УДК 004.94 https://doi.org/10.15407/pp2025.01.094

В.Д. Міненко

ВІЗУАЛІЗАЦІЯ СЕМАНТИК РІЗНОФОРМАТНИХ ТЕКСТОВИХ ОПИСІВ

Дане дослідження спрямоване на вирішення задачі виявлення семантичного змісту з довільних текстових описів, представлених у різних форматах, з метою подальшої його візуалізації за допомогою сучасних засобів генеративного штучного інтелекту.

Стрімкий розвиток технологій штучного інтелекту надає принципово нові можливості для вирішення як задач аналізу тексту, так і генерації контентів – візуалізацій (у вигляді зображень чи відео). Унаслідок чого можна говорити про інший, сучасний рівень вирішення прикладних задач, що використовують подібну функціональність. Галузь генеративного штучного інтелекту доволі молода і містить чимало не вирішених проблем. Згенерована візуалізація характеризується не лише технічною якістю зображення чи відео, а й адекватністю відображення семантики вхідного текстового опису, яка зазвичай напряму залежить не тільки від можливості обраного застосунку ШІ- генерації, а й від структури та змісту вхідної текстової підказки.

Дана стаття описує алгоритм формування ланцюжка вирішення поставленої задачі від критеріїв вибору засобів реалізації та виокремлення проблем, що потребують вдосконалення та доробки, до визначення схеми композитного рішення. Метод, створенний в рамках запропонованого дослідження, має певні обмеження, а саме: він не підтримує мультимовний контент та не охоплює обробку діалектів, жаргонів, автоматичне визначення мови тексту.

Ключові слова: візуалізація семантики, семантично значущі елементи, генеративний штучний інтелект, обробка тексту природної мови, методи аналізу тексту, токенізація, лемінг, сегментація, модель ШІ-генерації, генеративна змагальна мережа, машинне навчання, моделі з відкритим кодом, метрики оцінювання якості візуалізації, текстова підказка, задача кластерізації, кластерний аналіз.

V. D. Minenko

VISUALIZATION OF THE SEMANTICS OF TEXT DESCRIPTIONS PRESENTED IN VARIOUS FORMATS

This study is aimed at solving the problem of identifying semantics from arbitrary texts presented in various formats and further visualizing it using modern tools of generative artificial intelligence.

The rapid development of artificial intelligence technologies provides fundamentally new opportunities for solving both text analysis tasks and content generation - visualizations (in the form of images or videos). As a result, we can talk about a different, modern level of solving applied problems using similar functionality. The field of generative artificial intelligence is still quite young and contains many unsolved problems. The generated visualization is characterized not only by the technical quality of the image or video, but also by the adequacy of the presentation of the semantics of the input text description, which usually directly depends not only on the possibility of the selected AI tool, but also on the structure and content of the input text prompt.

This article describes the algorithm to form a chain of solving the given task, from the criteria for choosing tools of developments and identifying problems that need improvement or resolving, to determining the scheme of a composite solution. The method created within the framework of the proposed study has certain limitations, namely: it does not support multilingual content and does not cover the processing of dialects, slangs, automatic detection of the language of the text.

Key words: visualization of semantics, semantically meaningful elements, generative artificial intelligence, natural language processing, text analysis methods, tokenization, lemming, segmentation, AI generation model, generative adversarial network, machine learning, open source models, visualization quality evaluation metrics, text prompt, clustering problem, cluster analysis.

Вступ

Стрімке зростання обсягів та різноманітності великих даних і розвиток технологій штучного інтелекту з одного боку висуває нові вимоги до обробки інформації, а з іншого - відкриває принципово інші можливості вирішення задач, що оперують цією інформацією. Метою даного дослідження є визначення методів виявлення семантики неструктурованої текстової інформації та формування алгоритму її візуалізуалізації з максимальним ступенем достовірності.

На сьогодні не існує готового сервісу чи системи, що реалізує поставлену задачу в цілому, охоплюючи весь ланцюжок від аналізу різноформатного (та такого, що походить з різних джерел) тексту природної мови до автоматичної візуалізації виявлених семантичних об'єктів візуалізації. Але існує чимала кількість розробок (методів, бібліотек, моделей), що реалізують окремі функції, а також ті, які можуть і мають бути використані як складові в побудові композитного рішення з певним удосконаленням, розширенням, розвитком і вирішенням інтеграційних питань.

Слід зазначити, що критерії вибору та вага семантичних елементів визначаються, перш за все, предметною областю та цілями класу прикладних задач, на які орієнтований алгоритм.

Якщо результат візуалізації є не лише швидким та яскравим, а передусім інформативним та достовірним, тобто таким, що адекватно відображає саме семантичний вміст вхідного тексту, то реалізація подібного алгоритму та створення методології його побудови уможливлює вирішення цілої низки суттєвих задач на принципово новому інтелектуальному рівні, як-от наприклад:

- візуальний аналіз інформації з метою виявлення суперечливої чи недостовірної інформації;
- візуальний моніторинг змінення в часі об'єктів візуалізації та візуальної сцени в цілому;
- інтелектуалізація систем ухвалення оперативних рішень;
- динамічне формування/коригування стратегії в реальному часі.

Методи аналізу текстів природною мовою

Всю сукупність наявних наразі методів аналізу текстових даних можна поділити на дві великі групи:

- статистичний аналіз,
- лінгвістичний аналіз.

Статистичний аналіз орієнтований на виявлення сенсу тексту за частотним розподіленням слів у ньому. Лінгвістичний аналіз – на виявлення сенсу тексту за його семантичною структурою. Однак казати про належність будь-якого з існуючих підходів до конкретної групи можна лише умовно. Як правило, у реальних задачах обробки тексту доводиться використовувати сучасні похідні з поєднанням методик обох груп з тим чи іншим акцентом.

Більш детальна загальна класифікація наведена на Рис.1.

Далі детальніше розглядаються найбільш значущі методи та методології обробки тексту природної мови.

Морфологічний аналіз. Основні підходи до морфологічного аналізу можна розділити на дві групи [2]: морфологічний аналіз на базі словників та без словників.

Застосування словників забезпечує можливість отримання максимальної інформації за формою відомого слова. Але тут одразу виникає питання повноти словників, що використовуються, та ризик виникнення збоїв на реальних текстах через імовірну наявність помилок.

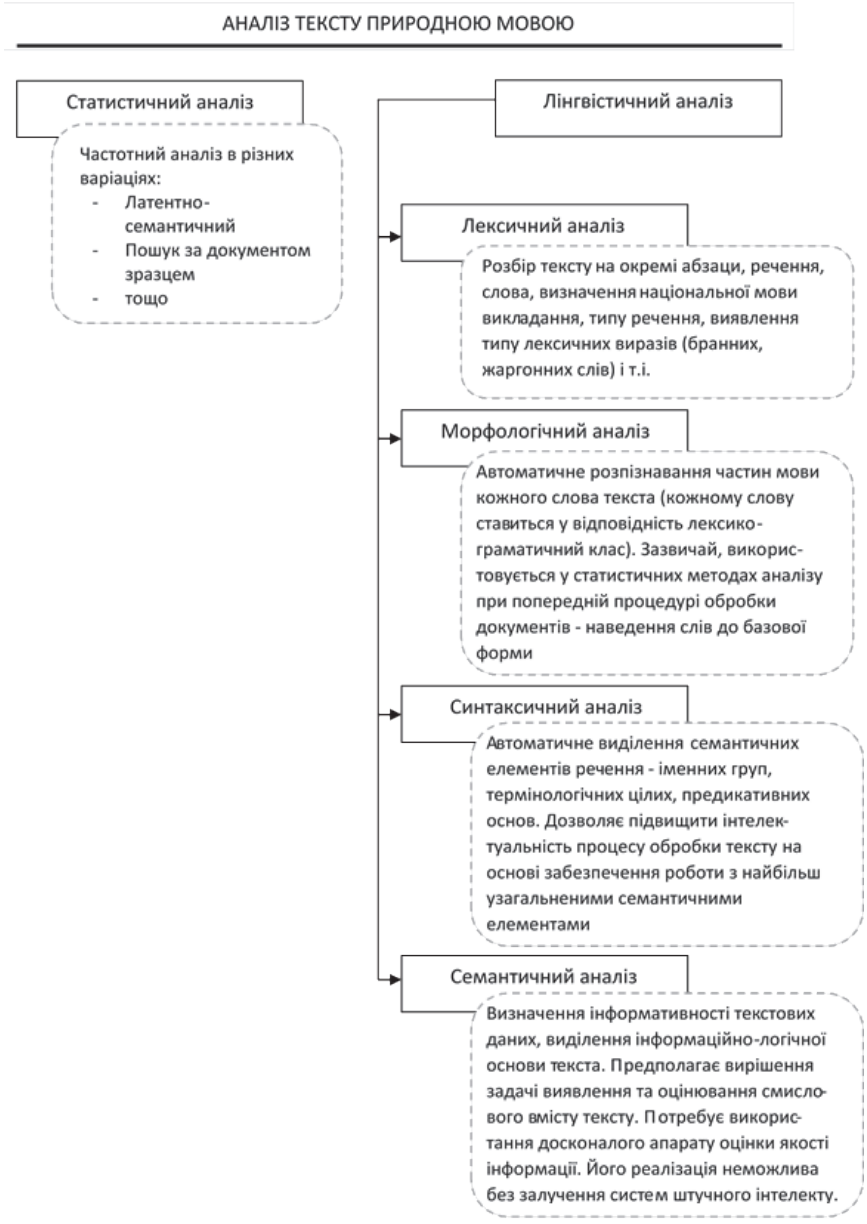

Рис.1. Загальна (умовна) класифікація методів аналізу тексту

Методи без словників для нормалізації слів використовують алгоритми, призначені для перетворення слів у різні граматичні форми. Їх можна поділити на: ймовірно статичні методи та методи лексикону основ і суфіксів.

Слід зазначити, що методи першої групи потребують великої вибірки, а для другої – потрібні великий обсяг лексиконів і методи їх отримання. Ефективність морфологічного аналізу зазвичай намагаються підвищити комбінацією різних підходів.

Морфологічний аналіз може використовувати наступні етапи обробки текстових контентів:

- розбиття тексту на окремі значущі одиниці (абзаци, речення), словоформи, відокремлюючи від тексту знаки, цифри тощо;
- нормалізація словоформ, що має вигляд лематизації або стемінгу;
- морфологічний розбір слова через пошук у лемі суфіксів та закінчень різних частин мов.

Кожний із цих етапів містить цілу низку непростих задач, вирішення яких є не тривіальною задачею.

У [3] пропонується формальний підхід, де для позначення частин мови вводиться множина частин мови:

$$Z = \{z_1, z_2, \dots, z_k\},$$

де z_i - i – та частина мови, k - кількість частин мови в обраній природній мові, а множина слів тексту представляється у вигляді об'єднання k-підмножин різних частин мови, водночас кожне слово тексту може бути віднесене до однієї з цих підмножин:

$$T = \bigcup_{j=1}^{k} W_j,$$

$t_i \in W_j, i = \underline{1,m}, j = 1,\overset{\frown}{k}$, W_j – підмножина слів j-ої частини мови.

Для відображення множини слів T до множини частин речі Z вводиться функція $F(T)$, результатом роботи якої є вектор з показниками приналежності до i-ї частини речі.

$$F: T \to X, F(T) = \{f_1(T), \dots, f_k(T)\} = \{x_1, \dots, x_k\}, i = \overset{\frown}{1,k},$$

де f_i - функція визначення показника приналежності слова до i-ї частини мови, x_i – показник приналежності слова до i-ї частини мови z_i.

Даний алгоритм є словниковим та для проведення аналізу потребує використання таблиць службових слів і таблиць лем та словоформ, з елементами яких здійснює ітераційне співставлення виокремлених із тексту словоформ.

Слід зазначити, що більшість методів даного класу стикаються з проблемою зниження якості аналізу, що обумовлене такими чинниками, як наявність у синтаксичних конструкціях декількох значень, застосування літературного стилю, наявність скорочень у тексті. У [4] автори пропонують вирішення даної проблеми через застосування словника скорочень та виконання попередньої фільтрації слів із низькою частотою появи у тексті. Йомовірно, було б доцільним залучення словників жаргонних, сленгових слів, діалектів, а також здійснювати попередню фільтрацію не лише слів із низькою частотою присутності в текстовому контенті, а й з найвищою, що забезпечить позбавлення «шумових» словоформ (займенників, прийменників тощо).

Також дуже поширеними зараз є методи морфологічного аналізу, засновані на системі машинного навчання [5]. Система машинного навчання здійснює аналіз конкретного тексту (або сукупності текстових контентів) та тренується на ньому, розпізнаючи певні закономірності, і на їхній основі робить деякі узагальнення. Згідно з набутою інформацію про властивості словоформ, що є закономірними у всіх контекстах, які пройшли аналіз, вона може робити прогнози щодо найбільш вірогідної граматичної інтерпретації словоформи у нових текстах. Методи контрольованого машинного навчання роблять прогноз, що є ймовірнісним, після тренування на корпусі текстів, розмічених інформацією про словоформи повністю або частково, а методи машинного самонавчання дозволяють працювати з корпусом, який ще не розмічений. Системи морфологічного аналізу на основі цього методу - це аналізатори, що використовують методику трансформаційних правил, виведених в результаті машинного навчання.

Слід зазначити, що крім загальнолінгвістичних задач, які підлягають вирішенню для досягнення якості аналізу тексту, існують специфічні проблеми кожної конкретної мови, пов’язані із розвитком її морфології, можливим вживанням великої кількості діалектів тощо.

Статистичний аналіз текстового контенту. Найбільш поширеними класами методів даної групи є латентно-семантичний та кластерний аналіз. Мета даних методів полягає у виявленні прихованих закономірностей або очевидних залежностей. Цим і обумовлюється їхнє особливе місце серед величезної кількості алгоритмів пошуку та обробки текстових даних.

Метод латентно-семантичного аналізу. Спершу трохи зупинимось на прак-

тичній значущості методів цього класу. Присутність у текстових контентах полісемії (одне слово має кілька різних значень) та синонімії (кілька слів з однаковим значенням) є стандартною ситуацією для будь-якої природної мови. Одним із можливих шляхів вирішення цієї проблеми – згрупувати слова з однаковими значеннями чи слова із сильною кореляцією. Їх можна представити у вигляді певної прихованої, або «латентної» змінної, яка представлятиме всі ці слова. Звідси й сам термін - «латентно-семантичний аналіз».

Формальніше метод *латентно-семантичного аналізу (LSA)* [6] — це повністю автоматичний метод витягування та виведення взаємозв'язків очікуваного контекстного використання слів в уривках дискурсу. Це нетрадиційний метод обробки природної мови або штучного інтелекту; він не використовує створені людиною словники, бази знань (БЗ), семантичні мережі, граматики, синтаксичні аналізатори, морфології тощо, а за вхідні дані приймає тільки необроблений текст, розібраний на слова, які визначаються як унікальні рядки символів, розділені на значущі фрагменти, як, наприклад, речення або параграфи. LSA є гібридним методом, який використовує комбінацію статистичних та стохастичних технік.

Метод LSA працює на колекції текстових документів. Результатом є матриця «терми-на-текстові документи», її елементи містять частоти використання термів у кожному з документів. Один із найрозповсюдженіших варіантів – LSA, заснований на використанні розкладення вихідної матриці за сингулярними значеннями (SVD). Використовуючи SVD, велика вихідна матриця розкладається на множину з k ортогональних матриць, лінійна комбінація яких є вдалим наближенням вихідної матриці. Згідно з теоремою про сингулярне розкладення, будь-яка дійсна прямокутна матриця X може бути розкладена у добуток трьох матриць:

$$X = U\Sigma V^T,$$

де матриці U та V – ортогональні, а $\mathbf{\Sigma}$ -діагональна матриця, значення на діагоналі якої називаються сингулярними значеннями матриці X. Особливість такого розкладення [7] полягає в тому, що у разі залишення лише k найбільших сингулярних значень, а в матрицях U та V лише відповідні цим значенням стовпці, то добуток отриманих матриць буде найкращим наближенням вихідної матриці X матрицею рангу k ($\dot{X}$):

$$X \cong \dot{X} = U_{lsa}\Sigma_{lsa}V_{lsa}$$

Якщо X - матриця «терми-на-документ», то $\dot{X}$, з одного боку, відображатиме основну структуру асоціативних залежностей, що є в X, а з іншого - не містить зайвої незначущої інформації (шуму).

Таким чином кожний терм та документ представляються за допомогою векторів у загальному просторі розмірності k (так званому просторі гіпотез). Тоді близькість між будь-якою комбінацією термів або документів може бути легко обчислена за допомогою різних метрик відстані (наприклад, скалярний добуток векторів, косинусна, евклідова або манхетенська відстань тощо).

Окреме питання – вибір оптимального значення розмірності k. В ідеалі, k має бути достатньо великим для відображення всієї реально існуючої структури даних. Але в той самий час достатньо малим, щоб не охопити випадкові та маловажливі залежності. Якщо обране k є занадто великим, то метод втрачає свою ефективність та наближається за характеристиками до стандартних векторних методів. Занадто маленьке k не дозволяє вміймати відмінності між схожими словами або документами. Дослідження показують, що зі збільшенням k якість спочатку зростає, а потім починає йти на спад.

На сьогодні відомі щонайменше ймовірнісний, інкрементальний та ієрархічний варіанти методу LSA, що активно застосовуються, зокрема, для автоматичного прогнозування інтересів користувачів у веб, виходячи з накопиченої інформації про їхні уподобання.

Головною перевагою LSA методу є саме його здатність виявляти залежності між словами, коли звичайні статистичні методи безсилі. Також LSA може бути використаний з навчанням (тобто з попередньою тематичною класифікацією текстових контентів) або без навчання (довільне розбиття довільного тексту), що залежить від задачі, яка вирішується.

Одним із головних недоліків латентно-семантичного аналізу є те, що, він розрахований на обробку документів колекції, тож ці документи мають бути доступними. Це обмежує його застосування. До цього можна ще додати значне зниження швидкості обчислень зі збільшенням оьсягу вхідних даних. Як продемонстровано у [8], швидкість обчислень відповідає порядку N^{2k}, де $N = N_{doc} + N_{term}$ – сума числа документів та числа термів, k – розмірність простору факторів.

Методи *Кластерного аналізу* [9] являють собою статистичну процедуру, задача якої полягає в розбитті вибірки інформаційних об'єктів на підмножини, що не перетинаються (жорстка кластеризація) і називаються кластерами. (У випадку м'якої кластеризації один об'єкт може належати кільком кластерам.) Кожен кластер має складатися зі схожих об'єктів, а об'єкти різних кластерів мають істотно відрізнятися один від одного.

Формально, є певна вибірка інформаційних об'єктів $X^l = \{x_1, \dots, x_l\} \subset X$ і функція відстані між об'єктами - $\rho(x, x')$. Треба розбити задану вибірку на кластери, що містять об'єкти, близькі за метрицею ρ. Кожному об'єкту $x_i \in X^l$ призначається мітка (номер) кластера y_i. Алгоритм кластеризації фактично є функцією $a: X \to Y$, яка будь-якому об'єкту $x \in X$ ставить у відповідність мітку кластера $y \in Y$. Множина міток Y в деяких випадках відома заздалегідь, однак частіше завдання полягає у визначенні оптимального числа кластерів з точки зору того чи іншого критерію якості кластеризації.

Методи кластеризації різняться правилами побудови кластерів [10], що є критеріями, за якими визначається «схожість» об'єктів. Кластери можна утворювати, ґрунтуючись на відстані між ними, щільності ділянок у просторі даних, інтервалах або на конкретних статистичних розподілах. Усе залежить від конкретного набору даних та мети використання результатів аналізу.

Методи кластерного аналізу можуть бути застосовані для вирішення цілої низки задач обробки текстових даних, які умовно можна об'єднати в 4 групи:

1) розробка системи класифікації інформаційних об'єктів;

2) дослідження корисних концептуальних схем їх групування;

3) представлення гіпотез на основі дослідження текстових даних;

4) перевірка гіпотез або досліджень для визначення, чи дійсно типи (групи), виділені тим або іншим способом, присутні в наявних текстових даних.

Нескладно помітити, що перелічені групи задач фактично є задачами машинного навчання. Тобто простежується тісний зв'язок між алгоритмами і методами машинного навчання та методами кластерного аналізу.

Одиницею кластерного аналізу є інформаційний об'єкт, заданий вектором ознак. Робота кластерного аналізу спирається на два припущення. Перше – розглянуті ознаки об'єкта в принципі допускають бажане розбиття сукупності об'єктів на кластери. Друге – правильність вибору масштабу або одиниці вимірювання ознак.

Усю сукупність методів кластерного аналізу можна поділити на дві групи: ієрархічні та неієрархічні, кожна з яких включає безліч методів та підходів. Суть ієрархічної кластеризації полягає в послідовному об'єднанні менших кластерів у більші (агломеративні методи) або поділі більших кластерів на менші (дивізимні). Тобто в першому випадку спочатку всі об'єкти є окремими кластерами і послідовно, крок за кроком, схожі об'єкти поєднуються в кластер, кластери збільшуються, а їх кількість зменшується. Друга група – логічна протилежність першій. Спочатку - всі об'єкти

належать одному кластеру, який на наступних кроках алгоритму ділиться на менші кластери, у результаті утворюється послідовність груп, що розщеплюються.

Задача кластеризації є досить суб'єктивною. Для її розв'язання може існувати більше одного правильного алгоритму. Кожен алгоритм дотримується свого набору правил для визначення «подібності» між об'єктами даних. Найбільш відповідний алгоритм кластеризації для конкретної проблеми часто потрібно вибирати експериментально, якщо немає математичної причини віддати перевагу одному алгоритму кластеризації над іншим. Алгоритм може добре працювати на певному наборі даних, але не працюватиме для іншого.

Програмна реалізація алгоритмів кластерного аналізу широко представлена в різних інструментах Data Mining, які дозволяють вирішувати завдання досить великої розмірності.

Взагалі *обробка природної мови* (Natural language processing - NLP) є загальним напрямком штучного інтелекту і математичної лінгвістики. NLP вивчає проблеми комп'ютерного аналізу і синтезу природних мов. Що ж до штучного інтелекту, аналіз означає розуміння мови, а синтез - генерацію грамотного тексту. Більшість NLP-методів є методами машинного навчання.

Методи машинного навчання в загальному процесі семантичного аналізу тексту. Машинне навчання (ML) є одним із визнаних успішних методів обробки даних для отримання з них корисної інформації. Їхньою особливістю є не прямий розв’язок задачі, а навчання на множині подібних прикладів, що дозволяє використовувати ці методи для обробки великих обсягів даних та виявляти в них нові, нетривіальні, корисні та доступні для інтерпретації знання. Існує твердження, що алгоритми ML вчаться витягувати інформацію із даних тим краще, чим більше даних для них доступно [11]. Окрім методів штучного інтелекту, під час розробки моделей машинного навчання як допоміжні використовуються засоби математичної статистики, чисельних методів, методів оптимізації, теорії ймовірностей, теорії графів тощо [11].

Найбільш використовуваними варіантами застосування моделей ML для обробки текстових даних – вирішення задач їх *класифікації* та *кластеризації*.

Класифікація [12] – встановлення функціональної залежності між вхідними і дискретними вихідними змінними. За допомогою класифікації вирішується завдання приналежності об’єктів до одного з відомих класів.

Кластеризація [12] – групування об’єктів на основі їхніх властивостей. Об’єкти в кластері мають бути схожими і відрізнятися від об’єктів інших кластерів. Чим більша схожість об’єктів усередині кластера і чим більше відмінностей між кластерами, тим точніша кластеризація.

Методи машинного навчання поділяють на дві основні категорії [13]: навчання з учителем (supervised) та навчання без учителя (unsupervised). Методи навчання з учителем поділяють вхідні дані на набір наперед заданих класів. Для навчання такого класифікатора потрібна навчальна вибірка, яка містить марковані зразки різних класів. Навчальна вибірка має бути репрезентативною, тобто містити варіативний контент із різними характеристиками (класифікаторами), щоб моделі могли відповідно навчитися їх розрізняти. З використанням цього набору даних можна навчити модель розпізнавати ознаки того чи іншого класу контенту.

Методи навчання без учителя не потребують навчальних даних, проте вони не ставлять у відповідність вхідним даним певний клас, а лише вивчають закономірності у вхідних даних та поділяють вхідні дані на кластери.

У [13] автори систематизували існуючі типи класифікаторів за різними критеріями, результат наведений у таблиці 1 нижче.

Таблиця 1. Різновидності підходів до класифікації залежно від критеріїв

Критерій	**Тип**	**Короткий опис**
Використання/ не використання навчальних даних	Класифікація з учителем	Вхідні дані поділяють, використовуючи набір зразків як навчальні дані
	Класифікація без учителя	Підходи відомі як кластеризація. Не беруть до уваги мітки навчальних даних для класифікації вхідних даних
	Напівавтоматичне навчання	Навчання відбувається з використанням даних як з мітками, так і без
Врахування/ не врахування будь-якого припущення про розподіл вхідних даних	Параметричні класифікатори	Припускається, що функція щільності ймовірності для кожного класу відома
	Непараметричні класифікатори	Класифікатори не обмежуються жодними припущеннями про розподіл вхідних даних
Розгляд одного класифікатора або ансамблю	Один	Використовується єдиний класифікатор для призначення мітки для об’єкта
	Ансамбль	Під час визначення мітки для об’єкта враховуються результати кількох класифікаторів
Використання/не використання технології жорсткого поділу, де кожен об’єкт належить лише одному кластеру	Жорсткий класифікатор	Технології жорсткої класифікації не враховують подальші зміни різних класів
	М’який (нечіткий) класифікатор	Нечіткі класифікатори моделюють поступові граничні зміни, забезпечуючи оцінку ступеня подібності всіх класів
Видача класифікатором розподілу ймовірності належності до всіх класів	Імовірнісний класифікатор	Класифікатор здатен для заданого зразка оцінити розподіл ймовірностей на множині класів
	Неймовірнісний класифікатор	Підхід визначає лише найбільш придатний клас для вхідного образу

Найбільш поширеними методами ML для задач класифікації [14] є штучні нейронні мережі [15], логістична регресія [15], метод опорних векторів [15] та випадковий ліс [16]. Порівняльна характеристика перелічених та інших методів класифікації наведена в [17].

З огляду на тематику даного дослідження, вирішення задачі класифікації для вхідної інформації може дозволити, наприклад, вибрати з множини вхідних повідомлень лише ті, що будуть значущими для візуалізації (стосуватися конкретної предметної області чи події), обрізати фейковий контент [17] чи просто зайвий, класифікувати повідомлення на такі, що відображають статичну та динамічну інформацію, або за призначенням (кому інформація має бути делегована), за темою, відправником, датою, типом повідомлення тощо [1].

Генеративний штучний інтелект

Основною відмінністю технологій генеративного штучного інтелекту (ШІ) є їхня здатність створювати нові дані будь-якого типу. Технології ШІ-генерації використовують штучний інтелект для створення на основі вхідних текстових описів нового контенту, який досить точно (текстово або візуально) відображає зміст і контекст вхідного тексту. Умовно їх можна розділити на три групи за формою результуючого контенту: *text-to-text* (генерується текст), *text-to-image* (генерується зображення) та *text-to-video* (генерується відео).

Технології генерації тексту (або «генерування природної мови») часто базуються на процесах Маркова та на глибоких генеративних моделях. Штучний інтелект або моделі машинного навчання генерують мову згідно правил граматики, синтаксису чи лексики. Модель генерації починається на концептуальному рівні (вибір з контенту даних для перетворення), зменшується до досконалих правил мови (правопис, граматика, вибір слів), та, врешті-решт, створює речення як ланцюжок слів.

Більшість сучасних сервісів *text-to-image* є поєднанням моделі, що обробляє природну мову (NLP), та генеративної змагальної мережі (GAN).

Технології *text-to-image* вміють «прочитати» фрагмент тексту та згенерувати зображення відповідно до його змісту. Як правило, це реалізується трьома кроками. Спочатку виконується аналіз вхідного опису та виявляється значуща інформація (зазвичай для цього використовуються методи аналізу природної мови): ключові слова, сутності, контекст опису. Далі на основі отриманої інформації створюється зображення, використовуючи здебільшого генеративні змагальні мережі (GAN) [31]. Й нарешті вдосконалення результуючого зображення – багатоітераційний процес аналізу та оптимізації версій зображення до досягнення бажаної якості та ступеня відображеня семантичного вмісту.

Технології *text-to-video* уможливлюють створення відеороліків на основі вхідної текстової підказки, та зазвичай охоплюють кілька підгалузей штучного інтелекту: обробку природної мови, комп'ютерний зір та машинне навчання. Стрімкий розвиток технологій генерації відеоконтенту здебільшого пов'язаний із розвитком дифузійних моделей (Stable Diffusion - SD). Вхідні описи можуть мати чималий обсяг. Будь-яке його змінення, навіть додавання/видалення одного слова в/з опис(у), може кардинально впливати на результат генерації. Кожне слово текстової підказки відіграє ключову роль у створенні відео. Приблизний алгоритм генерації можна розглянути на прикладі відомої технології T2V. Алгоритм T2V полягає в наступній послідовності основних кроків:

1) аналіз та інтерпретація вхідного тексту за допомогою методів токенізації з метою визначення його семантики – контексту, значення;

2) вибір відповідних візуальних ефектів та анімації, планування відеоконтенту на основі вхідної текстової підказки;

3) створення візуальних елементів на кшталт 3D-моделей або анімації. Для цього можуть бути використані GAN моделі або ці елементи можуть бути витягнуті просто з наявної бібліотеки відеоматеріалів;

4) побудова з отриманих візуальних об'єктів послідовності, що відповідає вхідному опису, додаючи в цю послідовність переходи та, можливо, синхронізуючі її зі звуком.

Слід зазначити, що розвиток ШІ технологій відеогенерації припадає лише на останні кілька років, тому існує ще багато невирішених проблем та обмежень. Так, наприклад, досі залишається серйозною проблемою генерація руху між відеокадрами.

Аналіз існуючих рішень

Задачі обробки природної мови є досить складними, але більшість із них уже реалізовано в існуючих готових застосунках та бібліотеках аналізу тексту. Генеративні моделі хоча і є досить молодою галуззю, однак вже існує чимала кількість го-

тових рішень. Тому першочергова задача полягає у виборі засобів реалізації з урахуванням вимог до функціональності та можливості подальшої інтеграції в систему, розвитку та вдосконалення.

Серед існуючих інструментів обробки тексту слід виділити:

- Морфологічний аналізатор *pymorphy2* [18] реалізований мовою програмування Python з додатковими розширеннями C ++. Він уміє надавати слову потрібної форми. Наприклад ставити у множину, або змінювати відмінок; повертати граматичну інформацію про слово (число, рід, відмінок, частина мови тощо). Використовує великі ефективно закодовані словники, створені на основі даних OpenCorpora та LanguageTool. Для забезпечення морфологічного аналізу розроблено набір лінгвістично мотивованих правил. Для російської мови pymorphy2 забезпечує ультрасучасний морфологічний аналіз. Але для української мови - вимагає більше специфічних правил для обробки слів позасловникового запасу, а також потребує анотованого корпусу української мови, бо підтримка української мови в цьому аналізаторі поки є експериментальною.

- *Text Analyzer* [19] – безкоштовний інструмент для пошуку та оптимізації ключових слів у тексті. Часто використовуються у вебсередовищі для аналізу сайтів, пошукових запитів, аналізу описів застосунків для Google Play та App Store тощо. Виконує частотний аналіз тексту, дозволяє виділяти ключові слова та спам.

- *SenseClusters* [20] – пакет програм (переважно мовою Perl), що дають можливість користувачу віднести до одного кластеру схожі контенти, використовуючи методи некерованого машинного навчання. Є досвід використання інструментів SenseClusters для розділення слів за їхнім сенсом, категоризації електронної пошти, дискримінації імен. Підтримує кілька різних методів кластеризації тексту. Включають власні методи SenseClusters techniques та LSA метод.

SenseClusters базується тільки на лексичних характеристиках та не використовує ніяких навчальних даних або зовнішніх джерел знань. Це обумовлює його мовну незалежність. Єдиною умовою є можливість токенізації мови за допомогою регулярних виразів Perl, що задаються користувачем. Загалом SenseClusters можна використовувати для вирішення будь-якої задачі, що потребує розпізнавання контекстуально схожих одиниць тексту або слів, які зустрічаються в подібних контекстах.

- Бібліотека для роботи з матрицями *JAMA* [21]. Базовий пакет лінійної алгебри для Java. Надає класи рівня користувача для побудови реальних щільних матриць і керування ними. Підтримує п’ять фундаментальних матричних розкладів, в тому числі сингулярне розкладання прямокутних матриць, що обумовлює широке використання бібліотеки для реалізації LSA методів.

- Електронний словник української мови *ВЕСУМ* [22] - це великий словник словозміни української мови, основними компонентами якого є реєстр лем, коди класів словозміни й правила генерації словоформ на основі цих кодів, а також застосування елементів програмованої логіки. Виконує завдання морфологічного аналізу й синтезу. Перше полягає в лематизації (зведенні окремої словоформи до леми) й присвоєнні цій словоформі відповідних граматичних тегів, а друге передбачає генерування всіх словоформ із певної леми з відповідними граматичними ознаками-тегами. Підтримує «динамічне тегування», а саме обробку утворюваних слів (складних іменників, прикметників, прислівників) шляхом розбиття їх на складові й присвоєння цим складовим відповідних граматичних ознак-тегів [23].

- *NLP UK* [24] - інструмент для аналізу та обробки української мови на основі словника ВЕСУМ (що використовується для тегування лексем) та двигуна LanguageTool (для аналізу текстів). Має підтримку токенізації, лематизації, частиномовного аналізу та базового зняття омонімії. Застосовувався на python3 та java.

- Браунський корпус української мови *БрУК* [25] відкритий, збалансований за жанрами та в майбутньому проанотований корпус сучасної української мови обся-

гом 1 млн слововживань. Корпус побудований на засадах, покладених в основу відомого корпусу англійської мови Brown. Його частина, проанотована за сутностями та готова для автоматичного анотування сутностей (люди, організації, локації та різне); містить векторні представлення слів, простий у використанні токенизатор (на абзаци, речення та слова) тощо.

- *UD Ukrainian* [26] - корпус дерев залежностей для української мови. Містить 122 тисячі токенів у 7000 реченнях художньої літератури, новин, статей, думок, Вікіпедії, юридичних документів, листів, дописів і коментарів за останні 15 років та першу половину 20 століття.

- *Java CoreNLP* [28] – пакет для обробки природної мови на Java. Дозволяє отримувати лінгвістичні анотації для вхідного текстового контенту, включаючи токени, речення, частини мови, іменовані сутності, числові та часові значення, проводити аналіз залежностей, зв'язків, виокремлювати почуття, настрої (тобто емоційні аспекти) та посилання на джерела цитування. Наразі, CoreNLP підтримує 6 мов (арабську, китайську, англійську, французьку, німецьку та іспанську, і, на жаль, не підтримує українську мову).

- *NLTK* (Natural Language Toolkit) [29] — це платформа для розробки програм обробки природної мови мовою програмування Python. Надає зручні інтерфейси для багатьох мовних корпусів, а також бібліотеки для обробки текстових даних, а саме: класифікації, токенізації, стемінгу, розмітки частин мови (POS-тегування), синтаксичного та семантичного аналізу.

- *Stanza* [27] – це пакет засобів на Python для аналізу природної мови (для понад 70 мов). Містить інструменти для декомпозиції тексту у списки речень і слів, для створення базових форм цих слів, визначення частин мови та морфологічних особливостей. Stanza побудовано з високоточними компонентами нейронної мережі, які забезпечують ефективне навчання та оцінку за допомогою власних анотованих даних. Забезпечує надійну текстову аналітику, включаючи токенізацію, розширення багатослівних маркерів (MWT), лематизацію, визначення тегів частин мови (POS) і морфологічних ознак, синтаксичний аналіз залежностей і розпізнавання іменованих об'єктів. Модулі побудовано на основі бібліотеки PyTorch. Крім того, Stanza включає інтерфейс Python до пакета Java CoreNLP і успадковує звідти додаткову функціональність, таку як синтаксичний аналіз, кореферентна роздільна здатність та зіставлення лінгвістичного шаблону.

Проведений аналіз дозволив визначити основні критерії вибору засобів обробки тексту для подальшого використання у вирішенні задачі, а саме - підтримуванні функціональності, відкритості коду та мультимовності.

Інша група – генеративні моделі. Детальний опис проаналізованих генеративних моделей наведений в [30]. Здійснений аналіз дозволив сформувати наступну множину критеріїв вибору такої моделі для реалізації задачі:

- достатня смислова якість візуалізації (тобто можливість отримання адекватного результату);
- відкритість коду, що дає можливість удосконалення, доробки та кастомізації моделі;
- доступна вартість (в ідеалі безкоштовний варіант);
- простота та зручність використання;
- технічна якість візуалізації.

Задача візуалізації семантики текстового контенту

Задача полягає у створенні автоматизованого алгоритму, що реалізує візуалізацію саме семантичного вмісту вхідних текстових даних, отриманих з різних джерел: повідомлення чатів, месенджерів, контент електронних листів, текстові файли тощо. Загальний технологічний ланцюжок вирішення задачі на верхньому рівні наведений на рисунку 2.

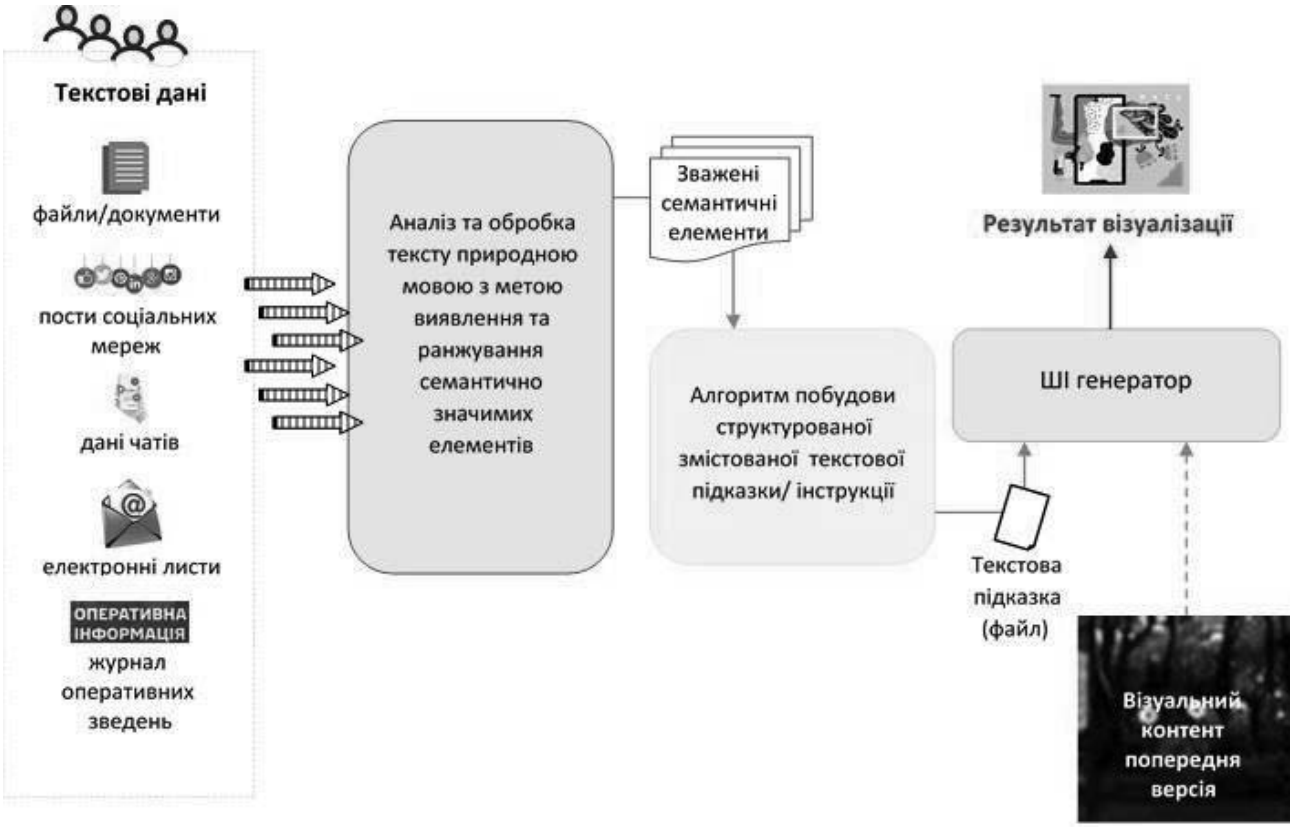

Рис. 2. Алгоритм вирішення задачі візуалізації семантики тексту

Алгоритм має забезпечувати побудову файлу інструкцій (підказок) для візуалізації на базі вибраних в ході аналізу первинних текстових даних ключових семантичних елементів. Файл інструкцій є вхідним для моделі text-to-image та має містити підказку у вигляді структурованих текстових даних, структура та зміст яких забезпечує якісну, достовірну за семантикою візуалізацію. Фактично це є перетворенням тексту в текст визначеної структури зі збереженням ключових семантичних елементів, виявлених під час обробки тексту.

Результат візуалізації є графічним підґрунтям для визначення певних показників, що дозволяють математично оцінити результат вирішення прикладної задачі. Це може бути, приміром, ступінь зміни системи в часі (від попередньої візуалізації) або достовірність вхідної інформації. Такі метрики якості даватимуть можливість оцінити та удосконалити не лише результат візуалізації, а й алгоритм в цілому.

Визначення ключових семантик у вхідному текстовому описі. Ця підзадача в дечому подібна до задач автоматичного реферування, де кінцевою метою є формування стислого та змістовного конспекту, але він формується саме з ключових семантик, виявлених під час аналізу й обробки. Це є однією з найскладніших задач обробки природної мови. Задача даного рівня полягає у виділенні з первинних текстових даних семантичних елементів - ключових фраз (фрагментів, речень), що відображатимуть семантику вхідного тексту.

Обробка тексту природної мови включає наступні основні задачі (див. рис. 3): поділ тексту на токени, переведення всіх літер у нижній регістр, виокремлення основи слова, отримання базової словникової форми слова, видалення стопслів, нормалізацію тексту, розмічування тексту на частини мови, виявлення іменованих сутностей і, нарешті, витягнення з неструктурованого або мало структурованого тексту структурованої інформації, такої, як сутності, зв’язки між ними, їхні атрибути.

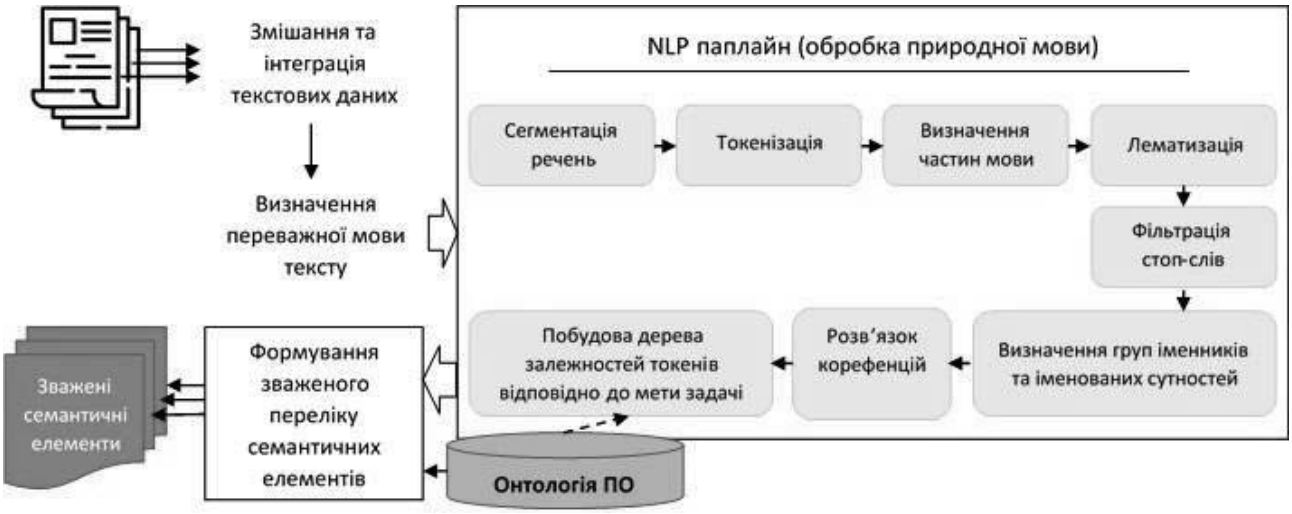

Рис. 3. Задачі обробки текстових даних

Формування файлу підказки. Зрозуміло, що такий алгоритм не може бути повністю універсальним. Досягти високого рівня ефективності (семантичної достовірності зображення відповідно до вхідних даних) можна тільки враховуючи особливості класу прикладних задач, де він буде застосовуватися, та відповідної предметної області. Це дозволяє сформувати адекватну систему характеристик, що уможливлюють певну якість отримання візуалізації. Це можуть бути класи значущих семантик, основні види дій та станів, географічні локації, часові характеристики, використання версійності та номер попередньої версії.

Алгоритм формування файлу підказки наведений на Рис. 4.

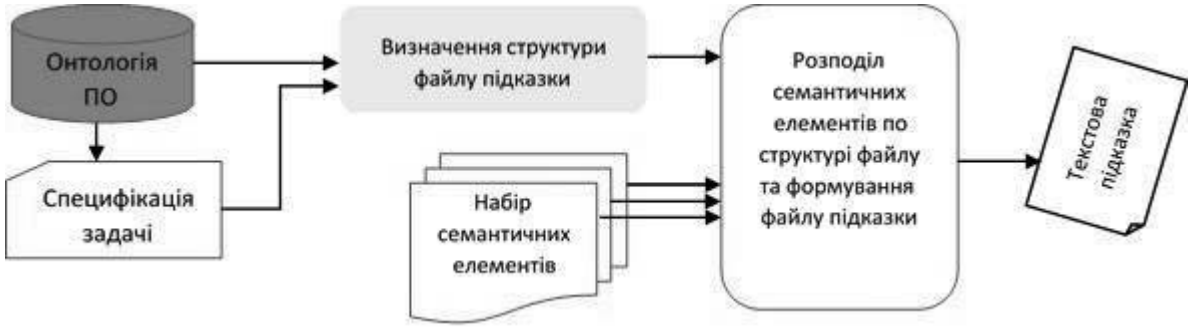

Рис. 4. Формування файлу підказки

Висновки

В роботі проведено аналіз методів обробки тексту природною мовою з метою виявлення його семантичного змісту та технологій генеративного штучного інтелекту, а також найбільш вживаних на сьогодні застосунків, як для аналізу текстових даних, так і ШІ-генерації. Це дозволило виокремити спільні характеристики існуючих систем, їхні недоліки та сформувати основні критерії вибору застосунку ШІ-генерації для його подальшої інтеграції до загальної системи для реалізації функцій створення візуального контенту. На основі проведеного аналізу вироблено основний ланцюжок вирішення задачі, специфіковано складові підзадачі та визначені головні етапи проведення досліджень. Наукова новизна запропонованого алгоритма полягає у:

1) вдосконаленні процесу обробки тексту шляхом зведення та інтеграції текстових даних з різних джерел;

2) створенні технології інтеграції сервісів обробки тексту та візуалізації шляхом алгоритмізації та автоматизації етапу переходу від сформованого списку значущих візуальних елементів тексту до візуалізації цих семантик засобами ШІ генерації;

3) визначенні метрик якості результатів візуалізації.

Подальший розвиток досліджень передбачає, перш за все, вдосконалення аналізу текстових даних з метою отримання більш якісної семантики відповідно до особливостей та цілей вирішення прикладної задачі, а саме охоплює вирішення наступних задач:

1) створення системи критеріїв значущої інформації в аспекті прикладної задачі, що вирішується. Наприклад, як системи з двох онтологій: онтології предметної області, де однією з характеристик сутності чи зв'язку між сутностями є вага (тобто, це саме семантична вага, що враховує інтереси прикладної задачі, а не вага терміну, що визначається частотою його вживання) та онтології задачі з визначенням ролей користувачів, джерел надходження текстової інформації, сутностей, що визначаються практичними цілями реалізації.

2) вирішення задачі класифікації вхідних текстових даних за заданою системою критеріїв. Як класифікатори можуть використовуватись тематика, джерела інформації, ролі кінцевих користувачів, локації, показники часу, яким датується вхідний контент тощо (залежно від практичної задачі, що вирішується);

3) створення технології "очищення" нелітературної мови (обробка суржика, сленгу, діалектів) та нормалізації тексту із залученням словників синонімів;

4) удосконалення процесу аналізу тексту з урахуванням критеріїв значимості у виявленні ключових семантичних фрагментів.

Ще одна задача, яка може бути реалізована на основі класифікації вхідних текстових даних/вихідних візуалізацій за ролями користувачів за попередньо визначеними критеріями/правилами, - це створення системи "маршрутизації завдань", яка пов'язує вхідні дані й, відповідно, отриманий відеоконтент із адресатом (роль користувача).

Література

1. Yakymenko, D. O., Kataieva, Ye. Ye. Methods and Means of Intelligent Analysis of Text Documents. Вісник Черкаського державного технологічного університету, 2022. №2, С. 43-52. https://er.chdtu.edu.ua/handle/ChSTU/4165
2. Bisikalo, O., Vysotska, V., Burov, Y. Conceptual Model of Process Formation for the Semantics of Sentence in Natural Language. Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2020). Volume I: Main Conference Lviv, Ukraine, April 23-24, 2020, 27p. CEUR Workshop Proceedings, available at: http://ceur-ws.org/Vol-2604/paper12.pdf
3. Іващенко О. О., П'ятикоп О. Є. Моделювання методу морфологічного аналізу україномовного тексту. Наукові праці ДонНТУ, Серія "Інформатика, кібернетика та обчислювальна техніка", 2020. № 2(31). С. 65 -72. https://iktv.donntu.edu.ua/wp-content/uploads/2021/04/08_Yvashchenko-Piatykop-1.pdf
4. Singh, J., Singh, G., Singh, R. Morphological evaluation and sentiment analysis of Punjabi text using deep learning classification. Journal of King Saud University: Computer and Information Sciences. 2021, Vol.33, № 5. P. 508 - 517. https://www.sciencedirect.com/science/article/pii/S1319157818300612?via%3Dihub
5. Яровий А., Кудрявцев Д., Крилик Л. Удосконалення методу семантичного аналізу тексту. Інтелектуальні Інформаційні Технології. 2020, С. 34-36. https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/30887/WORK-IES-2020-34-36.pdf?sequence=1
6. Landauer T., Foltz P., Laham D. Introduction to Latent Semantic Analysis. Discourse Processes, 1998. № 25. P. 259–284.
7. Press, W., Teukolsky, S., Vetterling, W. Singular Value Decomposition. Numerical Recipes in C., 2nd edition. Cambridge: Cambridge University Press, 1992. P. 59 -71.

8. Deerwester, S., Dumais, S., Furnas,G. Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 1990. Vol. 41 № 6. P. 391–407. URL: http://wordvec.colorado.edu/papers/Deerwester_1990.pdf
9. Основні поняття кластеризації та постановка задачі. https://csc.knu.ua/media/study/asp/mod_probl_inf_tech_sys_analysis_ivohin/lecture/lec11.pdf
10. Методи кластерного аналізу. Ієрархічні методи. https://moodle.znu.edu.ua/pluginfile.php/486140/mod_resource/content/1/Лекція%2010.pdf
11. Grolinger, K., Hayes, M., Higashino, W.A. Challenges for MapReduce in big data in Proc. IEEE World Congr. Services (SERVICES), 2014, pp. 182-189. https://ir.lib.uwo.ca/cgi/viewcontent.cgi?article=1095&context=electricalpub
12. Коновалова К. Машинне навчання: методи та моделі: підручник для бакалаврів, магістрів та докторів філософії спеціальності 051 «Економіка»// Харків: ХНУ імені В. Н. Каразіна, 2020. 280 с. https://www.researchgate.net/publication/345765254_MASINNE_NAVCANNA_METODI_TA_MODELI
13. Новіков О.М., Лавренюк М.С. Огляд методів машинного навчання для класифікації великих обсягів супутникових даних. Системні дослідження та інформаційні технології. 2018. № 1. С. 52-71. http://jnas.nbuv.gov.ua/article/UJRN-0001075162 (дата звернення: 01.06.2024)
14. Maulik, U., Chakraborty, D. Remote Sensing Image Classification: A survey of support-vectormachine-based advanced techniques. IEEE Geoscience and Remote Sensing Magazine. 2017. Vol. 5, № 1. P. 33–52.
15. Bishop C.M. Pattern Recognition and Machine Learning. NY: Springer. 2006. 738 p.
16. Gislason, P.O., Benediktsson, J.A., Sveinsson, J.R. Random forests for land cover classification. Pattern Recognition Letters. 2006. Vol. 27 N 4. P. 294–300.
17. Праздніков В.О., Сугоняк І.І. Моделі та методи машинного навчання для розпізнавання фейкового контенту. Технічна інженерія, 2023. Том 2 №92. С.131-136. https://www.researchgate.net/publication/376878645_Modeli_ta_metodi_masinnogo_navcanna_dla_rozpiznavanna_fejkovogo_kontentu
18. Морфологічний аналізатор Pymorphy2. https://pymorphy2.readthedocs.io/en/stable/
19. Text Analyzer. https://asomobile.net/en/blog/text-analyzer/
20. Sense Clusters. https://metacpan.org/pod/Text::SenseClusters
21. JAMA: A Java Matrix Package. https://math.nist.gov/javanumerics/jama/
22. Рисін, А., Старко, В. Великий електронний словник української мови (ВЕСУМ). Вебверсія 6.1.0. 2005-2023. https://vesum.nlp.net.ua/
23. Старко, В., Рисін., А. Великий електронний словник української мови (ВЕСУМ) як засіб NLP для української мови. Галактика слова. 2020. С.134-141. https://www.researchgate.net/publication/344842033_Velikij_elektronnij_slovnik_ukrainskoi_movi_VESUM_ak_zasib_NLP_dla_ukrainskoi_movi_Galaktika_Slova_Galini_Makarivni_Gnatuk
24. LanguageTool API NLP UK. URL: https://github.com/brown-uk/nlp_uk
25. Браунський корпус української мови. https://github.com/brown-uk/corpus
26. Universal Dependencies corpus for Ukrainian. https://github.com/UniversalDependencies/UD_Ukrainian-IU/tree/master
27. Stanza – A Python NLP Package for Many Human Languages. https://stanfordnlp.github.io/stanza/
28. Manning, C., Surdeanu, M., Bauer, J. The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014. P. 55-60
29. Natural Language Toolkit: Documentation. 2024. https://www.nltk.org/
30. Міненко В., Аналіз застосування ШІ-генераторів для розв'язання складних бізнес-задач. Системи керування та комп'ютери, 2024, № 4. С. 10 – 18.
31. Іванов А., Онищенко В. Методи генерації зображень з використанням мереж GAN. Адаптивні системи автоматичного управління. 2023. Том 1 №42, С.153-159 https://asac.kpi.ua/article/view/279109

Одержано: 10.01.2025
Внутрішня рецензія отримана: 17.01.2025
Зовнішня рецензія отримана: 19.01.2025

Про автора:

Міненко Валерій Дмитрович,
аспірант
https://orcid.org/0009-0003-5299-6786

Місце роботи автора:

Інститут програмних систем
НАН України.
Засновник, Twigames Inc.
3422 Old Capitol Trail, Suite# 241,
Wilmington, DE 19808, US
Моб. тел.: +380(68) 807 49 48
E-mail: valerii@twigames.net