно, що $\sum_{i=1}^{s} \nu_i = 1$. Отже, можна визначити лінійний метричний простір документів $D = \{d \mid d = (\nu_1, \nu_2, \ldots, \nu_s), \forall i = 1,\ldots,s, \nu_i \in [0,1]\}$ із метрикою L_2: $\forall x, y \in D, \rho(x, y) = \sqrt{\sum_{i=1}^{s} (x_i - y_i)^2}$. Зауважимо, що цей простір є опуклим, тобто $\forall x, y \in D, \forall \alpha \in [0,1] : (1-\alpha)x + \alpha y \in D$. Ця властивість є важливою для категоризації простору документів за допомогою карт Кохонена та алгоритму ієрархічної кластеризації UPGMA.

Скорочення розмірності простору документів. Ще в середині XX століття лінгвіст Зіпф, досліджуючи частоту появи слів у англійських текстах, виявив закономірність, яку згодом назвали законом Зіпфа: *якщо впорядкувати слова за зниженням частоти їх появи в конкретному тексті, то частота появи i-го слова в заданій послідовності наближено дорівнює* $\nu_i \approx \frac{C}{i^\alpha}$. Для англійської мови $C \approx 0.04, \alpha \approx 1.012$. Як виявилось пізніше, це співвідношення справджується не тільки для англійської мови, а й для інших мов. Також було доведено, що довільно генеровані тексти також підпорядковуються цьому закону [2].

Закон Зіпфа був використаний у цьому дослідженні для визначення частоти слів. Наприклад, потрібно відібрати 100 перших слів із найбільшою частотою в документі. Нехай у документі найчастіше трапляється слово “the” із частотою 0.1. Для більшої точності, визначимо константу C: $\nu(the) = 0.1 = \nu_1 \approx \frac{C}{1} \Rightarrow C \approx \nu_1 = \nu(the) = 0.1$. Тобто частота сотого слова приблизно дорівнює: $\nu_{100} \approx \frac{C}{100} = \frac{0.1}{100} = 0.001$. Отже, для відбору ста слів із найбільшою частотою в документі потрібно відібрати слова, частота яких більша або дорівнює 0.001. Внаслідок того, що одне й те саме слово в різних формах розглядається як різні слова, розмірність документів може сягати 10^5. Використання векторів таких розмірностей має два недоліки: опрацювання документів, поданих такими векторами, є надзвичайно ресурсоємним; простори таких великих розмірностей є дуже розрідженими. Наприклад, слово, яке трапилось в одному з документів 100 разів, може не зустрітись жодного разу в усіх решті документах. Для подолання цих, тісно пов'язаних із категоризацією, проблем широко використовують різні методи скорочення вимірності [5]. Основна ідея цих методів полягає у відображенні документів великої розмірності в простір меншої розмірності, беручи до уваги взаємозалежності між компонентами векторів. Потреба скорочення вимірності породжує низку нових проблем.

Невідома реальна розмірність, тобто не існує способу визначення мінімальної кількості вимірів, достатньої для подання даних.

Нелінійність залежності між компонентами, оскільки залежність між компонентами, зазвичай, є дуже складною.

Невідома значущість інформації. Ситуація, коли скорочення вимірності не призводить до втрати інформації, є ідеальною. Часто скорочення вимірності неминуче веде до втрат інформації.

Внаслідок того, що процес скорочення вимірності є досить складним, не існує єдиного методу, який був би однаково ефективним у всіх випадках. У наш час існує ціла низка різних методів, які, загалом, можна згрупувати в три категорії:

– до першої категорії належать методи, які впродовж скорочення вимірності використовують інформацію про приналежність документів до того чи іншого класу. Ці методи ставлять за мету мінімізацію втрати інформації порівняно з оригінальним багатовимірним простором;

– до другої категорії належать методи, основані на статистичному аналізі, аналізі головних значень та багатовимірному масштабуванні. Ці методи є особливо ефективними, коли взаємозв'язки між компонентами є лінійними;

– до третьої категорії належать методи типу самоорганізаційних карт Кохонена.

Фомулювання цілей статті

Нині розроблена ціла низка методів для розв’язання задачі категоризації електронних документів: одні намагаються вирішити цю проблему штучно, додаючи до документів метаінформацію про їхній зміст, інші – використовують різноманітні лінгвістичні алгоритми, основним недоліком яких є залежність від мови. І лише дуже мала кількість методів є незалежною від мови. Найвідомішим із таких методів є WEBSOM [4]. Цей метод використовує двошарову нейромережу для категоризації сукупностей документів. Основною його перевагою є висока якість категоризації, головним недоліком – високі вимоги до обчислювальних ресурсів, що унеможливлює його застосування на персональних комп'ютерах. У цій роботі пропонується метод, який є, по суті, компромісом між якістю та швидкодією порівняно з WEBSOM, що дає можливість розробити програмне забезпечення аналізу персональних колекцій документів окремими користувачами. Кінцевою метою дослідження є створення програмного забезпечення, яке б автоматично категоризувало всі електронні документи в межах файлової системи комп'ютера та давало користувачу можливість переглядати категорії, визначати зміст нових документів, встановлюючи категорію, до якої вони належать, та ефективно самонавчатись.

Викладення основного матеріалу

Метод визначення множини ключових слів

Запропонований метод скорочення розмірності векторів, які подають документи, оснований на визначенні множини ключових слів. Зазвичай *ключовим* називають слово, яке *характеризує зміст документа*. Таке визначення є неповним, коли йдеться про категоризацію сукупностей документів. Для прикладу розглянемо дві сукупності документів: перша містить документи із програмування, математики, екології, літератури та медицини; друга – документи, які описують мову програмування Java, серед яких можна виділити книги про Java як мову, книги про віртуальну машину Java та книги про Java-технології.

Слово “java” є ключовим у випадку першої сукупності документів, оскільки наявність цього слова в документі характеризує його як документ, у якому йдеться про програмування. У випадку другої сукупності наявність слова “java” в документі не дає жодної інформації про його відмінності від інших документів, а отже, воно не є ключовим. Із цього прикладу видно, що поняття ключового слова істотно залежить від сукупності документів, які розглядаються. Тому *ключовими* називатимемо слова, які *найкраще характеризують документ з-поміж інших документів цієї сукупності*.

Природно, постає запитання: як визначити множину ключових слів для заданої сукупності документів? Для відповіді на це запитання розглянемо конкретну сукупність з 77 документів, яку можна поділити на дві, повністю відмінні за змістом, категорії: книги із програмування мовою Java та книги зі стоматології. Проаналізуємо частоти появи слів “java”, “class”, “the”, “and”, “dental”,

“implant” у документах цієї сукупності. Звернемо увагу на те, що слова “java” та “class” можна вважати ключовими, бо їхня наявність у документі свідчить, що він стосується категорії програмування (останнє слово може вживатись й в іншому контексті, проте в цьому випадку його частота буде значно меншою). Те саме стосується слів “dental” та “implant” – їхня наявність у документі свідчить про його належність до категорії стоматології. Водночас, слова “the” та “and” не дають жодної інформації про зміст документа. Графіки частот появи цих слів у документах вибраної сукупності наведено на рис. 1, 2.

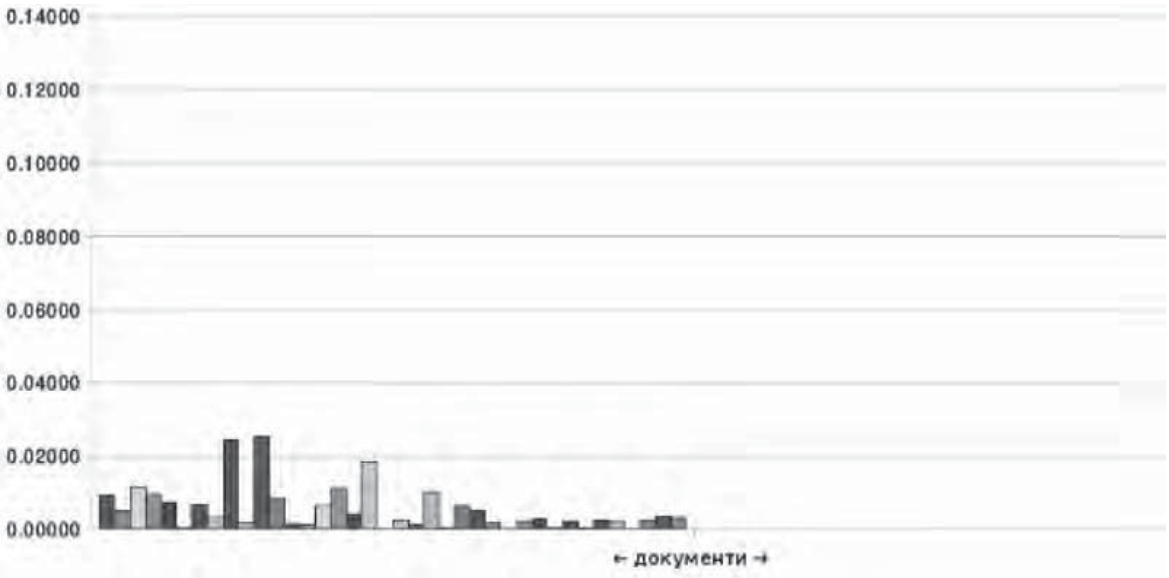


Рис. 1. Частота появи слова “java” у документах сукупності (з'являється в 38 з 77 документів)

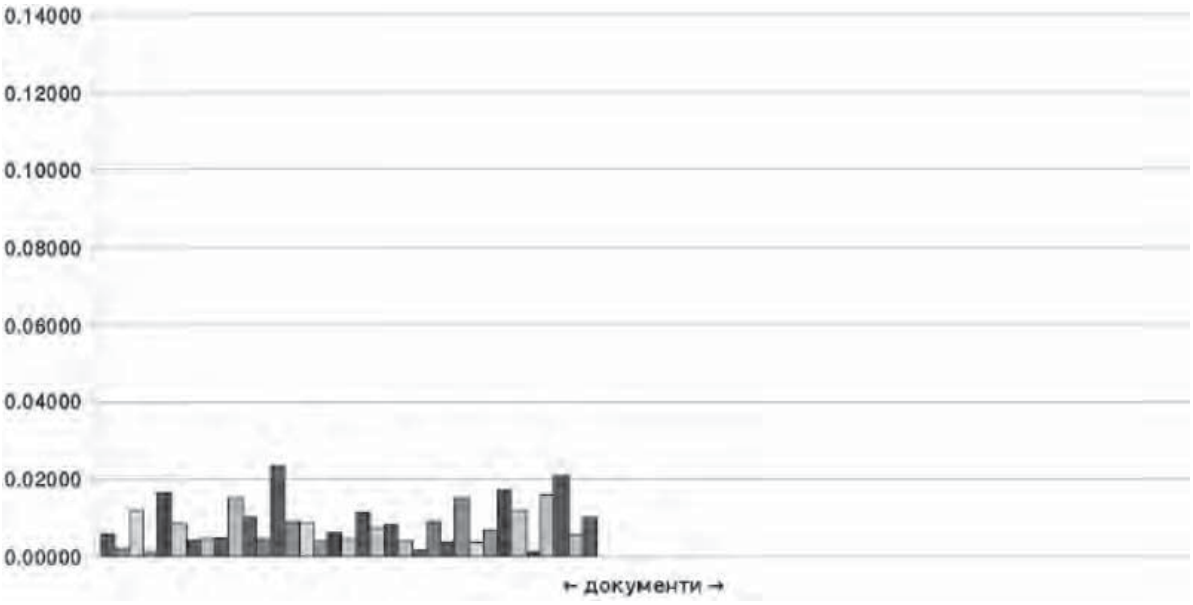


Рис. 2. Частота появи слова “class” у документах сукупності (з'являється в 36 з 77 документів)

Як видно з цих графіків, ці слова є тільки в частині документів. Це свідчить про те, що ці документи належать до категорії книг програмування. Тепер розглянемо графіки частот появи слів “the” та “and” у документах вибраної сукупності (рис. 3, 4).

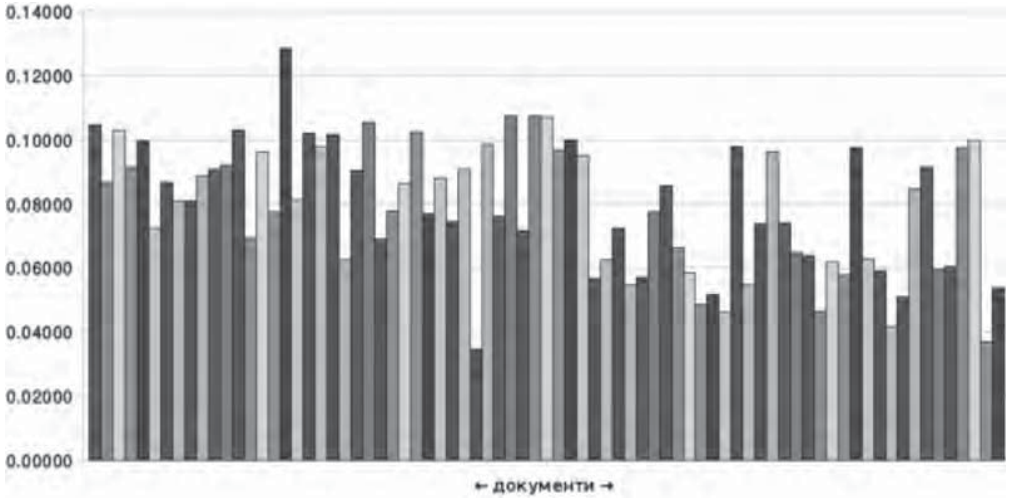


Рис. 3. Частота появи слова "the" у документах сукупності (з'являється у всіх 77 документах)

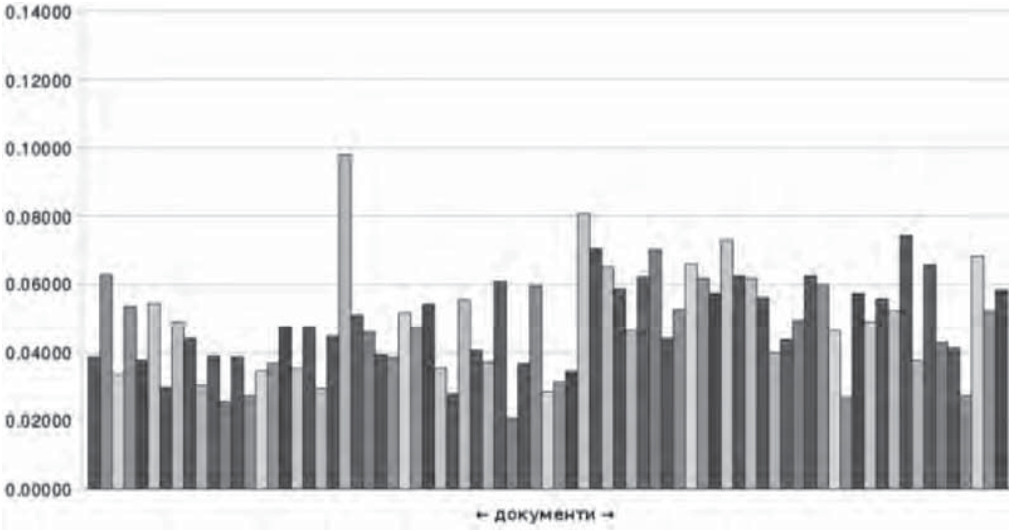


Рис. 4. Частота появи слова "and" у документах сукупності (з'являється у всіх 77 документах)

Розподіл частот появи слів "the" та "and" рівномірніший, на відміну від слів "java" та "class". Вони виявлені в усіх документах із порівняно високою частотою. Порівняємо частоту цих двох слів із частотами "dental" та "implant", які вживаються у специфічніших контекстах.

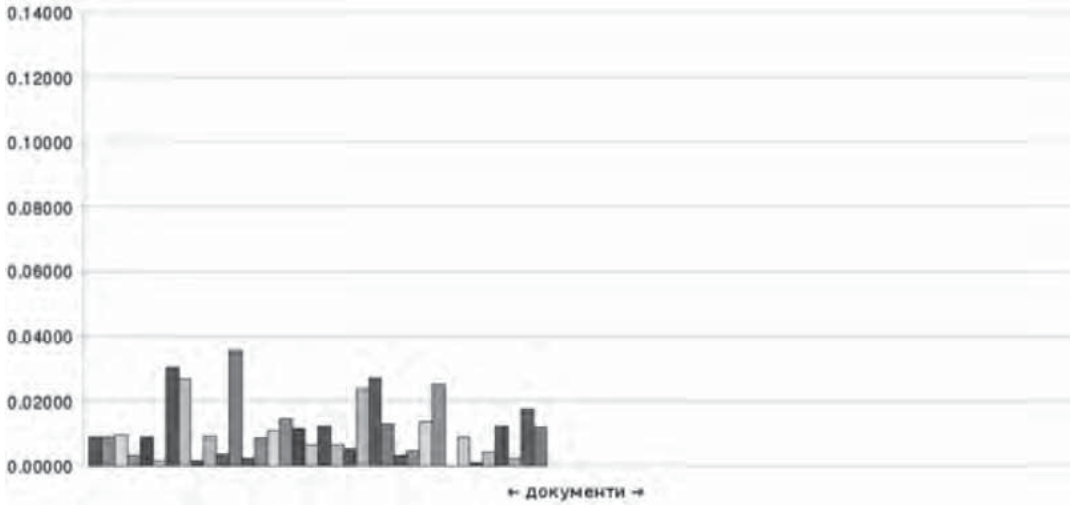


Рис. 5. Частота появи слова "dental" у документах сукупності (з'являється в 40 з 77 документів)

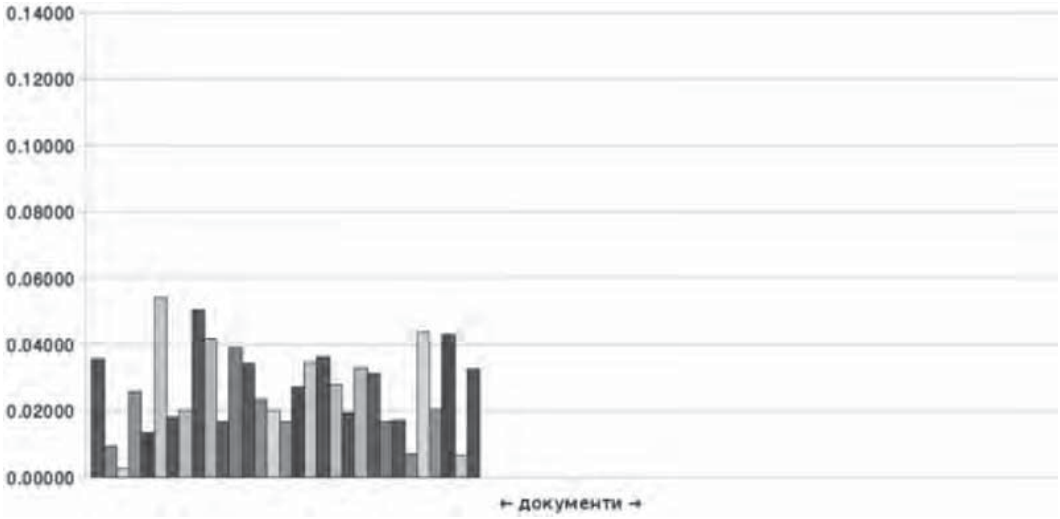


Рис. 6. Частота появи слова "implant" у документах сукупності (з'являється в 31 з 77 документів)

З вищенаведених рисунків видно, що розподіл частот слів "java", "class", "dental" та "implant" є схожими, на відміну від розподілів слів "the: та "and". Слова "java", "class", "dental" та "implant" є ключовими, "the" та "and" нічого не кажуть нам про контекст.

Для виділення ключових слів із-поміж усіх інших потрібно вміти розділяти частоти їхніх розподілів. На перший погляд здається, що можна просто порівнювати кількості документів, у яких вони трапляються. Проте це припущення є хибним, бо на початку нам не відома кількість категорій у сукупності, тому ключові слова можуть бути й у половині всіх документів (у випадку двох категорій), і в третині (у разі трьох категорій). Насправді, якщо ще раз переглянути наведені рисунки, то можна зауважити, що дисперсія частот слів "java", "class'", "dental" та "implant" відрізняється від дисперсії слів "the" та "and". Отже, як дискримінатор можна використовувати *дисперсію частот появи слова в документах*. Міри дисперсії можна поділити на дві групи: *абсолютні* та *відносні* [7]. Перші відповідають абсолютному відхиленню від середнього (залежать від величини значень), другі – указують на відносне значення відхилення від середнього (не залежать від величини значень). Прикладами абсолютних дисперсій є розмах вибірки ($\rho = max(x_1,\ldots,x_n) - min(x_1,\ldots,x_n)$), стандартна девіація ($D = \sum_{i=1}^{n}(x_i - \bar{x})^2$), варіанса ($s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$), стандарт ($s = +\sqrt{s^2}$), середня абсолютна девіація ($AD = \sum_{i=1}^{n} | x_i - \bar{x} |$). Прикладами відносних дисперсій є варіація ($v = \frac{s}{\bar{x}}$) та відносна середня абсолютна девіація ($RAD = \frac{AD}{\bar{x}}$). Під час досліджень дисперсій слів у сукупностях документів виявлено, що тільки відносні міри дисперсії відділяють ключові слова від неключових. При використанні абсолютних мір дисперсії (наприклад, варіанси) значення дисперсій неключових слів було значно більшим від значень дисперсій ключових. Під час дослідження значень відносних мір дисперсії серед ключових слів ці значення були набагато більшими, ніж серед неключових. Крім того, слово "the", яке має найбільшу частоту в англійських текстах, завжди мало найменшу відносну дисперсію. Слово "and" опинилось на другому місці після слова "the"'. Така відмінність між абсолютними та відносними мірами дисперсії пояснюється тим, що частоти неключових слів, зазвичай, є на порядок більшими порівняно із ключовими. Тому і абсолютні міри дисперсії (які прямо залежать від величини значень) неключових слів є більшими порівняно із ключовими. При виборі відносних мір дисперсій значення для неключових слів є дуже низькими, оскільки такі слова вживаються у всіх документах із приблизно однаковою частотою.

Під час подальшого дослідження відносних мір дисперсії, при виборі варіації, виникли проблеми, пов'язані з точністю обчислень. Оскільки частоти багатьох ключових слів становили 10^{-4}, то під час обчислення варіації близько 300 перших слів мали однакові значення варіації із точністю до 15 знаків після коми. Це було спричинено тим, що при обчисленні варіації значення частот спочатку підносяться до квадрата, із суми квадратів береться корінь, після чого це значення ділиться на усереднене значення того самого порядку, що призводить до значних втрат у точності. Ці втрати зберігались і в разі використання покращених схем обчислень варіації. Тому потрібно було шукати відносну міру дисперсії, яка б максимально зберігала точність. Такою виявилась *відносна середня абсолютна девіація*:

$$RAD = \frac{\sum_{i=1}^{n} | x_i - \bar{x} |}{\bar{x}}.$$

У разі використання цієї міри дисперсії слова, що найкраще характеризуватимуть контекст категорії документів, матимуть найбільші значення, водночас, слова, які не містять жодної інформації про контекст, – найменші.

Для покращання якості вибору ключових слів доцільно додатково опрацьовувати сукупність документів у спосіб, оснований на припущенні, що *слова, які з'являються в тексті з досить малими частотами, не несуть інформації про контекст*. Справді, як було згадано вище, слово “class” може вживатись в звичайному оповіданні, проте частота його появи буде значно меншою, ніж частота в книзі про програмування. Тому перед тим, як визначати ключові слова, потрібно встановити мінімально допустиму частоту, за якої братимуться до уваги тільки ті слова, частота яких є більшою за мінімально допустиму. Для того щоб не вгадувати значень частот слів у тексті, можна скористатись законом Зіпфа. Під час експериментів було виявлено, що для оптимального визначення ключових слів потрібно брати не більше ніж 00 слів, які вживаються в документі з найбільшою частотою. Отже, метод скорочення розмірності простору векторів, які подають документи, можна записати так:

Вихідний простір документів. $D = \{d \mid d = (\nu_1, \nu_2, \cdots, \nu_s), \forall i = 1, \ldots, s, \nu_i \in [0,1]\}$.

Видалення слів із малою частотою. Визначаємо мінімальну допустиму частоту слів $0 \le \nu_m in < 1$. Кожен вектор $d \in D$ замінюємо на вектор $\tilde{d} = (\tilde{\nu}_1, \ldots, \tilde{\nu}_s), \tilde{\nu}_i = \begin{cases} \nu_i, \nu_i \ge \nu_{\min} \\ 0, \nu_i < \nu_{\min} \end{cases}$

Визначення ключових слів. Для кожного слова, виявленого у певній сукупності з m документів, будуємо вектор $\nu = (\nu_1, \ldots, \nu_m)$, де ν_i – це частота появи цього слова в i-му документі (частота з урахуванням перетворення векторів із попереднього пункту). Після цього визначимо відносну середню абсолютну девіацію цього вектора $RAD(\nu) = \frac{\sum_{i=1}^{m} | \nu_i - \bar{\nu} |}{\bar{x}}$. Це значення відповідає значущості слова для визначення контексту. Упорядковуємо слова за спаданням обчисленого значення девіації.

Скорочення розмірності вихідного простору. Визначаємо кінцеву розмірність k, до якої необхідно скоротити вихідний простір (доцільно зауважити, що якість подання документів, а отже, і якість подальшої категоризації залежить від розмірності, тому рекомендовано вибирати розмірності, не менші за 100). З упорядкованого списку з попереднього пункту вибираємо перших k слів, і подаємо документи як вектори частот появи цих слів розмірності k.

Загальний процес категоризації документів можна записати так.

Подання документів у вигляді векторів частот появи всіх слів. Кожен із N документів, які потрібно категоризувати, подається у вигляді вектора частот появи всіх слів відповідних документів.

Скорочення вимірності векторів. З огляду на необхідну якість та швидкість категоризації вибираємо розмірність векторів k, до якої потрібно скоротити вихідні вектори. Для цього можна використати обговорені підходи. У результаті отримуємо N векторів розмірності k, кожен з яких подає документ із початкової сукупності.

Категоризація утворених векторів. Задача категоризації документів звелась до задачі категоризації N векторів розмірності k. Для категоризації було вибрано алгоритм самоорганізаційних карт Кохонена.

Категоризація документів за допомогою самоорганізаційних карт Кохонена

Розглянемо організацію процесу категоризації документів та візуалізації результатів за допомогою карт Кохонена [3, 4]. Принцип роботи карт Кохонена полягає у побудові відображення вихідного простору векторів високої розмірності на двовимірну ґратку, що уможливлює візуалізацію багатовимірних даних на звичайному дисплеї. Ця властивість є особливо привабливою з огляду на потребу подання результатів категоризації документів кінцевому користувачу. Детальніше з теорією карт Кохонена можна ознайомитись в працях [4] та [8–13].

Залежно від складності вхідного простору, параметрів навчання та кількості елементів карти Кохонена якість побудованого відображення може бути різною. Тому виникає необхідність застосовувати кількісні критерії перевірки якості сформованої карти:

MSE (Mean square error) – визначає, наскільки добре карта Кохонен апроксимує вхідний простір. $MSE = \frac{1}{N}\sum_{i=1}^{N} | x_i - bmu(x_i) |$, де N – кількість векторів у вхідному просторі, x_i – вектор із вхідної множини, $bmu(x_i)$ (best-matching unit) – елемент-переможець для вектора x_i, тобто елемент карти Кохонена, який є найближчим у сенсі метрики вхідного простору. Отже, MSE вказує, наскільки добре карта Кохонена "натягнута" на вхідний простір.

TE (Topological error) – визначає якість збереження топології вхідних даних: $TE = \frac{1}{N}\sum_{i=1}^{N} t(x_i)$, $t(x_i) = \begin{cases} 1, sbmu(x_i) \notin N_{bmu(x_i)} \\ 0, sbmu(x_i) \in N_{bmu(x_i)} \end{cases}$, де $N, x_i, bmu(x_i)$ – мають те саме значення, що й для MSE, $sbmu(x_i)$ (second best matching unit) – другий елемент-переможець для вектора x_i, тобто елемент, який є наступним найближчим після елемента-переможця, $N_{bmu(x_i)}$ – множина всіх елементів, які є безпосередніми сусідами елемента-переможця для вектора x_i.

Візуалізація даних. Найпоширенішим способом візуалізації даних за допомогою карт Кохонена є алгоритм U-Matrix [14]. Під час візуалізації карти для кожного елемента обчислюється значення висоти U-height, яке дорівнює сумі відстаней до вагових векторів сусідніх елементів. Щоб знайти значення висоти для елемента всередині прямокутної ґратки, обчислюють суму відстаней до чотирьох сусідніх елементів. Отже, якщо елемент розташований на межі кластера, значення його висоти буде великим, оскільки відстані до деяких із сусідніх елементів (а саме тих, що належать до інших кластерів) є великими. Якщо ж він міститься всередині кластера, то елементи, які його оточують, мають близькі вагові вектори (оскільки вони належать до того самого кластеру), отже, значення його висоти буде невеликим. Розмістивши значення висот у матриці, яка відповідає ґратці елементів, отримаємо матрицю, яку й називають U-Matrix. Цю матрицю можна зобразити графічно за допомогою чорно-білої палітри, присвоюючи малим значенням висот кольори, ближчі до чорного, а великим значенням – ближчі до білого. Чорним областям відповідатимуть кластери, а білим – межі між ними (див. рис. 7).

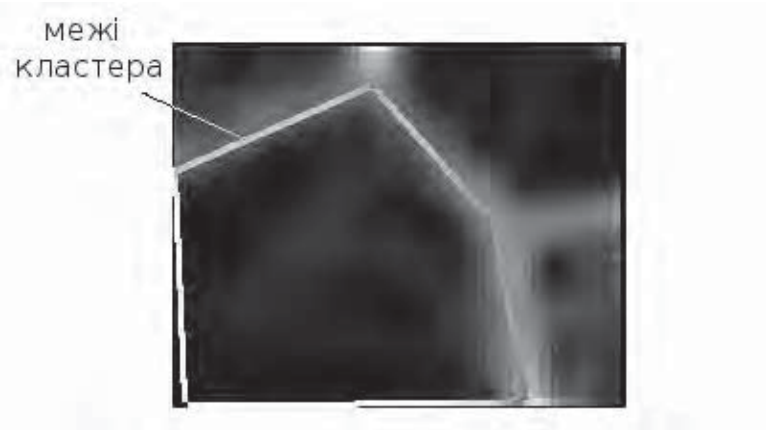


Рис. 7. Приклад U-Matrix для навченої SOM

Як видно з рис. 7, U-Matrix здатна візуалізувати структуру даних, проте ця візуалізація може бути не зовсім чіткою, і в багатьох випадках важко визначити, скільки кластерів відображаються навченою картою, використовуючи лише U-Matrix. Для визначення чітких меж кластерів дуже зручно користуватись алгоритмом UPGMA, застосованим до навченої карти Кохонена [6]. Алгоритм UPGMA належить до алгоритмів ієрархічної кластеризації. Відповідно до [6], алгоритм поділу навченої карти на кластери з використанням алгоритму UPGMA є таким:

– побудувати U-Matrix для навченої карти Кохонена;
– застосувати алгоритм UPGMA до множини вагових векторів елементів карти;
– визначити необхідну кількість кластерів;
– зобразити ці кластери.

У результаті цього алгоритму отримаємо зображення, подане на рис. 8.

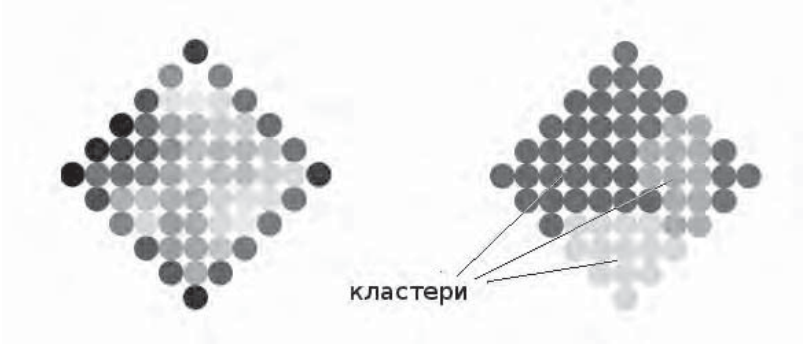


Рис. 8. Приклад застосування UPGMA до навченої карти

З рис. 8 легко бачити, що з U-Matrix практично неможливо візуально відокремити межі кластерів. Водночас, запропонований алгоритм із використанням UPGMA вказує на чіткі межі п'яти кластерів.

Комп'ютерні експерименти

Експеримент №1. У *першому експерименті* була поставлена задача категоризації 76 документів, серед яких були документи із двох протилежних за змістом областей: книги про програмування мовою Java та статті зі стоматології. До першої категорії належав 41 документ, до другої – 35. Попередньо було вилучено всі слова з кількістю літер, меншою від 3, після чого з документів вибирались слова із частотою, більшою від 0.001. У результаті цього було отримано 4256 різних слів. Варто зазначити, що в процесі вилучення слів із малою частотою було вилучено 26839 слів. Після цього для кожного слова було визначено величину відносної середньої абсолютної девіації. Далі наведено перелік ключових слів із найбільшими дисперсіями:

patol 18.051890675594095	quantify 4.485684210526316
native 15.723159829959815	cases 4.421684210526316
transitional 15.517801857585138	specialist 4.360080737299108
graph 11.561446604452176	standards 4.249803613511391
phenomenon 11.193505722973475	quick 4.222481029944604
solution 10.37666928515318	vector 4.206010674965926
dogs 10.001659554291132	height 4.197924388435879
keeping 9.27521712306013	comparisons 4.1523385028697035
microscopy 9.141299193930774	condition 4.089620994215543
social 8.983276536665404	handler 4.074756462254946
reverse 7.652719460990139	scheme 4.0498042627229225
report 7.300666700166849	bioactive 4.035440584463992
observations 7.249400509258114	fibrin 4.003631440077767
afilm 7.1154164939908835	visits 3.9782090974041444
piece 6.9822312190733244	institute 3.9012860204810664
calcif 6.209398496240602	jan 3.88268319940397
rats 6.209398496240602	diagnosis 3.8785566003174985
folder 6.1353980254123375	multivariate 3.8726346348228766
messages 5.890612407519004	iliac 3.8130624488682834
probing 5.84845315692188	principal 3.7136842105263153
training 5.82383645063071	guidelines 3.7085203992849567
paper 5.721835781690826	sutures 3.70199146514936
biopsies 5.690967283072546	threaded 3.6982144897755234
ieee 5.679356385604476	sun 3.6280163950598427
border 5.576773835449721	manuscript 3.5932741877472725
override 5.257561518323167	anti 3.591541353383459
economic 5.067475843886294	remove 3.5878733997155052
icon 5.065663441022656	strong 3.583273251622206
neutrophil 4.855384765054967	interoperability 3.5214328925762683
completion 4.826892614842046	architectural 3.5144304791830323
resource 4.789269877853397	trans 3.482255639097745
equivalent 4.736541353383459	impulse 3.478322429450249
fluid 4.716752602393228	vibration 3.478322429450249
theoretical 4.643443023142271	role 3.456069366354095

Рис. 9. Перелік слів із найбільшими дисперсіями

Для високої ефективності категоризації із цих слів було вибрано 500 ключових. Було створено карту Кохонена із 61 елементом та прямокутною ґраткою. Навчання тривало 61000 ітерацій із такими параметрами:

$\sigma_0 = 5$ – початкова ефективна ширина функції сусідства ;

$\tau_1 = 37342.23$ – швидкість спадання ефективної ширини σ ;

$\eta_0 = 1$ – початкове значення функції навчання ;

$\tau_2 = 13050.55$ – швидкість спадання функції навчання η .

Далі зображено графіки зміни *MSE* і *TE* упродовж процесу навчання кожні 100 ітерацій.

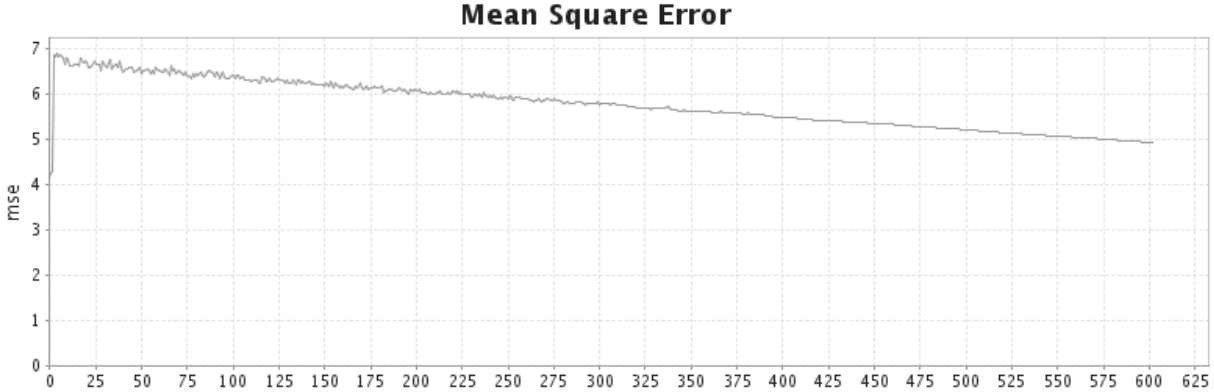


Рис. 10. Графік зміни MSE упродовж навчання кожні 100 ітерацій

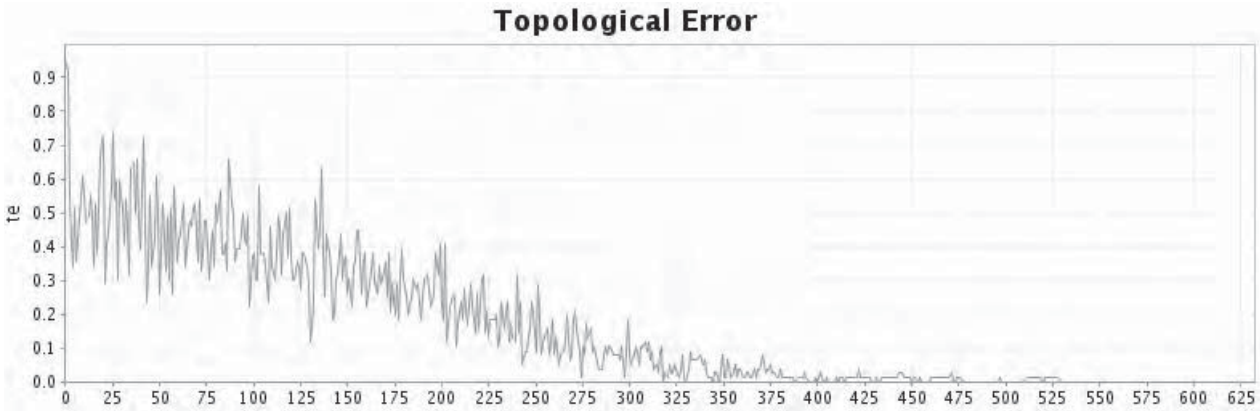
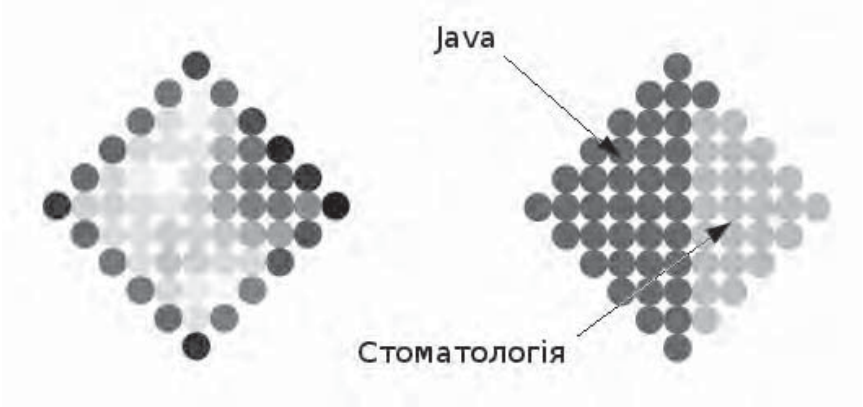


Рис. 11. Графік зміни TE упродовж навчання кожні 100 ітерацій

Якщо TE дорівнює нулю, це свідчить про добре топологічне впорядкування елементів карти впродовж начання, яке відповідає топології вхідного простору документів.

На рис. 12 подано зображення U-Matrix для навченої карти, та зображення карти із чіткими межами кластерів, визначеними за допомогою запропонованого алгоритму.



а *б*

Рис. 12. U-Matrix (а) та карта із застосованою до неї UPGMA (б)

Відсоток успішності категоризації, тобто відношення між документами, що потрапили у свою категорію, до загальної кількості документів, становить 98,7 % (один із документів не потрапив у свою категорію), що можна вважати дуже успішним результатом.

Експеримент №2. У *другому експерименті* сукупність складалась з 110 документів, серед яких були документи із трьох сфер: книги про програмування мовою Java, статті зі стоматології та художні твори. До першої категорії належав 41 документ, до другої – 35, до третьої – 33. Як і в попередньому експерименті, було вилучено всі слова з кількістю літер, меншою від 3, після чого з документів вибирались слова із частотою, більшою від 0.001. У результаті цього було отримано 4699 різних слів. Варто зазначити, що в процесі вилучення слів із малою частотою було вилучено 40514 слів. Після цього для кожного слова було визначено величину відносної середньої абсолютної девіації. Далі наведено перелік ключових слів із найбільшими дисперсіями.

cir 18.06002464688596	ground 4.538005948532264
patol 18.06002464688596	autologous 4.524298392899976
native 15.735360786897614	quantify 4.493818181818181
jane 14.043426665301665	dentist 4.4349431818181815
graph 11.569580575744041	cases 4.429818181818182
fiber 11.138475405614216	animals 4.41974451196825
solution 10.384803256445048	specialist 4.368214708590974
keeping 9.283351094351996	vector 4.214144646257791
microscopy 9.149433165222637	cone 4.212587412587413
polly 9.136581718487134	height 4.210125345373677
social 8.99141050795727	comparisons 4.160472474161569
reverse 7.660853432282003	advantages 4.112252964426878
report 7.312867657104648	potentially 4.08684101286841
observations 7.257534480549981	login 4.0836837704243685
efilm 7.123550465282748	scheme 4.057938234014789
piece 6.99036519036519	bioactive 4.04357455575586
folder 6.143531996704203	fibrin 4.011765411369634
messages 5.898746378810871	visits 3.9863430686960095
probing 5.856587128213746	institute 3.9094199917729338
newly 5.770707070707071	jan 3.8908171706958363
paper 5.729969752982692	diagnosis 3.8866905716093645
ieee 5.68749035689634	multivariate 3.8807686061147426
border 5.584907806741587	iliac 3.8211964201601503
hash 5.361283778372942	ghost 3.755821136027931
effective 5.301732421477644	principal 3.721818181818182
platelet 5.210523459966214	guidelines 3.716654370576823
economic 5.075609815178161	royal 3.716193181818182
asynchronous 5.046941304011151	sutures 3.7101254364412264
neutrophil 4.8635187363468315	absolute 3.6886472379295343
click 4.776563542197985	nasal 3.6804626030903407
equivalent 4.744675324675325	sighed 3.6441896691382634
fluid 4.724886573685094	remove 3.5960073710073712
theoretical 4.651576994434137	strong 3.591407222914072

Рис. 13. Перелік слів із найбільшими дисперсіями

Кількість ключових слів, вибраних для подання документів, становила 500, кількість елементів – 61, ґратка – прямокутна.

$\sigma_0 = 5$ – початкова ефективна ширина функції сусідства;

$\tau_1 = 37342.23$ – швидкість спадання ефективної ширини σ;

$\eta_0 = 1$ – початкове значення функції навчання;

$\tau_2 = 13050.55$ – швидкість спадання функції навчання η.

Далі зображено графіки зміни *MSE* і *TE* упродовж процесу навчання кожні 100 ітерацій (рис. 15, 16).

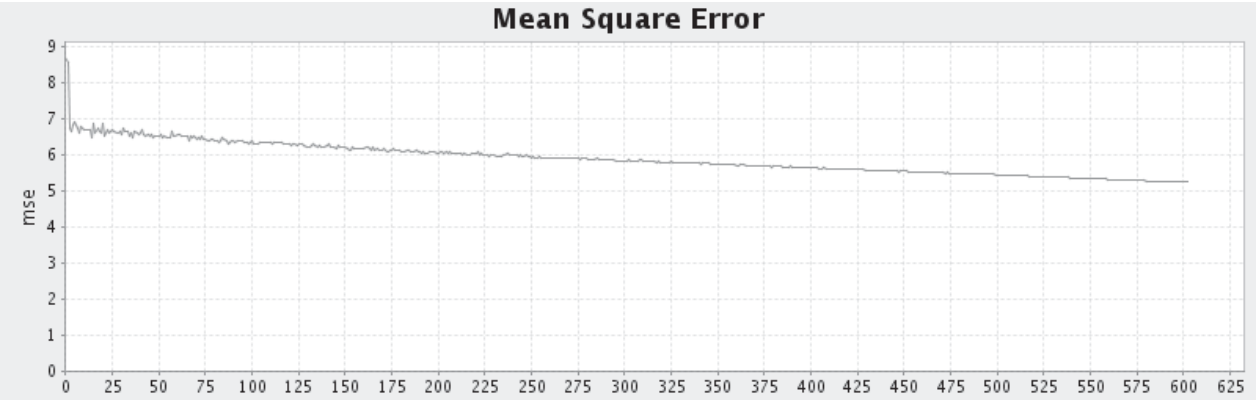


Рис. 14. Графік зміни MSE упродовж навчання кожні 100 ітерацій

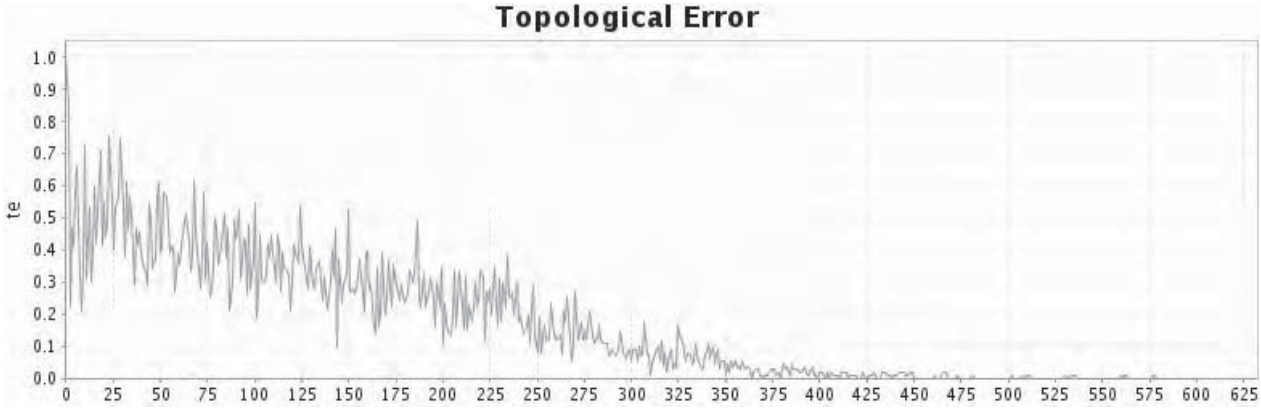
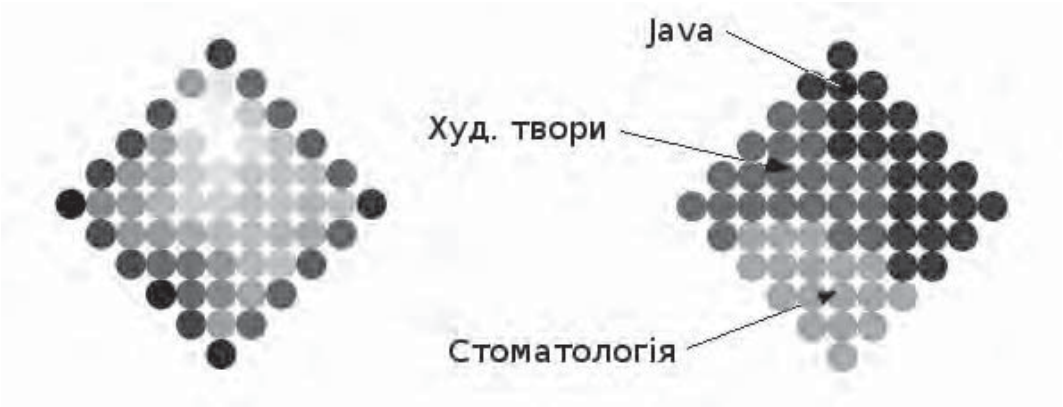


Рис. 15. Графік зміни ТЕ упродовж навчання кожні 100 ітерацій

На рис. 16 наведено U-Matrix для навченої карти та зображення карти із чіткими межами кластерів.



а *б*

Рис. 16. U-Matrix (а) та карта із застосованою до неї UPGMA (б)

Відсоток успішності категоризації становить 91,8 % (дев'ять документів не потрапили у свої категорії), що, як і в попередньому експерименті, також можна вважати досить успішним результатом.

Експеримент № 3. У третьому експерименті сукупність складалась з 151 документів, серед яких були документи із чотирьох сфер: книги про програмування мовою Java, статті зі стоматології, художні твори та книги з області штучного інтелекту. До першої категорії належав 41 документ, до другої – 35, до третьої – 33, до четвертої – 41. Як і в попередньому експерименті, було вилучено всі слова з кількістю літер, меншою від 3, після чого з документів вибирались слова із частотою, більшою від 0.001. У результаті цього було отримано 6213 різних слів. Варто зазначити, що в процесі вилучення слів із малою частотою було вилучено 47118 слів. Після цього для кожного слова було визначено величину відносної середньої абсолютної девіації. Далі наведено перелік ключових слів із найбільшими дисперсіями.

Рис. 17. Перелік слів із найбільшими дисперсіями

Кількість ключових слів, вибраних для подання документів, становила 500, кількість елементів – 61, ґратка – прямокутна.

$\sigma_0 = 5$ – початкова ефективна ширина функції сусідства;

$\tau_1 = 37342.23$ – швидкість спадання ефективної ширини σ;

$\eta_0 = 1$ – початкове значення функції навчання;

$\tau_2 = 13050.55$ – швидкість спадання функції навчання η.

Графіки зміни *MSE* і *TE* упродовж процесу навчання кожні 100 ітерацій наведено на рис. 18, 19.

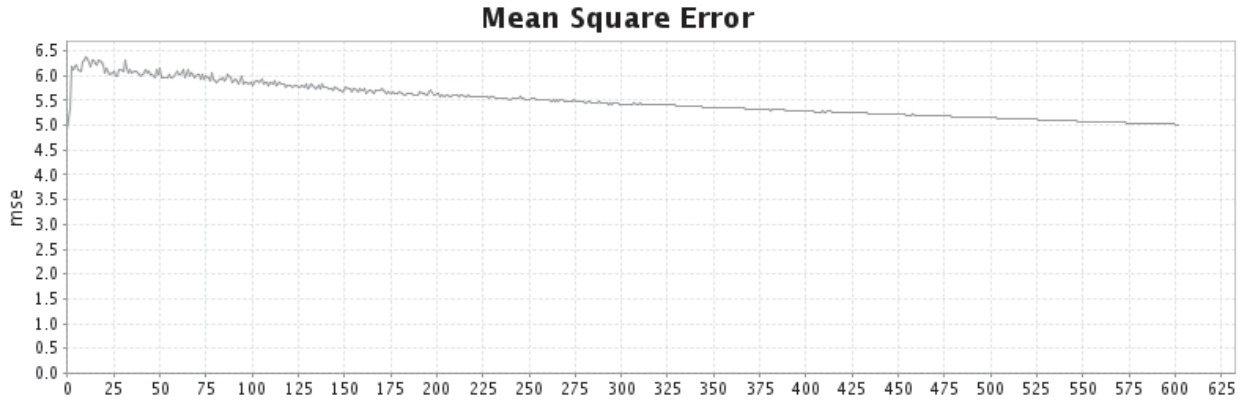


Рис. 18. Графік зміни MSE упродовж навчання кожні 100 ітерацій

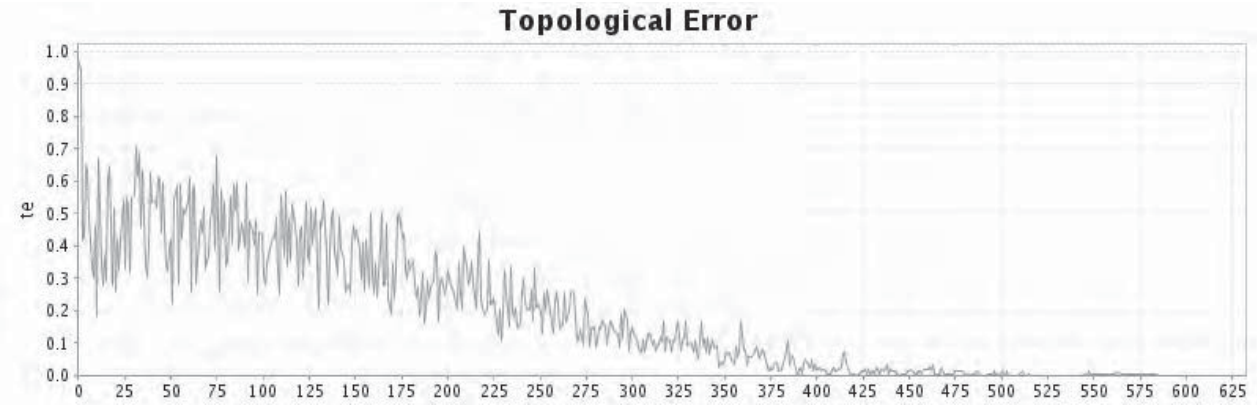
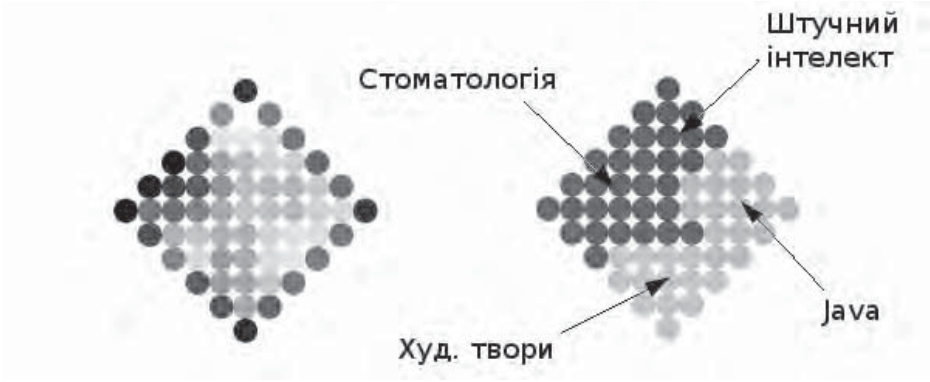


Рис. 19. Графік зміни TE упродовж навчання кожні 100 ітерацій

Якщо TE дорівнює нулю, це свідчить про добре впорядкування елементів карти. На рис. 20 подано U-Matrix для навченої карти та зображення карти із чіткими межами кластерів, визначеними за допомогою UPGMA.



а *б*

Рис. 20. U-Matrix (а) та карта із застосованою до неї UPGMA (б)

Відсоток успішності категоризації становить 74,8 % (38 документів не потрапили у свої категорії). Проблеми здебільшого виникали при розділенні категорій книг з області штучного інтелекту та книг про програмування). Цей результат можна вважати допустимим з огляду на складні взаємозв'язки між двома категоріями. Використання більшої карти могло би поліпшити якість розпізнавання.

Висновки

Запропонований метод категоризації дав доволі успішні результати під час категоризації порівняно малих сукупностей даних. Зазначимо, що за більших сукупностей документів категоризацію виконувати легше, оскільки навчальна множина представлена повніше. Основними перевагами цього підходу є:

– *швидкодія*, яка забезпечується простими обчисленнями під час визначення ключових слів та значним скороченням розмірності зі збереженням даних, необхідних для категоризації;

– *інтуїтивність результатів*, яка забезпечується завдяки поєднанню карт Кохонена та UPGMA для візуалізації утворених категорій;

– *можливість вибору різних параметрів*, що уможливлює знаходження компромісу між швидкодією та якістю категоризації;

– *незалежність від мови* – цей метод не використовує жодних лінгвістичних методів, які б узалежнювали його від мови документа.

Недоліки цього методу такі:

– трактує синоніми як різні слова, що істотно знижує якість;

– трактує одне й те саме слово в різних формах, як різні слова, що також негативно впливає на якість категоризації;

– *програє в якості* порівняно з WEBSOM.

Надалі планується:

– дослідити ефективність цього методу під час категоризації документів інших мов (наприклад, німецької);

– застосувати міркування, викладені в цій роботі, до так званого WEBSOM (двошарового SOM для кластеризації документів);

– розробити програмне забезпечення для категоризації документів на персональних комп'ютерах і локальних мережах.

1. Pöllä M. An Analysis of Interdisciplinary Text Corpora // M. Pöllä, T. Honkela, H. Bruun, A. Russell // Proceedings of The Ninth Scandinavian Conference on Artificial Intelligence (SCAI 2006), October 25-27, Helsinki, 2006. 2. Li Wentian. Random Texts Exhibit Zipf's-Law-Like Word Frequency Distribution / Wentian Li // IEEE Transactions on Information Theory, 1992 – Vol. 38 Issue 6. P. 1842–1845. 3. Haykin Simon. Neural Networks: A Comprehensive Foundation / Simon Haykin. – 2nd ed. – Pearson Education, Ninth Indian Reprint, 2005. 4. Kohonen Teuvo. Self-Organizing Maps / Teuvo Kohonen. – 3. ed. – Berlin; Heidenberg; New York; Barcelona; Hong Kong; London; Milan; Paris; Singapore; Tokyo: Springer, 2001. 5. Huang Shiping. Exploration of Dimensionality Reduction for Text Visualization / Huang Shiping, Matthew O. Ward, Elke A. Rundensteiner // Technical report Computer Science Department Worcester Polytechnic Institute, 2003. 6. Hodych O. Determining cluster boundaries within Self-Organizing Maps / O. Hodych, I. Nikolski, V. Pasichnyk, Yu. Shcherbyna // Вісник Національного технічного університету „Харківський політехнічний інститут”. – Харків, 2007. – № 5. – С. 97–109. 7. Сеньо П.С. Теорія ймовірностей та математична статистика / П.С. Сеньо. – К.: Центр учбової літератури, 2004. – 448 с. 8. Годич О.В. Застосування штучної нейронної мережі типу SOM для розв’язування задачі діагностування / О.В. Годич, Ю.В. Нікольський, Ю.М. Щербина. // Вісник Нац. ун-ту “Львівська політехніка”. – Львів, 2002. – № 464: Інформаційні системи та мережі. – С. 31–43. 9. Годич О. В. Самоорганізація нейромереж та класифікація даних / О.В. Годич, Ю.М. Щербина // Вісник Львівського ун-ту ім. І. Франка. – Львів, 2003. – № 7: Прикладна мат. та інформ. – С. 234–247. 10. Годич О.В. Навчання {SOM} методом нейтронної міграції / О.В. Годич // Вісник Нац. ун-ту “Львівська політехніка”. – Львів, 2004. – № 519: Інформаційні системи та мережі. – С. 55–72. 11. Hodych O. Synthesis of Self-Organizing Map and Feedforward Neural Network for Better Forecasting / O. Hodych, Yu. Shcherbyna, M. Zylan // International Journal of Computing. – Ternopil, 2004. – Vol 3, № 3. – P. 68–75. 12. Пасичник В.В. Исследование эффективности алгоритмов обучения нейросетей Кохонена / В.В. Пасичник, О.В. Годыч, Ю.В. Никольский, Ю.Н. Щербина // Управляющие системы и машины. – К., 2006. – № 2. – С. 63–80. 13. Годич О. Аналіз структури медичних даних із застосуванням мереж Кохонена / О. Годич, Ю. Нікольський, В. Пасічник, Ю. Щербина // International Journal of Computing. – Ternopil, 2007. – Vol 6, № 3. – P. 124–136. 14. Ultsch A. Self-Organizing Neural Networks for Knowledge Acquisition / A. Ultsch // In Proc. of the 10th ECAI, Vienna, Austria, 1992. – P. 208–210.