УДК 004.853, 004.55 https://doi.org/10.15407/pp2019.01.057

Ю.В. Рогушина

ЗАСОБИ ТА МЕТОДИ АНАЛІЗУ НЕСТРУКТУРОВАНИХ ДАНИХ

Проаналізовано сучасні засоби аналізу неструктурованих даних (НСД) та вплив Big Data на актуальність цього напрямку досліджень. Розглянуто перспективи використання фонових знань для такого структурування. Обґрунтовано доцільність застосування для цього таких стандартів W3C, як RDF та OWL. Використання семантичних Wiki-технологій для створення розподілених інформаційних ресурсів не тільки дозволяє досить легко додавати структурування до НСД, але й є джерелом фонових знань для аналізу довільних природномовних текстів відповідної предметної області. Запропоновані в роботі моделі та методи дозволяють вдосконалити процес генерації таких знань.

Ключові слова: неструктуровані дані, Text Mining, онтологія, Semantic Web, Wiki.

Вступ

На даний час світовим співтовариством вже усвідомлений головний напрямок у боротьбі з інформаційним вибухом – перехід від збереження й обробки даних до накопичення й обробки знань. Тому виникає потреба у засобах та методах здобуття знань з тих даних, що генеруються в процесі діяльності людства та можуть бути корисними для подальшого використання. Актуальність проблеми загострюється через стрімке поширення Big Data, яке викликає потребу в нових, більш ефективних методах аналізу розподілених та гетерогенних даних.

Обробка великих обсягів інформаційних ресурсів різного походження та з наперед не відомими моделями даних (в такому випадку говорять про неструктуровані дані), для яких не придатні традиційні СКБД, потребує спеціалізованих засобів їх представлення та аналізу.

Ще у 1998 році аналітики з Merrill Lynch сформулювали емпіричне правило: біля 80 % – 90 % всієї потенційно корисної ділової інформації генерується в неструктурованій формі [1]. Прогнозується, що до 2025 року глобальна датасфера зросте до 163 зетабайт, і 70 % – 80 % її буде неструктурованою.

Визначення НСД

НСД – дані, для яких не визначені окремі елементи, їх властивості, можливі значення та спосіб їх кодування.

НСД – це інформація, яка не має попередньо визначеної моделі даних або не організована заздалегідь. Це призводить до проблем, пов'язаних з її зберіганням (традиційні БД не розраховані на таку невизначеність) та аналізом. Саме НСД потенційно мають найбільшу цінність як джерела нових знань: чим більше даних доступних для аналізу, тим точніші результати. Прикладами НСД можуть бути книги, журнали, документи, метадані, медичні записи, аудіо, відео, аналогові дані, зображення, файли та неструктурований текст, наприклад, тіло повідомлення електронної пошти, Web-сторінки або слова документ процесора.

Сьогодні у більшості випадків під НСД розуміють текстову інформацію – набори слів природної мови (ПМ) довільної довжини, поєднані за слабо формалізованими лінгвістичними правилами та представлені в електронній формі. Це пояснюється тим, що саме текстова інформація містить найбільш корисні для подальшого використання відомості. Такі НСД можуть містити також дати, числа тощо. Приклади текстових НСД:

- електронна пошта;
- ПМ-документи в різних форматах;
- відомості з соціальних мереж (YouTube, Facebook, Twitter, LinkedIn, Flickr тощо);
- дані з мобільних пристроїв (текс-

тові повідомлення й інформація про місце розташування) та Інтернету речей;

- контент Web-сайтів.

Найбільш поширені приклади НСД інших типів [2] – це потокове відео, інформація від супутників, дані радарів чи сонарів. Засоби аналізу таких НСД значно більш спеціалізовані.

Іноді досить складно відрізняти структуровані та НСД. Один з критеріїв визначення структурованості даних – для елемента таких даних можна створити синтаксичний аналізатор. Термін НСД не є точно визначеним з декількох причин [3]:

- структура може міститися у таких даних, але не мати формального визначення;
- дані, що мають певну структуру, можуть бути охарактеризовані як неструктуровані, якщо ця структура не є корисною для цілей їх обробки;
- неструктурована інформація може мати певну структуру (бути слабо структуровану або навіть структурованою), яка не може бути застосована для автоматизованої обробки без додаткових уточнень.

Таким чином, дані розглядаються як НСД у тих випадках, коли відомості про їх структуру не можуть зробити аналіз даних більш ефективним.

Неструктурована інформація може зберігатися у формі об'єктів (файлів чи документів), що самі мають структуру. Наприклад, тіло листа або вкладення до електронної пошти – це неструктуровані дані, але їх місцезнаходження в пошті задається її структурою. Сполучення структурованих і неструктурованих даних у сукупності також є НСД.

Властивості НСД

НСД, на відміну від структурованих даних, які здебільшого не мають антропогенних особливостей, досить часто створюються безпосередньо людьми, і тому системи обробки та аналізу НСД мають враховувати «людський фактор».

Властивості НСД:

- *гетерогенність*. Для НСД існує величезна кількість різних способів створення, джерел інформації та причин, через які ці дані не можуть бути структуровані і поміщені в будь-яку СКБД, а лише у файли різноманітних форматів (приклад – наукові статті мають певну структурованість та обов'язкові елементи, але їх неможливо представити інакше, як файли текстових редакторів);
- *неоднозначність*. Висловлення двох осіб, що збігаються дослівно, можуть мати різний зміст у залежності від досвіду, поглядів тощо, а та сама ідея може бути виражена різними словами (наприклад, твердження експерта "я не зрозумів цю статтю свідчить про низьку якість статті, а те саме твердження студента – про його низьку освіту);
- *контекстна залежність.* Те саме слово чи ім'я можуть у різних умовах інтерпретуватися по-різному ("модель" у техніці та у математиці мають різне значення);
- *динаміка значення.* Слова можуть дуже швидко змінювати свій зміст, наприклад, назва нікому раніше не відомого населеного пункту через події, що відбувалися в ньому, може стати загальновідомою та отримати додаткове значення;
- *етнокультурна залежність.* У різних етносах і культурах, що використовують ту саму мову, слова можуть набувати різного сенсу і позначати зовсім різне.

Такі технології, як Data Mining, обробка природної мови і Text Mining, надають різні методи для пошуку структури в НСД. Загальні методи структуризації тексту зазвичай включають у себе ручну розмітку метаданими або тегами для подальшого структурування. Стандарт архітектури керування неструктурованою інформацією (Unstructured Information Management Architecture – UIMA) надає загальну основу для обробки цієї інформації для здобуття сенсу та створення структурованих даних.

Історія виникнення аналізу НСД

Найбільш ранні дослідження Business Intelligence (BI) зосереджувалися саме на неструктурованих текстових даних, а не на числових даних [4]. Проте ли-

ше на початку 21 століття технології наздогнали наукові дослідження. Поява Big Data наприкінці 2000-х років викликала підвищений інтерес до застосування неструктурованих аналітичних даних.

У 80-х і 90-х роках 20 ст. бізнес-аналітика (OLAP, інтелектуальний аналіз даних, ETL та сховище даних) була зорієнтована на структуровані числові дані, що зберігалися в реляційних базах даних.

Виділення аналізу НСД (UDA – unstructured data analysis) в окремий науково-технічний напрямок датується початком 2000 років, коли аналітики Gartner опублікували інформацію про високі затрати часу та праці на обробку даних – рутинна, не автоматизована робота з контентом займала до половини робочого часу. Незручність була пов'язана саме з необхідністю обробки текстових НСД у різних форматах: електронних листів, службових записок, новин, чатів, звітів, маркетингових матеріалів, презентацій тощо, які не можливо було занести до реляційних СКБД (деякі з таких даних є слабо структурованими або квазіструктурованими та супроводжуються метаданими – автор, місце створення, розмір – які можна помістити до СКБД).

На сьогоднішній день НСД складають найбільшу частку даних, що зберігаються (понад 80 % усіх збережених даних, а їхня кількість зростає на порядок швидше в порівнянні з структурованими даними), тому методи та засоби їх використання швидко розвиваються. Ці методи спрямовані на перетворення цих даних на структуровану інформацію, яка може використовуватися різними способами.

Text Mining як окремий напрямок з'явився наприкінці 1990-х років 20 ст. Ранні підходи розглядали текст як "мішок слів" (“bag of words”), таких як абревіатури, множини і сполучення, а також терміни з декількох слів, відомі як n-грами. Основний лексичний аналіз може враховувати частоти слів і термінів для виконання елементарних функцій, таких як спроби класифікувати документи за темами. Але не було можливості зрозуміти семантику документів. Нині Text Mining шукає приховані відношення та інші складні структури в наборах текстових даних.

Text Mining як основа аналізу неструктурованої текстової інформації

Аналіз текстових даних як технологія базується на лінгвістиці та інтелектуальному аналізі даних, що спочатку застосовувалися в аналітиці для розпізнавання в тексті особистих і географічних назв, дат, телефонних номерів та адреси електронної пошти. Більш складні методи дозволяли знаходити поняття і відношення між ними та навіть настрої.

Додання структури до НСД є складною науковою проблемою, якій приділяють увагу науковці на протязі довгого часу [5]. Актуальність проблеми збільшилася з поширенням Big Data. У найбільш узагальненому вигляді розв’язок проблеми пов’язують з побудовою розміченого графу, що відповідає вмісту НСД, та із співставленням таких графів. Інший аспект цієї проблеми пов’язують із знаходженням релевантних знань, з якими співставляють НСД.

Методи Data Mining включають класифікацію, кластеризацію, аналіз зв'язків, дерева рішень тощо [6]. Інтелектуальне моделювання використовується для таких бізнес-функцій, як оцінка кредитів, виявлення ризиків, виявлення шахрайства та прогнозування для прогнозування тенденцій залежної від часу інформації. Всі ці методи Data Mining можуть бути пристосовані до даних, отриманих з текстових джерел – наприклад, необхідно знизити високу розмірність текстової інформації. Дослідники використовують для вирішення цих питань статистичні методи (такі як розкладання сингулярних значень і векторні машини підтримки для зменшення розмірності) у поєднанні з алгоритмами машинного навчання (деревами рішень, нейронними мережами тощо) і більш глибокою лінгвістикою, що підтримує такі функції, як використання контексту для визначення семантичної неоднозначності.

Text Mining можна визначити як процес здобуття знань з колекції ПМ-документів за допомогою набору ін-

струментів для їх аналізу [7]. Аналогічно до Data Mining, засоби Text Mining прагнуть здобути з даних потрібну для діяльності користувача інформацію. У випадку Text Mining джерела даних – це колекції документів, і цікаві для користувачів шаблони потрібно знайти не серед формалізованих записів бази даних, а в неструктурованих текстових даних у документах цих колекцій.

Text Mining можна розглядати як окремий випадок Data Mining. Тому не дивно, що системи Text Mining та Data Mining мають багато подібностей в архітектурі. Наприклад, обидва типи систем використовують процедури попередньої обробки, алгоритми виявлення шаблонів і засоби візуалізації результатів для покращення перегляду наборів відповідей. Text Mining використовує багато специфічних типів моделей у своїх основних операціях виявлення знань, які були впроваджені та перевірені в дослідженнях Data Mining.

Оскільки Data Mining припускає, що дані зберігаються у структурованому форматі, попередня обробка в ньому фокусується на задачах очищення та нормалізація даних і створення великої кількості об'єднань таблиць. На відміну від цього, для Text Mining операції з попередньої обробки пов’язані з ідентифікацією та пошуком репрезентативних властивостей для документів, поданих природною мовою (ПМ). Ці операції попередньої обробки забезпечують перетворення НСД, що зберігаються в колекціях документів, в більш чітко структурований проміжний формат. Тому Text Mining також спирається на досягнення в інших дисциплінах, пов’язаних з обробкою природної мови: методи інформаційного пошуку, здобуття інформації та комп’ютерної лінгвістики на основі корпусу (рис. 1).

Для продуктивного здобуття корисних відомостей з даних, що містять «людську інформацію», крім пошуку, застосовують технології Text Mining, спеціалізовані на обробці ПМ. Уперше термін Text Mining було використано в 1995 році як альтернатива терміну «здобуття знань з тексту» (Knowledge Discovery from Text, KDT).

Складові Text Mining

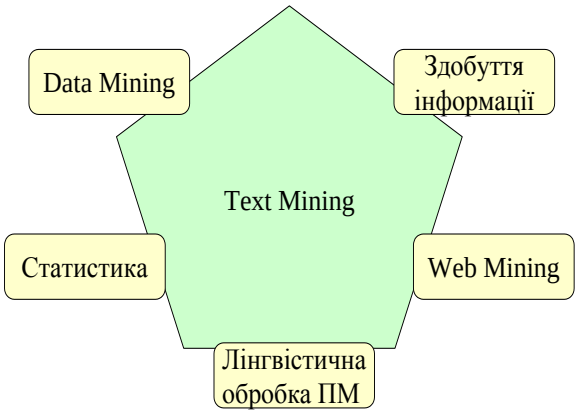

Рис. 1. Складові Text Mining

Text Mining має забезпечити перехід від НСД до структурованих з наступним аналізом. Найчастіше в цьому процесі ігнорується велика частина специфічних особливостей ПМ, які застосовуються тільки на попередньому етапі розбору текстів, а на наступних використовується модель «мішка слів», у якій не важливий порядок слів.

Emanu Text Mining. Потреба в технологіях Text Mining загострилася, коли кількість текстів стала перевищувати можливості сприйняття людиною та виникла потреба в автоматизації здобуття їх змісту. На рис. 2 показана узагальнена схема процесу Text Mining. На етапі попередньої обробки НСД перетворюються в структуровану інформацію, в якої потім виділяються істотні ознаки – атрибути та здійснюється їх дослідження.

Автори книги «Вступ до неструктурованих даних» (2007) (“Tapping into Unstructured Data”) Бів Інмон та Ентоні Несвіч, аналізуючи зв’язок між Business Intelligence та Text Mining у другій частині «Інтегрування неструктурованих даних у текстову аналітику і BI» (“Integrating Unstructured Data and Textual Analytics into Business Intelligence”), поділяють Text Mining на два напрямки: «виявлення» (Discovery) – дедуктивні методи підтвердження або спростування гіпотез та «аналіз» (Analysis) – статистика, кластеризація тощо.

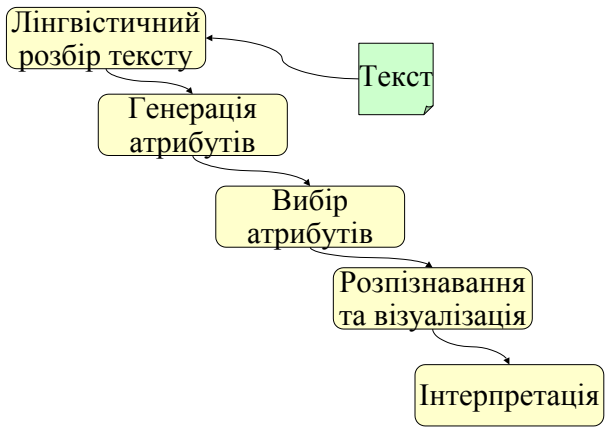

Рис. 2. Етапи Text Mining

Основні елементи Text Mining. Ключовим елементом Text Mining є колекція документів. У найпростішому випадку це довільна група текстових документів. Більшість рішень Text Mining спрямовані на виявлення моделей (шаблонів) у дуже великих колекціях документів, у тому числі – у сховищах Big Data.

Колекції документів можуть бути *статичними*, тобто початковий набір документів залишається незмінним, або *динамічними*, тобто до початкового набору можуть додаватися нові документи, а існуючі – оновлюватися.

Якщо колекція документів має великий розмір та швидко змінюється, то ручні спроби її аналізу не є ефективними. Автоматичні методи виявлення та вивчення взаємозв'язків між документами різко підвищують швидкість та ефективність дослідницької діяльності, але їх неможливо застосовувати на непідготовлених колекціях документів.

Документ – ще один основний елемент Text Mining. Це одиниця дискретних текстових даних в колекції, може корелюватися з деякими документами реального світу, такими як звіт, електронна пошта, стаття, прес-реліз або новини.

Документ може одночасно входити до різних колекцій документів або різних підмножин однієї колекції.

Елементи структурування текстових документів. Такі НСД, як текстовий документ, з деяких точок зору можна розглядати як структурований об'єкт. Наприклад, з лінгвістичної точки зору кожен документ містить велику кількість семантичної та синтаксичної структури, яка прихована в тексті. Крім того, елементи розмітки, такі як знаки пунктуації, великі літери, числа та спеціальні символи, та елементи форматування (таблиці, стовпці, параграфи тощо) можуть розглядатися як мова «м'якої розмітки», що допомагає визначити важливі підкомпоненти документів – назву, імена авторів, підрозділи тощо. Послідовність слів також може бути структурно значущим виміром документа. Крім того, деякі текстові документи можуть містити вбудовані метадані у вигляді формалізованих тегів розмітки, які автоматизовано генеруються текстовими редакторами.

Документи, які мають відносно мало таких елементів структурування (наприклад, наукові публікації та бізнес-звіти), називають *вільно форматованими (free- format*) або *слабо структурованими* (*weakly structured*). Документи з відносно більшою кількістю елементів структурування (наприклад, електронна пошта, Web-сторінки HTML) називають *частково структурованими* (*semistructured*) .

Операції попередньої обробки дозволяють використовувати в Text Mining багато різних елементів, що містяться в ПМ-документі для його перетворення з НСД з неявним структуруванням в явно структуровані дані. Однак, з огляду на потенційно велику кількість слів, фраз, речень та елементів форматування, які може мати навіть невеликий документ, навіть не враховуючи потенційно велику кількість різних значені, які кожен із цих елементів може мати в різних контекстах і комбінаціях, найважливішим завданням для більшості систем Text Mining є ідентифікація спрощеної підмножини властивостей (ознак) документів. Такий набір ознак називають *репрезентативною моделлю* документа: окремі документи характеризуються за допомогою наборів ознак, які містять їхні репрезентативні моделі. Але слід враховувати, що навіть у найбільш ефективних репрезентативних моделях кожен окремий документ у колекції має надзвичайно велику кількість властивостей.

Тому проблеми, пов'язані з високою розмірністю характеристик (тобто розміром і масштабом можливих комбінацій значень ознак для даних), зазвичай мають значно більше значення в системах Text Mining, ніж у класичних системах Data Mining.

Структуровані представлення ПМ-документів мають набагато більшу кількість потенційно репрезентативних ознак – і, отже, більшу кількість можливих комбінацій їх значень – ніж в реляційних або ієрархічних базах даних. Наприклад, у відносно невеликій колекції з 10–15 000 документів, можна виявити більше 25 000 нетривіальних слів. Навіть якщо працювати з більш оптимізованими типами властивостей, десятки тисяч ознак, пов’язаних з різними поняттями. можуть бути актуальними для однієї предметної області (ПрО). Кількість атрибутів у реляційній базі даних, які аналізуються в задачі інтелектуального аналізу даних, зазвичай значно менше. Висока розмірність потенційно репрезентативних властивостей спонукає до попередньої обробки тексту, спрямованої на створення спрощених моделей подання.

Ще однією характеристикою ПМ-документів є *розрідженість властивостей* (*feature sparsity)* – лише невелика частка всіх властивостей, можливих для колекції документів у цілому, з'являється в кожному окремому документі, і, таким чином, коли документ представляється у вигляді бінарного вектора ознак, майже всі значення вектора дорівнюють нулю.

Розмір кортежу також розріджений. Тобто деякі функції часто з'являються лише в декількох документах, а це означає, що підтримка багатьох моделей досить низька.

Властивості окремого ПМ-документа – це символи, слова, терміни і поняття. Оскільки алгоритми Text Mining обробляють представлення документів через набір властивостей, а не безпосередньо самі документи, виникає потреба у компромісі між двома важливими цілями.

Перша ціль полягає у тому, щоб досягти правильної класифікації обсягу і семантичного рівня властивостей для точного відображення значення документа в процесі виконання операції попередньої обробки тексту. Друга ціль – вибрати таке визначення властивостей, що є найбільш обчислювально ефективним і практичним для виявлення шаблонів. Такий вибір може підтримуватися валідацією, нормалізацією або посиланням на властивості з контрольованих словників або зовнішніх джерел знань, таких як словники, тезауруси, онтології або бази знань, щоб допомогти у створенні менших наборів властивостей з більшою семантичною значимість.

Хоча для представлення ПМ-документів можна використовувати багато потенційних властивостей, найчастіше використовуються такі чотири типи.

- *Символи*. Букви, цифри, спеціальні символи та пробіли є будівельними блоками семантичних ознак вищого рівня, таких як слова, терміни та поняття. Представлення на рівні символів може включати повний набір всіх символів для документа або деякого фільтрованого піднабору. Представлення на основі символів без інформації щодо позицій (тобто підходи з “мішком символів” – “bag-of-characters “) зазвичай мають дуже обмежену корисність для Text Mining. Представлення, які включають певний рівень позиційної інформації (наприклад, біграми або триграми) дещо корисніші.

- *Слова*. Конкретні слова, вибрані безпосередньо з ПМ-документа, є базовим рівнем для семантики. Одна властивість на рівні слів повинна мати значення не більше одного лінгвістичного маркера. Фрази та багатослівні вирази не складають окремих властивостей на рівні слів. Представлення документа на рівні слів може включати в себе ознаки для кожного слова в цьому документі, тобто текст документа представляється повним набором властивостей рівня слова. Це може призвести до того, що деякі представлення колекцій документів на рівні слів містять десятки або сотні тисяч унікальних слів у своєму просторі ознак. Проте, більшість представлень документів на цьому рівні демонструють принаймні деяку мінімальну оптимізацію і тому складаються з підмножин репрезентативних властивостей, які фільтруються

від таких елементів, як стоп-слова, символи та беззмістовні числа.

- *Терміни* – це окремі слова та багатослівні фрази, вибрані безпосередньо з корпусу вихідного документа за допомогою методології вилучення термінів. Функції на рівні термінів, у сенсі цього визначення, можуть бути складені тільки з конкретних слів і виразів, знайдених у рідному документі, для якого вони мають бути загалом репрезентативними. Отже, представлення документа на основі термінів обов'язково складається з підмножини термінів у цьому документі. Наприклад, якщо документ містив речення. Існують різні методології видобування термінів, які можуть конвертувати необроблений текст документа в послідовність нормалізованих термінів (токенізованих і лематизованих форм слова), помічених тегами відповідних часток мови. Іноді для нормалізації термінів також використовується зовнішній лексикон для забезпечення контрольованого словника. Методики видобуття термінів використовують різні підходи для генерування та фільтрації списку найбільш значущих термінів документа з цього набору нормалізованих термінів.

- *Поняття* – це властивості, створені для документа за допомогою різних методик категоризації. Властивості рівня понять можуть бути створені для документів вручну, але тепер частіше видобуваються з документів за допомогою складних процедур попередньої обробки, які ідентифікують окремі слова, багатослівні вирази, цілі речення або навіть більші синтаксичні одиниці, які потім відносяться до конкретних ідентифікаторів понять.

Багато методологій категоризації включають ступінь перехресного посилання на зовнішнє джерело знань; для деяких статистичних методів цим джерелом може бути просто анотована колекція документів. Для категоризації вручну і на основі правилах перехресні посилання і перевірка перспективних властивостей на рівні понять зазвичай включають взаємодію з зовнішніми БЗ, таким як існуюча онтологія домену, лексика або ієрархія формальних. На відміну від властивостей на рівні слів і термінів, властивості документа на рівні понять можуть складатися з слів, які не містяться у цьому документі.

З чотирьох типів описаних тут ознак терміни та поняття відображають властивості з найбільш виразними рівнями семантичної значущості, тому існує багато переваг для їх використання для представлення документів в Text Mining.

Що стосується загального розміру наборів властивостей, то представлення на основі термінів і понять мають приблизно однакову ефективність, але в цілому набагато ефективніші, ніж моделі документів на основі символів або слів. Представлення на рівні термінів легше згенерувати автоматично з тексту, ніж представлення на рівні понять. Проте представлення на рівні понять набагато корисніше для обробки синонімії та полісемії.

Представлення на основі понять дозволяють використовувати дуже складні ієрархії понять і різноманітні знання про домен, що надаються онтологіями та базами знань. Але представлення на рівні понять мають кілька потенційних недоліків: а) відносна складність застосування евристик під час операцій попередньої обробки, б) залежності багатьох понять від домену.

Використання фонових знань в Text Mining

У системах Text Mining поняття належать не тільки до дескриптивних атрибутів певного документа, а й до доменів (ПрО). ПрО у Text Mining – це спеціалізована область інтересів, для якої можуть бути розроблені спеціальні онтології, лексикони та таксономії.

Системи Text Mining можуть використовувати інформацію з формалізованих зовнішніх джерел знань для цих ПрО, щоб покращити попередню обробку документів та виявлення знань.

Знання ПрО (інша поширена назва – *фонові знання* (*background knowledge*)), можуть бути використані в Text Mining для попередньої обробки для поліпшення здобуття понять. Доступ до фонових знань – хоча і не є абсолютно необхідним для створення ієрархій концепцій в контексті

єдиного документу або збору документів – може відігравати важливу роль у розробці більш значущих, послідовних і нормалізованих ієрархій концепцій.

Text Mining використовує фонові знання більшою мірою Data Mining: властивості не є просто елементами в плоскому наборі, як це часто буває у структурованих даних, тому що вони пов'язуються за допомогою лексиконів і онтологій для підтримки розширених запитів.

Незважаючи на те, що операції попередньої обробки Text Mining відіграють важливу роль у перетворенні неструктурованого вмісту необробленої колекції документів у більш сприйнятливе представлення даних на рівні понять, основна функціональність систем Text Mining полягає в аналізі моделей *спільного виникнення* понять (*"concept co-occurrence"*) в документах колекції. В Text Mining використовуються алгоритмічні та евристичні підходи для розгляду розподілів, наборів, що часто повторюються *("frequent sets"*), та різних асоціацій понять на міждокументному рівні з метою надання користувачеві можливості виявити природу та взаємозв'язки понять, що відображені у колекції в цілому.

Наприклад, у колекції новин велика кількість статей, де йдеться одночасно про подію X та компанію Y, а також статей, де йдеться одночасно про компанію Y та продукту Z, може вказувати на інтерес до зв'язку між X та Z, хоча цей зв'язок не присутній у жодному документі.

У класичному Data Mining фонові знання із зовнішніх джерел використовуються для обмеження пошуку.

Системи Text Mining можуть використовувати інформацію з зовнішніх джерел знань в операціях попередньої обробки текстів і перевірки понять. Крім того, доступ до фонових знань може відігравати важливу роль у розробці змістовних, послідовних і нормалізованих ієрархій понять.

Додаткові знання, крім того, можуть бути використані іншими компонентами системи видобування тексту. Наприклад, одним з найбільш важливих застосувань фонових знань є побудова значущих обмежень для операцій виявлення знань. Аналогічно, фонові знання можуть також використовуватися для формулювання обмежень, які дозволяють користувачам підвищувати гнучкість при перегляді великих наборів результатів або при форматуванні даних для презентації.

Системи Text Mining можуть використовувати фонові знання, представлені у вигляді онтологій ПрО, що описує сукупність всіх важливих для ПрО фактів, класів і відношень між цими класами. Її можна розглядати як словник, побудований таким чином, щоб бути одночасно зрозумілим для людей і придатним для машинної обробки. Онтологія дозволяє визначити відношення часткового порядку між поняттями ПрО.

Один з прикладів онтології, що застосовується в Text Mining, – WordNet. Це розробка Принстонського університету для моделювання ПМ.

Системи розробки тексту також використовують фонові знання, що містяться в лексиконах ПрО. Цей термін близький до поняття тезаурусу.

Лексикон ПрО для онтології О – це кортеж

$$Lex = < S_C, \mathrm{Re}\, f_C >,$$

що складається з множини S_C, елементи якої – назви понять ПрО, а відношення $\mathrm{Re}\, f_C \subseteq S_C \times c$ лексичне посилання для понять, для яких $(c,c) \in \mathrm{Re}\, f_C$ виконується для всіх $c \in C \cap S_C$.

На основі $\mathrm{Re}\, f_C$ можна визначити, що для $s \in S_C$

$$\mathrm{Re}\, f_c(s) = \{c \in C \mid (s,c) \in \mathrm{Re}\, f_C\}.$$

Лексикон, подібний до WordNet, може служити точкою входу для фонових знань. Використовуючи лексикон, система Text Mining може нормалізувати ідентифікатори концепції, доступні для анотування документів у його корпусі під час попередньої обробки. Це дозволяє підтримувати за допомогою онтології, пов'язаної з лексиконом, такі операції, як вирішення синонімії, так здобуття інформації про семантичні відношення між поняттями. Крім то-

го, фонові знання дозволяють задавати параметри (значення атрибутів певного поняття) для пошукового запиту щодо екземплярів цього поняття, та визначати їх взаємини з екземплярами інших понять. Наприклад, можна шукати компанії, визначивши значення таких атрибутів, як продукція та місцезнаходження, або шукати компанії, місцезнаходження яких відноситься до класу “Столиця” .Такі атрибути та відношення мають бути доступні користувачеві у списку вибору при формуванні конкретного запиту. Крім того, це дозволяє визначити у запиті те відношення між поняттями, яке задовольняє користувачів. Наприклад, це дозволяє відокремити покупців продукту X від продавців цього продукту.

Моделі подання структурованих даних та їх використання для НСД

В роботі [8] пропонується опис простору даних, який дозволяє класифікувати моделі даних та засоби їх обробки.

Простір даних DS=<DB, DW, ODW, Wb, Nd, Gr, Int, Se, Wo, EM> – це множина даних з різними моделями подання. До таких моделей авторка відносить бази даних DB, сховища даних DW, статичні Web-сторінки Wb, НСД Nd, мультимедійні дані Gr, локальні сховища ODW, а також засоби інтеграції Int, пошуку Se та обробки Wo, що об’єднані середовищем управління моделями (EM).

Ці моделі даних ієрархічно впорядковані відповідно до їх виразної потужності: реляційна, багатовимірна, об’єктно-реляційна моделі, розширена мова розмітки інформації (Extensible Markup Language – XML) зі схемою, середовище опису ресурсів (Resource Description Framework – RDF), стандартний засіб опису зв’язків між об’єктами даних – онтології, описані за допомогою Web Ontology Language – OWL, структурований текст, неструктурований текст (рис. 3). Кожен учасник простору даних підтримує деяку модель даних і деяку мову запитів, відповідну до цієї моделі.

Документи та Web-сторінки можуть розглядатися в такому випадку як НСД. Процес розміщення таких інформаційних джерел у певній таксономії пов’язаний з їх класифікацією. Застосування стандартів W3C та онтологічного аналізу може застосовуватися для додавання структури до НСД.

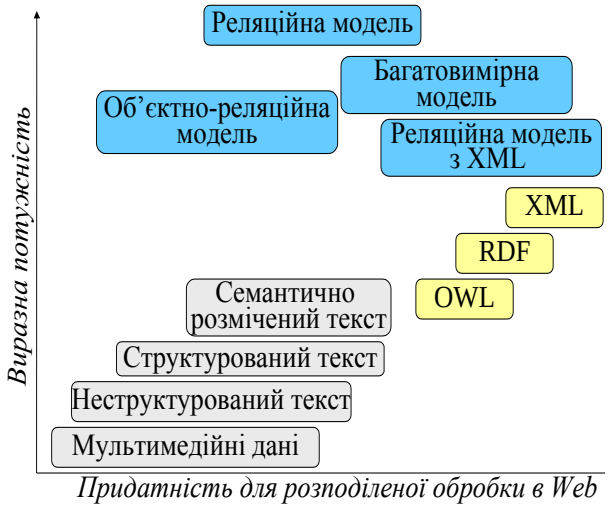

Рис. 3. Моделі подання даних

Найбільш розповсюдженою моделлю збереження структурованих даних з кінця 70-х років 20 ст. є реляційна модель, а стандартом на їхню обробку – мова SQL. Однак для НСД ця модель не ефективна.

Нереляційні моделі даних. Сьогодні задачі, що виходять за рамки реляційної моделі, прийнято відносити до класу NoSQL (звичайно розшифровується як Not Only SQL), кожен підклас якого вирішує окрему проблему, що погано реалізується за допомогою SQL, – наприклад, документо-орієнтовані, об'єктні та графові БД. Такі БД мають певні обмеження на операції, що підтримуються традиційними БД. Наприклад, великі розподілені БД повністю відмовляються від транзакцій, що забезпечує підвищення продуктивності за рахунок використання паралелізму.

Реалізована в проекті Hadoop технологія роботи з даними докорінно відрізняється від традиційних реляційних СКБД, призначених для роботи зі структурованими даними. NoSQL – сімейство технологій роботи з даними, які відрізняються від традиційних реляційних СКБД за наступними ознаками [9]: відсутність підтримки мови структурованих запитів

SQL; робота з неструктурованими чи слабко структурованими даними; відсутність механізмів забезпечення цілісності даних у тому вигляді, як вони реалізовані в класичних СКБД; розподілена реалізація з широкими можливостями горизонтального масштабування. У цілому основне призначення NoSQL полягає у можливості обробки великої кількості неструктурованих даних за нерегламентований час, але з гарантованим результатом. У цьому складається принципова відмінність NoSQL від традиційних СКБД, які забезпечують збереження інформації в чітко структурованому вигляді і гарантують час виконання операцій.

RDF як модель даних. Великий клас задач, які важко розв'язувати на реляційній моделі, – це задачі на сильно зв'язаних даних (графові задачі). Для них сьогодні найбільше поширення одержали RDF-сховища, які використовують стандарти W3C для мови RDF (Resource Description Framework) і запити SPARQL [10].

Основа RDF – це представлення даних у вигляді тверджень-трійок "суб'єкт-предикат-об'єкт". Для ідентифікації суб'єктів, об'єктів і предикатів використовується ідентифікатор URI (Uniform Resource Identifier), що є узагальненням поняття URL. Крім того, для подання об'єктів можуть використовуватися літерали.

На відміну від реляційної моделі, модель RDF досить гнучка – кожен суб'єкт може містити свої власні предикати й об'єкти, наприклад, у єдиній базі товарів усі товари мають предикат «Ціна», але в той же час холодильники можуть мати предикат «Обсяг морозильної камери», а телевізори – предикат «Діагональ екрана».

Модель RDF описує орієнтований граф, у якому кожна трійка – це опис зв'язку між двома вузлами.

Модель RDF служить для опису даних, але не описує методів їхньої обробки. Існує багато мов запитів до RDF: DQL, N3QL, R-DEVICE, RDFQ, RDQ, RDQL, SeRQL і т. д., але найпоширенішою є SPARQL – стандарт W3C, який, на відміну від SQL з неоднозначною граматикою і семантикою, має чітку структуру і більшу виразність. Основна частина запиту на SPARQL – шаблон, що описує підграф, який потрібно знайти в графі RDF. Цей шаблон представляється у вигляді набору трійок з перемінними. На сьогоднішній день SPARQL є однією з найбільш виразних мов обробки даних. Крім мови запитів, стандарт SPARQL регламентує протокол взаємодії з базою даних і формат результату, що є великим кроком вперед у порівнянні з SQL.

Рівень стандартизації RDF і SPARQL набагато вище, ніж у SQL, – зусиллями комітету W3C визначені стандарти не тільки на модель RDF і мову SPARQL, але і на ідентифікацію ресурсів (URI), протокол взаємодії компонентів (HTTP), точку доступу SPARQL тощо. Завдяки стандартизації дані з будь-якого RDF-сховища можна завантажувати в RDF-сховища різних виробників. Запити на SPARQL однаково виконуються на різних сховищах. У RDF легко зберігати метадані. На основі метаданих можна робити складні запити, вибираючи, скажемо, дані з конкретних джерел, у конкретному часовому діапазоні тощо.

Сьогодні спостерігається бурхливий розвиток ринку засобів розробки на основі моделі RDF. Деякі з них мають спеціалізовану архітектуру для обробки графів, а інші побудовані поверх реляційних БД. Найбільш поширені з них.

- *Apache Jena* – Java API для розробки застосунків Semantic Web, що містить кілька сховищ даних: Jena TDB – сховище RDF-трійок, Jena SDB – інтерфейс до реляційного сховища, In-Memory – сховище в пам'яті.
- *Ontotext OWLIM* – сімейство семантичних RDF-репозиторіїв з власним ядром, реалізованим на Java, з підтримкою семантики на RDFS (RDF Scheme) і OWL.
- *OpenLink Software Virtuoso* з власним RDF-сховищем, повною реалізацією SPARQL та можливістю читання RDF з файлів формату XML і Turtle.

Великі корпорації, такі як IBM і Oracle, також розробляють власні RDF-рішення. IBM пропонує NoSQL Graph Support, з інтерфейсом на основі розширення API Jena. Oracle у Spatial and Graph

Option підключила RDF до засобу обробки просторових даних Spatial Data Option.

RDF-сховища дозволяють збирати, зберігати й індексувати дані з різних джерел – зокрема, при рішенні актуальної задачі інтеграції сервісів, що зводиться до об'єднання розрізнених реляційних БД у єдину базу і приводить до задачі обробки квазіструктурованих даних. Дані усередині кожної з таких БД строго структуровані для роботи з реляційною моделлю, але кожна база структурована по-своєму, тому задача їхньої інтеграції в рамках реляційної моделі потребує переробки всього рішення. Якщо ж конвертувати такі бази в модель RDF, то інтеграція зводиться до простого злиття RDF-графів і переписуванню запитів з SQL у SPARQL.

RDF-сховища найбільш придатні для задач, що потребують виявлення та аналізу великої кількості взаємозв'язків. До таких задач відносяться:

- обробка семантичних мереж (і інших графових структур), отриманих в результаті аналізу природномовних текстів;
- представлення й обробка даних з соціальних мереж (побудова портрета користувача, виявлення центрів поширення інформації у соціальних мережах тощо);
- обробка даних складних наукових експериментів.

Практично всі задачі, у яких кількість взаємозв'язків між сутностями перевищує кількість сутностей чи орієнтованих на аналіз взаємозв'язків, можуть розглядатися як кандидати на рішення засобами систем RDF.

Сучасні програмні засоби обробки неструктурованих даних

Існує велика кількість програмних засобів для обробки та керування НСД. Деякі з них використовують системи *керування корпоративним контентом* (CMS), що можуть підтримувати весь життєвий цикл його контента (Web-контент, документи тощо). Багато постачальників CMS масштабують свої рішення для обробки Big Data та орієнтовані на опрацювання великих обсягів НСД у реальному часі, використовуючи такі технології, як Hadoop, MapReduce і потокова передача.

Методи роботи з НСД іноді протиставляють технологіям BI, однак точніше говорити про їх взаємне доповнення [11]. Основний недолік BI пов’язують з їхньою недостатньою динамічністю та непристосованістю для обробки Big Data у режимі реального часу. Крім того, традиційні методи BI орієнтувалися на аналіз структурованої інформації. Інтеграція BI з технологіями обробки НСД називають *Embraced Enterprise Search and Retrieval* (ESR): в них реалізовано дві всеохоплюючі (Embraced) функції – корпоративний пошук (Enterprise Search) і здобування інформації з даних (Retrieval). ESR, крім доступу до нових типів даних, дозволяють здобувати більше корисної інформації також і зі звичайних структурованих даних.

Проблеми аналізу НСД загострилися через нові джерела таких даних – соціальні мережі, мобільні пристрої, реєстратори. Використання інформаційно-пошукових систем (ІПС), що традиційно застосовуються для пошуку в Web, ускладнюється великими обсягами та великою швидкістю накопичення НСД, що характерні для Big Data. Водночас як застосування для цього технологій корпоративного пошуку виявилося надто коштовним.

Середня довжина запитів до ІПС не перевищує двох-трьох слів, користувачі рідко застосовують логічні операції. У традиційних ІПС кожен запит виконується незалежно від попередніх, і пошукові машини дають ту саму відповідь будь-якому користувачу поза залежністю від передісторії його роботи з базою. Деякі компанії (наприклад, Google) використовують додаткову контекстну інформацію (метадані), що відноситься до предмета пошуку, та рейтинги сторінок. Але й такі системи не враховують особливості корпоративних даних, структурувати які все ж більш легко, ніж інформацію від довільних користувачів.

Задачі, які вирішують системи CMS, – оцінка причин відтоку клієнтів шляхом побудови профілів клієнтів, ана-

ліз відгуків та їх емоційного забарвлення, оцінка компаній у ЗМІ, внутрішні розслідування (пошук та захист від видалення документів, пов'язаних з певним інцидентом, в якому аналізуються НСД з різних корпоративних джерел – поштових серверів, корпоративних порталів, телефонних і відеоконференцій, та побудова взаємозв'язків між ними).

Засоби, що використовуються в CMS для аналізу НСД, порівнюють за наступними параметрами [12], значення яких наведено у табл. 1.

Таблиця 1. Параметри порівняння засобів CMS

Параметр	*Можливі значення*
Тип засобу	Засоби Text Mining Обробка контенту баз даних Інтеграція Text Mining та обробки контенту баз даних
Можливості	Аналіз ключових слів Статистичний аналіз Лінгвістичний аналіз
Джерела даних	Структуровані бібліографічні джерела даних Неструктуровані джерела даних Гібридні джерела даних
Результати	Списки документів Таблиці Графіки Карти

Для того, щоб визначити типові операції аналізу НСД в CMS, розглянемо кілька прикладів програмних продуктів, що широко застосовуються для такого аналізу.

Autonomy *IDOL* (Intelligent Data Operating Layer) [13] базується на обробці змісту (Meaning-Based Computing) текстів незалежно від форми їхнього представлення і форматів та забезпечує пошук понять (концептів) за пов'язаними з ними словами ПМ [14]. Для цього використовують різні підходи – пошук за ключовими словами, що враховує найпростіші закономірності (частоту повторень слів тощо), ранжирування (PageRank) на основі частоти звертань до того чи іншого документа, федеративний пошук (Federated Search), та концептуальний пошук (Conceptual Search) та мультимедійний пошук (Audio and Video Search), що сполучує власне пошук з розпізнаванням образів. Обробка змісту даних починається з їх класифікації та кластеризації. IDOL для розуміння змісту даних використовує метод байєсівського виведення (розрахунок імовірності події на основі статистики її здійснення в минулому) і теорію інформації Клода Шеннона разом із традиційними підходами до аналізу. Це дозволяє визначити категорії документів за допомогою статистичного аналізу слів, що зустрічаються в цих документах.

Endeca Latitude [15] – технологія Text Mining, що призначається для аналізу потоків сирої текстової інформації з різних джерел та фокусується на розкритті змісту даних на противагу традиційному аналізу. Вона містить *Latitude Information Integration Suite* – набір засобів для збору і попередньої обробки потоку сирих вхідних даних (структурованих, неструктурованих і квазіструктурованих), а також середовище для створення аналітичних застосувань *Latitude Studio* та гібридну пошуково-аналітичну СКБД з високою масштабованістю *MDEX Engine*.

Ця платформа забезпечує здобуття наступних п'яти типів інформаційних шаблонів, за допомогою яких користувачі задають режими створення цільових моделей пошуку інформації [16]:

1) оптимізація, що керується порівнянням (Analyze-Compare- Evaluate);

2) оптимізація, орієнтована на дослідження (Explore-Analyze-Evaluate);

3) стратегічний аналіз (Analyze-Comprehend-Evaluate);

4) стратегічний нагляд (Monitor-Analyze-Evaluate);

5) синтез, керований порівнянням (Analyze-Compare- Synthesize).

На вході Endeca Latitude працює *Latitude Information Integration Suite*, що складається з трьох основних компонентів:

- *Latitude Content Acquisition System* – система збору контента, що містить колекцію конекторів для виділення, очищення й інтеграції НСД з файлових систем, Web-сайтів тощо;
- *Latitude Data Integrator* – інтегратор, що виконує функції, аналогічні ETL (Extract, Transform, and Load – Витяг, Перетворення та Завантаження) у сховищах даних;
- Open *Interfaces and Connectors* – інтерфейси і конектори для отримання даних з Apache Hadoop та інших джерел.

MDEX Engine націлена на пошук і виявлення знань і є гібридом ІПС та аналітичної СКБД, що призначена для обробки даних, що швидко змінюються.

Принципова відмінність MDEX від традиційних СКБД полягає у наближенні записів, що зберігаються в ній, до реальностей навколишнього світу. Ці записи містять пари атрибутів "ключ/значення" (key/value). У формі атрибутів зберігаються ієрархічно організовані дані, наприклад елементи ієрархій XML, причому так, що користувач має можливість буквально угвинчуватися (drill-into) у набори даних, використовуючи для цього інструменти Latitude Studio. Таким чином MDEX дозволяє максимально позбутися процесів моделювання та працювати з даними у тому вигляді, як вони надійшли і зберігаються, – те, що називають «завантажив і пішов».

У MDEX реалізований фасетний пошук – пошук в інформаційних середовищах, побудованих за принципами *фасетної класифікації*.

Фасетна класифікація (класифікація двокрапкою, класифікація Ранганатана) – це сукупність кількох незалежних класифікацій, що здійснюються одночасно за різними базисами. В такій класифікації поняття представлені у вигляді перетину ряду ознак, а класифікаційні індекси синтезуються за допомогою комбінування фасетних ознак відповідно до фасетної формули [17].

Ця класифікація запропонована Шиалі Ранганатаном, відомого створенням "П'яти законів бібліотечної науки" (1931) [18], як варіант бібліотечно-бібліографічного підходу до багатоаспектної класифікації для звичайних паперових бібліотек і пізніше поширилися для комп'ютерних застосувань.

Це неієрархічна система організації інформації, у якій прості поняття розподілені у фасети – групи однорідних понять, пов'язані узагальненням за однією певною ознакою. Її структура є прямим відображенням системної характеристики класифікації, тобто базується на поділі об'єктів за кількома класифікаційними ознаками одночасно [19]. Фасетною ознакою може бути будь-яка класифікаційна ознака, яка використовується для угрупування понять у фасетні ряди, у результаті чого утворюються підкласи.

Особливість фасетної класифікації пов'язана з представленням фасетних ознак через їх послідовність, тобто результат класифікації залежить від впорядкування фасет (це визначає їх важливість для класифікації). Більш того, послідовність ознак у цій класифікації впливає на зміст поняття (наприклад, "процес: матеріал: устаткування: властивість"), яке визначає фасетна формула – індекс, що складається з послідовності фасетних ознак, розділених двокрапкою.

Такий підхід забезпечує багатоаспектний пошук інформації. У цій класифікації сполучаються індекси з різних таблиць у певних комбінаціях, що дозволяє отримати індекси для різноманітних предметів. Основна таблиця фасетної класифікації в кожній *предметній області* (ПрО) представлена набором таблиць, що будуються за класифікаційними ознаками (категоріями, фасетами) різного ступеня узагальнення – загальні (наприклад, "Властивості"), спільні для великих груп ПрО (наприклад, "Обладнання") та специфічні для окремих ПрО (наприклад, "Алгоритми сортування даних"). Таблиці таких категорій розробляються відповідно до специфіки кожної ПрО, а типові ознаки, характерні для більшості або всіх відділів фасетної класифікації, відображаються у

додаткових таблицях, а в особливій таблиці міститься визначення характеру зв'язків між поняттями ("Вплив", "Порівняння" тощо).

ClearForest (http://www.clearforest.com/Technology/) пропонує рішення Text Analytics, що містить платформу видобування тексту, аналітичну платформу та середовища розробки. Інструмент видобування тексту виконує генерування матриць спільного застосування термінів, кластеризації даних, видобутку термінів і тегування, тобто обирає відповідні терміни з неструктурованого тексту, наприклад, статей новин, Web-опитувань і документів HTML. Після структуризації ця інформація може бути використана в автономних аналітичних застосунках або в поєднанні зі структурованими даними, щоб забезпечити більш комплексний бізнес-інтелект. Терміни витягуються для подальшого аналізу і автоматично класифікуються в попередньо визначені категорії або таксономії.

Інструмент дозволяє візуалізувати взаємозв'язки між колекціями таксономій, щоб отримати інформацію, яка є актуальною, дієвою та додає цінності іншим інструментам Business Intelligence.

Однією з переваг ClearForest є перетворення НСД у структуровані дані за допомогою модуля Packaged Extraction Module. Наприклад, текст патентних документів перетворюється у структуровані таблиці з такими параметрами, як "проблеми" та "технологічні процеси".

Inxight (http://www.inxight.com/products/smartdiscovery_as/) – набір програмних рішень для аналізу ПМ дослідницького центру Xerox Palo Alto (PARC), що дозволяють розуміти документи настільки глибоко, щоб забезпечити їх індексацію, класифікацію та витяг всіх необхідних понять, сутностей та відношень. Програмне забезпечення ідентифікує більше 35 типів інформації в одному документі. Джерелами даних є текстові НСД, наприклад, новини, Web-сайти, внутрішні документи та повнотекстові патенти. Метадані і об'єкти можуть бути здобуті з попередньо оброблених документів. Результатом роботи Inxight є ієрархічна категоризація документів, тобто документи аналізуються на основі заздалегідь визначених категорій в ієрархіях.

Важливою особливістю Inxight є можливість одночасного пошуку в декількох онлайнових БД, відома як федеративний пошук ("fedederated search"). Inxight працює з 32 мовами і ідентифікує 27 типів об'єктів. Розробника Inxight стверджують, що лінгвістичні алгоритми, використані в цьому продукті, є найпотужнішими в даній галузі. Недоліком системи є потреба у значних витратах часу на аналіз тексту.

Платформа *Velocity Platform (http://vivisimo.com/html/velocity*) компанії Vivisimo складається з трьох пов'язаних програмних продуктів:

- *Search Engine* – багатофункціональна пошукова машина, агенти-краулери якої здатні аналізувати файли різних типів (HTML, TXT, RTF, Adobe Acrobat PDF, PostScript, MS Word, Excel, PowerPoint, WordPerfect, ZIP, GZIP, TAR Lotus Notes) та здобувати інформацію з реляційних СКБД різними мовами (всі європейські мови, арабська і китайська);
- *Clustering Mashine* – засіб кластеризації, що групує результати, отримані від Google, Autonomy, FAST і Ultraseek, а також тексти в різних форматах;
- *Content Integrator* – інтегратор, що забезпечує федерований пошук, що вміє працювати з метаданими і передавати результати до Clustering Engine.

Інші відомі програмні продукти, орієнтовані на аналіз текстових НСД, – це Goldfire Innovator (http://www.invention-machine.com/GoldfireInnovator.htm), OmniViz (http://www.biowisdom.com/solutions/), TEMIS (http://www.temis.com).

Сфера застосування засобів аналізу НСД

Призначення багатьох комерційних систем, що здійснюють аналіз текстових НСД, пов'язане з підтримкою зворотного зв'язку з клієнтами та аналізом емоційного інформаційного фону, що складається на-

вколо організації і її конкурентів [20]. Джерелами даних для них є ЗМІ, портали новин, соціальні мережі, аналітичні портали, внутрішні інформаційні ресурси компаній тощо. У цілому робота з НСД – це пошук і агрегація контента з різних джерел, витяг даних відповідно до заданих параметрів і їхній семантичний аналіз, а також надання підсумкових відомостей користувачу в зручному вигляді.

Наведемо ще кілька характерних прикладів таких систем:

- *First Rain* компанії First Rain – рішення для пошуку, збору й аналізу інформації тільки з Web-ресурсів (звітів компаній та аналітичних оглядів), яке класифікує знайдені відомості за стандартизованих набором тем і значущістю для клієнта;
- *Digimind* – рішення для пошуку структурованих і неструктурованих даних, з Web і соціальних мереж, що забезпечує класифікацію оброблених матеріалів та представлення підсумкових даних у вигляді, зручному для користувача;
- *InfoNgen* – набір рішень для пошуку, збору й аналізу НСД, що агрегують відомості з різних Web-джерел, електронної пошти та внутрішніх інформаційних ресурсів організації та категоризують їх відповідно до таксономії клієнта та дозволяють враховувати специфічні особливості кожного джерела;
- *Factiva* – набір інформаційно-аналітичних рішень, що дозволяє збирати мультимедійний контент з сайтів новин;
- *«Голос клієнта»* – рішення для аналізу структурованих і неструктурованих даних для обробки відгуків клієнтів з соціальних мереж, центрів роботи з клієнтами і CRM, форумів і блогів.

Семантичний аналіз НСД дозволяє визначити заголовок, резюме, зміст, дату публікації тощо, заданих користувачем елементів (наприклад, назв компаній, найменувань продуктів, послуг), відкинути непотрібні дані (рекламні оголошення, правові обмеження), розпізнати семантичну структуру тексту та семантичні залежності. У ході морфологічного і лексичного аналізу кожен текст розділяється на зв'язані між собою слова, що зіставляються з заздалегідь визначеними тегами. В процесі аналізу враховуються синоніми, можливі варіанти написання слів (іншими мовами або з типовими помилками), абревіатури.

Крім того, існує можливість визначення емоційної тональності тексту, що дозволяє оцінити відношення авторів документа до окремих інформаційних об’єктів, – позитивне чи негативне, а також задати цінність кожного позитивного і негативного висловлення в залежності від цілей користувача.

Постановка задачі

У зв’язку з тим, що велика частка інформаційних ресурсів – це неструктуровані текстові дані, виникає потреба у створенні засобів, що забезпечують здобуття з цих НСД тієї інформації, що необхідна користувачам для розв’язку їх поточних проблем. Використання традиційних засобів Text Mining може бути недостатньо ефективним для обробки Big Data, і це викликає необхідність інтелектуалізації засобів аналізу НСД. Основою такого аналізу може стати застосування фонових знань щодо предметної області, формалізованих за допомогою онтологій. Це викликає потребу у методах побудови спеціалізованих онтологій для задач користувачів та їх застосування для семантичної розмітки природномовних текстів. Пропонується використовувати для цього технології Wiki та їх семантичне розширення, а також створювати семантично розмічені Wiki-ресурси як основу для структурування довільних природномовних текстів.

Технологія Wiki як засіб структурування інформації

Під *Wiki-технологією* зазвичай розуміють таку технологію побудови Web-ресурсу, яка дає змогу відвідувачам брати участь у редагуванні його вмісту – виправляти помилки, додавати нові матеріали, не використовуючи спеціальні програми, явно вказувати зв’язки між окремими сторінками за допомогою гіперпосилань та

визначати категорії, до яких вони відносяться [21].

Формат Wiki-сторінок – це спрощена мова розмітки, що використовується для того, щоб виділити в тексті різні структурні й візуальні елементи або вказати на них. Важливою особливістю Wiki є те, що вносити структурування до текстових НСД за допомогою Wiki-розмітки може практично кожен користувач. На сьогодні існує велика кількість Wiki-двигунів та створених на їх основі розподілених інформаційних ресурсів різного обсягу та спрямованості. Найбільш великим та відомим з них є Вікіпедія.

Основними елементами Wiki-розмітки є гіперпосилання та категорії. Їх застосування дозволяє досить легко перетворювати НСД у частково структуровані дані. Крім того, аналіз структурування Wiki-ресурсів на рівні слів та понять дозволяє отримувати знання для структурування інших НСД.

Семантизація Wiki-ресурсів

Semantic MediaWiki (SMW) – це надбудова над інструментальним засобом побудови Wiki-сайту MediaWiki [22]. Переваги SMW – це обробка інформації на семантичному рівні, наявність засобів групового керування знаннями, відносно висока виразна потужність, надійна реалізація і зручний інтерфейс користувачів, наявність документації та спільнот користувачів [23]. Це дозволяє інтегрувати інформацію з різних Wiki-сторінок, здійснюючи пошук на рівні знань, та генерувати за Wiki-сторінками онтологічні структури, які можуть використовувати інші ІІС.

Крім категорій, в SMW для структурування інформації використовуються такі механізми, як *семантичні властивості*. Вони дозволяють семантично пов’язувати Wiki-сторінки як між собою, так і з різними даними. Кожна семантична властивість має тип, назву і значення, а також власну Wiki-сторінку в спеціальному просторі імен, яка дозволяє визначати її місце в ієрархії властивостей та документувати те, як цю властивість необхідно використовувати.

З точки зору онтологічного аналізу, кожна Wiki-сторінка являє собою онтологічний елемент, тобто елемент одного з RDF-класів – Thing, Class, ObjectProperty, DatatypeProperty, AnnotationProperty. Крім того, кожна стаття має власний URI, який дозволяє уникнути плутанини між поняттями і HTML- сторінками. Зазвичай, статті є екземплярами класів онтології OWL, категорії – класами, а відношення – об'єктними властивостями онтології.

Виходячи з цього, для будь-якої сторінки SMW за запитом може генерувати відповідний OWL/RDF-файл. Найпростіший спосіб отримати цей RDF – просто використати посилання "Переглянути як RDF" ("View as RDF"), що знаходиться в нижній частині кожної анотованої сторінки. Ця сторінка може виступати як кінцева точка (endpoint) для зовнішніх сервісів (зовнішньої точки доступу), які хочуть отримати доступ до семантичних даних SMW. На жаль, ця функція реалізована дуже невдало та підтримує надто мало опцій.

Оскільки SMW сумісна з моделлю знань OWL DL, то існує можливість використання в Wiki-ресурсах існуючих онтологій. Це можливо здійснити двома шляхами: імпорт онтології дозволяє створювати і модифікувати сторінки у Wiki для подання відношень, заданих в деякому існуючому OWL DL-документі; а повторне використання словника дозволяє користувачам відображати (задавати відповідності) Wiki-сторінки на елементи існуючих онтологій. Функція імпорту онтології для читання RDF-документів витягує RDF-твердження, які можуть бути представлені у Wiki.

Семантичні Wiki-ресурси можуть використовуватися як основа для автоматизованої генерації розподілених баз знань в форматі RDF. Експорт в OWL/RDF є засобом забезпечення зовнішнього повторного використання даних з Вікі, але тільки практичне застосування цієї функції може показати якість згенерованого RDF. З цією метою для видачі RDF, розробники системи використовували ряд інструментів Semantic Web.

Таким чином, наявність перевіреного та семантично розміченого Wiki-ресурсу дозволяє побудувати онтологію тієї ПрО, що цікавить користувача, яка може використовуватися в Text Mining для структурування НСД з цієї ПрО. Переваги використання моделі даних RDF вище наведені.

Основна проблема отримання фонових знань для Text Mining з Wiki-ресурсів пов'язана з тим, що сьогодні:

- преважна частка Wiki-ресурсів не семантизована;
- Wiki-ресурси здебільшого не є реферованими та авторськими, і тому наявність в них помилок (фактичних, структурних та змістовних) досить ймовірна;
- семантизовані Wiki-ресурси з високим рівнем довіри до контенту здебільшого високо спеціалізовані та орієнтовані на подання знань відносно вузьких ПрО (крім того, навіть в таких ресурсах зазвичай Wiki-онтологія, що лежить в основі їх семантичної розмітки, зазвичай не є доступною для зовнішніх користувачів);
- пошук та аналіз RDF та OWL із зовнішніх сховищ та репозиторіїв – досить складна задача, незважаючи на наявність спеціалізованих пошукових запитів, а знайдені таким чином онтології можуть не повністю відповідати поточним потребам користувача.

Крім вбудованих в Semantic MediaWiki засобів генерації RDF, існує багато більш спеціалізованих алгоритмів побудови онтологій на основі семантичної Wiki-розмітки [24]. Ці алгоритми дозволяють використовувати семантичний пошук та фонові знання щодо специфіки ПрО для формування корпусу Wiki-текстів, за якими створюється онтологія.

Це викликає потребу у розробці та вдосконаленні енциклопедичних онлайнових видань на базі семантичних Wiki-ресурсів. Саме до таких продуктів відноситься портальна версія Великої української енциклопедії е-ВУЕ. (http://vue.gov.ua), яка використовує вільне програмне забезпечення MediaWiki версії 1.29.1. та його семантичне розширення Semantic MediaWiki версії 2.5.5. (рис. 4). Це інноваційний проект із створення національної енциклопедії на основі сучасних засобів подання знань.

Рис. 4. Головна сторінка е-ВУЕ

Принциповими відмінностями е-ВУЕ від інших онлайнових довідників та енциклопедій (наприклад, від Вікіпедії) є:

- рецензованість – е-ВУЕ є науковим виданням з високою репутацією, яке подає експертні знання у вигляді авторських статей та надає перевірені сталі факти;
- наявність обробки інформації на рівні знань – пошук за семантичними властивостями;
- використання авторських статей.

Щоб використовувати семантизований Wiki-ресурс як розподілену базу знань (БЗ), створено Wiki-онтологію – модель знань цього ресурсу. Використання цієї моделі для семантичної розмітки забезпечує формування та програмної реалізації відповідного набору ієрархічно пов'язаних категорій, шаблонів типових інформаційних об'єктів, їх семантичних властивостей та запитів, що їх використовують. Наявність формальної моделі дозволить запобігти неоднозначної інтерпретації знань різними розробниками та користувачами ресурсу.

До пошуку подібних статей доцільно застосовувати принципи фасетної класифікації, тому що в е-ВУЕ для категоризації статей використовуються одночасно різні незалежні таксономії, такі як:

- галузі знань та їх підгалузі (рис. 5);

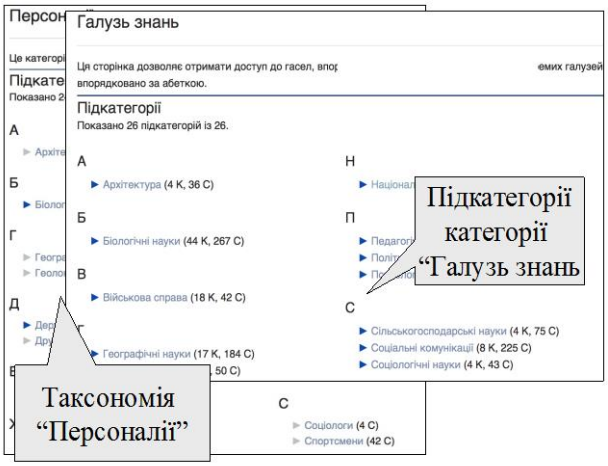

Рис. 5. Категорії e-BYE

- типові інформаційні об’єкти;
- наявність різних типів мультимедійного супроводу;
- таксономія географічних об’єктів;
- природа, цивілізація та персоналії;
- класифікація за авторами та модераторами.

Кожна стаття може використовувати один або кілька шаблонів для типових інформаційних об’єктів, що дозволяють задати значення семантичних властивостей сторінки та змістовно визначити її відношення з іншими сторінками енциклопедії.

Таким чином, *подібні* статті, – це статті, що віднесені до однакового або подібного набору категорії та мають подібні семантичні властивості, тобто аналіз близькості статей може оцінюватися через співставлення їх фасетних індексів.

Аналіз виразних можливостей розширених засобів подання та структурування інформації засобами технологічного середовища Semantic MediaWiki виявив, що, незважаючи на значно меншу, порівняно з онтологіями, їх виразну здатність, ці засоби дозволяють не тільки представляти класи та екземпляри онтологій, для яких існують однозначно визначені аналоги у Wiki [25] – категорії та Wiki-сторінки, але й представляти деякі більш складні знання. Запропоновано в цій роботі онтологічна модель Wiki-ресурсу дозволяє формально описувати такі характеристики семантичних властивостей різних типів, як припустимість неповних та множинних значень. Використання класів та екземплярів цієї онтології дозволяє генерувати Wiki-сторінки, на яких містяться результати виконання семантичних запитів, і за цими сторінками створювати онтології ПрО, що цікавлять користувачів. Застосування стандартів Semantic Web у Semantic MediaWiki забезпечує можливість використання цих онтологій у застосуваннях Text Mining без додаткової обробки.

Висновки

Проаналізувавши сучасні тенденції поширення неструктурованих текстових даних та засоби, що використовуються для їх аналізу, можна зробити висновки щодо високої актуальності цього напрямку та необхідність застосування до такої обробки інтелектуальних інформаційних систем. Big Data, значну частину яких складають саме неструктуровані тексти, потребують подальшого розвитку Text Mining та алгоритмів машинного навчання.

НСД, що складаються із природномовного тексту, у загальному випадку не мають попередньо визначеної моделі даних. Їх неоднозначність, гетерогенність та залежність від контексту значно ускладнюють класифікацію документів, ідентифікацію їх компонентів та автоматизоване здобуття з їх контенту знань, потрібних користувачеві, тоді як великі обсяги та динамічність таких даних не припускають ефективної ручної обробки.

Розглянуто засоби та методи структурування НСД, їх різноманітні програмні реалізації. Проаналізовано перспективи використання фонових знань для такого структурування. Обґрунтовано доцільність застосування для цього таких стандартів W3C, як RDF та OWL.

Використання семантичних Wiki-технологій для створення розподілених інформаційних ресурсів не тільки дозволяє досить легко додавати структурування до НСД, але й є джерелом фонових знань для аналізу довільних текстів відповідної предметної області. Запропоновані в роботі моделі та методи дозволяють вдосконалити цей процес.

Література

1. Grimes S. Unstructured Data and the 80 Percent Rule, 2008, Clarabridge, Bridgepoints. – http://breakthroughanalysis.com/2008/08/01 /unstructured-data-and-the-80-percent-rule/.
2. Неструктурированные данные в большой среде данных. – https://ru.howtodou.com/unstructured-data-in-big-data-environment.
3. Unstructured_data. – https://en.wikipedia.org/wiki/Unstructured_data.
4. Grimes S. A Brief History of Text Analytics. B Eye Network, 2016. – http://www.b-eye-network.com/view/6311.
5. Buneman P., Davidson S., Fernandez M., Suciu D. Adding structure to unstructured data. *International Conference on Database Theory*, 1997. P. 336–350.
6. Гладун А.Я., Рогушина Ю.В. Data Mining: пошук знань в даних. К.: ТОВ "ВД "АДЕФ-Україна", 2016. 452 с.
7. Feldman R., Sanger, J. The text mining handbook: advanced approaches in analyzing unstructured data. Cambridge university press, 2007. https://wtlab.um.ac.ir/images/e-library/text_mining/The%20Text%20Mining%20HandBook.pdf.
8. Шаховська Н. Особливості моделювання просторів даних, 2007. ena.lp.edu.ua/bitstream/ntb/35116/1/24_139-148.pdf.
9. Sadalage P., Fowler M. NoSQL Distilled. Pearson Education, 2012. 192 p.
10. Головков В., Портнов А., Чернов В. RDF – инструмент для неструктурированных данных. *Открытые системы. СУБД.* https://www.osp.ru/os/2012/09/13032513/.
11. Черняк Л. Аналитика неструктурированных данных. *Открытые системы*. 2012, № 06.
12. Yang Y., Akers L., Klose T., Yan, C. B. Text mining and visualization tools–impressions of emerging capabilities. World Patent Information. 2008. 30(4). P. 280–293. https://www.scss.tcd.ie/Khurshid.Ahmad/Research/High_Frequency_Trading/2008_Yanget al_TextMiningVis_ WorldPatent.pdf.
13. Autonomy IDOL. http://www.autonomy.com/content/Products/products-idol-server/index.en.html.
14. Lyte V., Jones S., Ananiadou S., Kerr L. UK institutional repository search: innovation and discovery, 2009. http://www.ariadne.ac.uk/issue/61/ lyte-et-al/.
15. Russell-Rose T., Lamantia J., Burrell M. A Taxonomy of Enterprise Search. *EuroHCIR*, 2011. P. 15–18. https://www.researchgate.net/profile/Joe_Lamantia/publication/235971352_A_Taxonomy_of_Enterprise_Search_and_Discovery/links/00b7d515063de775c8000000.pdf.
16. Lamantia J. 10 Information Retrieval Patterns, 2006. http://www.joelamantia.com/ information-architecture/10-information-retrieval-patterns.
17. Фасетна класифікація. http://uk.wikipedia.org/wiki/ Фасетна_класифікація.
18. Noruzi A. Application of Ranganathan's Laws to the Web. http://www.webology.org/2004/v1n2/a8.html.
19. Сербин О.О. Особенности фасетной классификации документов в условиях современной трансформации содержания науки о книге. Книжная культура в контексте международных контактов: Материалы III Международной научной конференции, Минск: ЦНБ НАН Беларуси; М.: ФГБУН НИЦ «Наука» РАН, 2015. С. 457–462. http://eprints.rclis.org/25289/1/serbin.pdf.
20. Оганесян А. Неструктурированные данные 2.0. *Открытые системы. СУБД*. 2012. № 04. https://www.osp.ru/os/2012/04/13015772/.
21. Wagner C. Wiki: A technology for conversational knowledge management and group collaboration. *The Communications of the Association for Information Systems*. 2004. Vol. 13(1). P. 264–289. http://aisel.aisnet.org/cgi/viewcontent.cgi? article= 3238& context= cais.
22. MediaWiki. https://www.mediawiki.org/wiki/MediaWiki.
23. Рогушина Ю.В., Прийма С.М., Строкань О.В. Створення та використання семантичних Wiki-ресурсів: навчальний довідник. Мелітополь, ФОП Однорог Т.В. 2017. 169 с.
24. Rogushina J. Processing of Wiki Resource Semantics on Base of Ontological Analysis. Proc.of VIII International scientific conference «Open Semantic Technologies for Intelligent Systems» OSTIS-2018, Minsk, 2018. P. 159–162. https://libeldoc.bsuir.by/bitstream/123456789/30389/1/Rogushina_Processing.PDF.
25. Rogushina J. Analysis of Automated Matching of the Semantic Wiki Resources with Elements of Domain Ontologies.

International Journal of Mathematical Sciences and Computing (*IJMSC*). 2017. Vol. 3. N 3. P. 50–58. http://www.mecs-press.org/ijmsc/ijmsc-v3-n3/IJMSC-V3-N3-5.pdf.

References

1. Grimes S. Unstructured Data and the 80 Percent Rule, 2008, Clarabridge, Bridgepoints. http://breakthroughanalysis.com/2008/08/01 /unstructured-data-and-the-80-percent-rule/.
2. Unstructured data in big data environment. https://ru.howtodou.com/unstructured-data-in-big-data-environment.
3. Unstructured_data. https://en.wikipedia.org/wiki/Unstructured_data.
4. Grimes S. A Brief History of Text Analytics. B Eye Network, 2016. http://www.b-eye-network.com/view/6311.
5. Buneman P., Davidson S., Fernandez M., Suciu D. Adding structure to unstructured data. // International Conference on Database Theory, 1997. P. 336–350.
6. Gladun A.Ya., Rogushina Y.V. Data Mining: Finding Knowledge in Data. K .: ADEF-Ukraine Ltd., 2016. 452 p. [in Ukrainian]
7. Feldman R., Sanger, J. The text mining handbook: advanced approaches in analyzing unstructured data. Cambridge university press, 2007. https://wtlab.um.ac.ir/images/e-library/text_mining/The%20Text%20Mining%20HandBook.pdf.
8. Shakhovska N. Features of modeling of data spaces, 2007. ena.lp.edu.ua/bitstream/ntb/35116/1/24_139-148.pdf. [in Ukrainian]
9. Sadalage P., Fowler M. NoSQL Distilled. Pearson Education, 2012. 192 p.
10. Golovkov V., Portnov A., Chernov V. RDF as a tool for unstructured data // Open Systems. https://www.osp.ru/os/2012/09/13032513/. [in Russian]
11. Chernyak L. Analytics of unstructured data. Open systems, 2012, № 06. [in Russian]
12. Yang Y., Akers L., Klose T., Yan, C. B. Text mining and visualization tools–impressions of emerging capabilities. World Patent Information, 30(4), 2008. P. 280–293. https://www.scss.tcd.ie/Khurshid.Ahmad/Research/High_Frequency_Trading/2008_Yangetal_TextMiningVis_ WorldPatent.pdf.
13. Autonomy IDOL. http://www.autonomy.com/content/Products/products-idol-server/index.en.html.
14. Lyte V., Jones S., Ananiadou S., Kerr L. UK institutional repository search: innovation and discovery, 2009. http://www.ariadne.ac.uk/issue/61/ lyte-et-al/.
15. Russell-Rose T., Lamantia J., Burrell M. A Taxonomy of Enterprise Search // EuroHCIR, 2011. P. 15–18. https://www.researchgate.net/profile/Joe_Lamantia/publication/235971352_A_Taxonomy_of_Enterprise_Search_and_Discovery/links/00b7d515063de775c8000000.pdf.
16. Lamantia J. 10 Information Retrieval Patterns, 2006. http://www.joelamantia.com/information-architecture/10-information-retrieval-patterns.
17. Faceted classification. http://uk.wikipedia.org/wiki/ Фасетна_класифікація. [in Russian]
18. Noruzi A. Application of Ranganathan's Laws to the Web. http://www.webology.org/2004/v1n2/a8.html.
19. Serbin O.O. Features of the faceted classification of documents under the modern transformation of the book science content // Book culture in the context of international contacts: Proc| of the III International Scientific Conference, Minsk: Central Scientific Library of the National Academy of Sciences of Belarus, 2015. P. 457–462. http://eprints.rclis.org/25289/1/serbin.pdf. faceted classification. [in Russian]
20. Oganesyan A. Unstructured Data 2.0 // Open Systems. N 04, 2012. https://www.osp.ru/os/2012/04/13015772/. [in Russian]
21. Wagner C. Wiki: A technology for conversational knowledge management and group collaboration // The Communications of the Association for Information Systems. 2004. Vol. 13(1). P. 264–289. http://aisel.aisnet.org/cgi/viewcontent.cgi? article= 3238& context= cais.
22. MediaWiki. https://www.mediawiki.org/wiki/MediaWiki.
23. Rogushina Y.V., Priyma S.M, Strokan O.V. Creating and use of the Semantic Wiki resources: tutorial. Melitopol, FOP Odnorog T.V., 2017. 169 p. [in Ukrainian]
24. Rogushina J. Processing of Wiki Resource Semantics on Base of Ontological Analysis. Proc.of VIII International scientific conference «Open Semantic Technologies for Intelligent Systems» OSTIS-2018, Minsk,

2018. P. 159–162. https://libeldoc.bsuir.by/bitstream/123456789/30389/1/Rogushina_Processing.PDF.
25. Rogushina J. Analysis of Automated Matching of the Semantic Wiki Resources with Elements of Domain Ontologies. International Journal of Mathematical Sciences and Computing (IJMSC). 2017. Vol. 3. N 3. P. 50–58. http://www.mecs-press.org/ijmsc/ijmsc-v3-n3/IJMSC-V3-N3-5.pdf.

Одержано 06.02.2019

Про автора:

Рогушина Юлія Віталіївна,
кандидат фізико-математичних наук,
старший науковий співробітник.
Кількість наукових публікацій в українських виданнях – 140.
Кількість наукових публікацій в зарубіжних виданнях – 30.
Індекс Хірша – 3.
http://orcid.org/0000-0001-7958-2557.

Місце роботи автора:

Інститут програмних систем
НАН України,
03181, Київ-187,
проспект Академіка Глушкова, 40.
Тел.: 066 550 1999.
E-mail: ladamandraka2010@gmail.com