DOI: https://doi.org/10.15276/ict.02.2025.14
УДК 004.93

Метод реалізації комплексної модульної системи аналізу інформаційних джерел

Угрин Дмитро Ілліч[1)]
Д-р техніч. наук, професор каф. Комп'ютерних наук
ORCID: https://orcid.org/0000-0003-4858-4511; d.ugryn@chnu.edu.ua. Scopus Author ID: 57163746300
Ушенко Юрій Олекснадрович[1)]
Д-р фіз.-матем. наук, професор каф. Комп'ютерних наук
ORCID: https://orcid.org/0000-0003-1767-1882; y.ushenko@chnu.edu.ua. Scopus Author ID: 6701840218
Каланча Артем Дмитрович[1)]
Аспірант каф. Програмного забезпечення комп'ютерних систем
ORCID: https://orcid.org/0009-0004-1451-7470; kalancha.artem@chnu.edu.ua
[1)] Чернівецький національний університет ім. Ю. Федьковича, вул. Коцюбинського, 2. Чернівці, 58012, Україна

АНОТАЦІЯ

У даних тезах представлено метод побудови модульної інформаційної системи для збору, обробки та аналізу текстових повідомлень з відкритих джерел, зокрема Telegram-каналів. Система поєднує процедури очищення й нормалізації тексту з розрахунком метрик: Cosine Similarity, часо-семантичного впливу (TSI), класифікації ворожої риторики, кластеризації та побудови графа взаємозв'язків. Для забезпечення стійкості використано асинхронний обробник із чергою повідомлень та шаблон Circuit Breaker. TSI дозволяє виявляти міжканальний вплив навіть за низької лексичної схожості, тоді як модуль ворожої мови аналізує риторику на рівні повідомлень і джерел. Система формує автоматичні висновки для кожного каналу та підтримує візуалізацію результатів, що підвищує швидкість і зручність аналітики. Розроблене рішення має прикладне значення для OSINT, інформаційної безпеки та дослідницьких завдань.

Ключові слова: Timed Semantic Influence, косинусна подібність, ворожа мова, кластеризація джерел, граф інформаційного впливу, автоматизований збір даних

Актуальність даного дослідження обумовлена високою динамікою, фрагментарністю та великою кількістю джерел, що впливають на формування суспільної думки. Особливо це стосується соціальних платформ і месенджерів, де відсутність редакційного контролю дозволяє поширювати інформацію будь-якого змісту: від новин і аналітики до дезінформації, маніпуляцій і ворожої риторики. У цьому контексті виникає потреба у створенні ефективних інструментів, здатних здійснювати автоматизований збір, обробку та аналітику повідомлень з відкритих джерел.

Особливу складність становить аналіз великої кількості текстових даних у реальному часі або близькому до реального. Питання не обмежується лише збором повідомлень. Важливо виявляти зв'язки між джерелами, оцінювати їхню риторику, виявляти координовану активність або інформаційні впливи, що можуть бути неочевидними при поверхневому аналізі. При цьому враховується як зміст повідомлень, так і часові патерни їх поширення.

Метою дослідження є формування методу реалізації архітектури та функціональних компонентів інформаційної системи, призначеної для аналізу відкритих джерел текстової інформації, яка б поєднувала інструменти збору, попередньої обробки, аналізу природної мови та візуалізації результатів. Така система повинна забезпечувати масштабованість, стійкість до збоїв, прозорість обчислень, а також можливість розширення для подальшого дослідження нових джерел та метрик. У межах дослідження розглядається набір технічних та аналітичних методів, а також підходу модульної структури, що дозволяє виконувати гнучкий, багатоетапний аналіз текстових повідомлень. Результати такого аналізу можуть бути використані як у наукових дослідженнях, так і в прикладних задачах, зокрема для виявлення інформаційних впливів, формування інформаційних графів, визначення рівня ворожості або тематичної класифікації джерел.

Дослідження MediaRank показує, що ранжування десятків тисяч джерел за цитуваннями, змістом і популярністю дозволяє оцінювати їхню якість та вплив [1], проте не відображає

This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/deed.uk)

часо-семантичні взаємодії на рівні окремих повідомлень. Екосистема NELA надає великі новинні датасети та інструменти для аналізу мереж поширення контенту, зокрема через графи схожості та копіювання, які співставляють із показниками надійності й упередженості [2].

У роботі Semantic “Echo” of Strategic Communications запропоновано вимірювати відлуння комунікацій у соцмережах за допомогою sentence-embedding і косинусної схожості між повідомленнями організацій та реакціями користувачів [3].

Інший підхід базується на Hawkes-процесах, які моделюють самозбуджувані події: кожна нова подія тимчасово підвищує ймовірність наступних, після чого вплив поступово зменшується [4].

Метод реалізації модульної системи для аналізу інформаційних джерел. Спершу виконується аудит вхідних даних: перевіряються структура, мова, стабільність полів та часові ритми, що забезпечує основу для формування валідних метрик. Далі визначаються функціональні вимоги (лексична подібність, часові індикатори, риторичні класи) та нефункціональні (продуктивність, масштабованість, відмовостійкість, безпека). На цій базі проектується модульна архітектура (API (FastAPI), UI (React), Scheduler (Cron Job), NLP, Preprocessing, Queue (Kafka), Database (MongoDB)), яка взаємодіє з Telegram API через Circuit Breaker як основним джерелом даних (див. рис. 1).

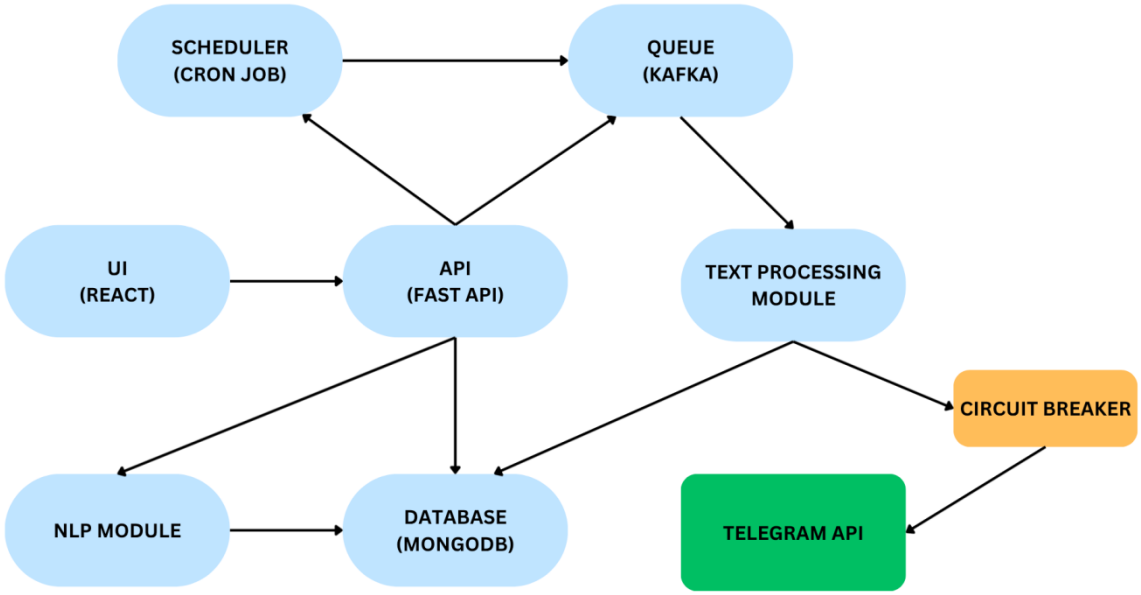

Рис.1. **Загальна архітектура модульної системи**

Ключовим етапом є реалізація й тестування *модулів попередньої* та *семантичної обробки*, побудова карт залежностей між метриками й параметрами, що дозволяє контролювати якість і витрати. Система підтримує аналіз окремих каналів і агреговану аналітику через матриці схожості та TSI, забезпечує автоматичне оновлення даних і надає користувачу статистику та підсумкові висновки про джерела. Це робить її здатною своєчасно виявляти риторику, зв’язки та вплив у динамічних інформаційних потоках.

У системі передбачено кілька ключових модулів. *UI* є основним інструментом взаємодії користувача з аналітикою: він відображає список джерел, статистику, графіки активності та результати NLP-обробки, а також забезпечує роботу з матрицями подібності й графами впливу. *API* виступає зв’язуючим елементом між інтерфейсом і модулями, відповідає за додавання нових джерел, запуск обробки, отримання статистики, а також виконує автентифікацію та обмеження доступу.

Черга завдань реалізована на базі Kafka й забезпечує масштабовану та надійну обробку великих потоків даних, гарантуючи доставку та балансування навантаження [5]. *Модуль*

обробки даних виконує парсинг, очистку, фільтрацію та стандартизацію повідомлень для подальшого NLP-аналізу, зберігаючи їх у структурованому вигляді. Для текстів використовується словник лем та кодування, що дозволяє компактно представляти повідомлення. Нарешті, планувальник автоматизує регулярне оновлення даних, створює завдання для черги й контролює періодичність запуску, що гарантує актуальність аналітики.

Модуль обробки природної мови є ключовим елементом системи: він отримує завдання через API, виконує аналіз текстів (Cosine Similarity, TSI, кластеризація, виявлення ворожої мови тощо) та зберігає результати з параметрами запуску й часовими позначками. Це дає змогу уникати дублювання обчислень і забезпечує масштабовану паралельну обробку.

База даних реалізована на MongoDB, що дозволяє гнучко зберігати різнорідні повідомлення у форматі JSON-документів. Завдяки масштабованості, шардінгу, індексації та агрегаціям MongoDB ефективно підтримує роботу з великими обсягами даних і швидкий доступ до аналітики.

Модуль попередньої обробки (Processing Handler) перетворює сирі повідомлення з Telegram у структуровані дані, готові до аналітики. Він працює асинхронно через Kafka та оптимізований для нестабільних зовнішніх джерел, використовуючи Circuit Breaker, щоб уникати перевантажень у разі збоїв API [6]. Під час завантаження повідомлень система одразу обчислює базові характеристики (довжина, кількість слів, наявність медіа, репости, URL), а також визначає мову. Далі відбувається токенізація тексту, очищення від шумових елементів, переклад російськомовних слів на українську [7], лематизація та формування додаткових метрик (частотність, синтаксичні особливості, унікальність слів).

Після цього дані кодуються у компактний числовий формат, що дозволяє значно економити пам’ять і прискорює доступ. Тести показали, що такий підхід майже вдвічі зменшує обсяг збережених колекцій у порівнянні з рядковим форматом, при цьому обробка десятків тисяч повідомлень займає менше секунди. Завершальним етапом є збереження у MongoDB, де фіксуються як первинні тексти, так і обчислені метрики, що створює основу для подальшого статистичного, семантичного чи графового аналізу.

Модуль обробки природної мови (NLP module) відповідає за проведення повноцінного аналітичного аналізу текстових даних, попередньо підготовлених обробником (Processing Handler). Основна функція модуля полягає у вилученні високорівневої інформації зі збережених повідомлень: від вимірювання лексичної подібності до побудови графів впливу. Для цього використовуються як класичні алгоритми обробки тексту, так і авторські підходи, зокрема Timed Semantic Influence. Кожен аналітичний крок реалізовано як окремий підмодуль, а його результати зберігаються до бази даних, що дозволяє уникнути повторних обчислень і використовувати дані для побудови складніших метрик.

Cosine Similarity Module. Цей підмодуль формує векторне представлення кожного інформаційного джерела на основі його лексичного профілю. Для цього обчислюється частотність використання лем у повідомленнях, після чого створюється багатовимірний вектор *z* (див. формулу 1), який репрезентує тематику, стиль і лексику джерела. На основі побудованих векторів обчислюється матриця косинусної подібності (див. формулу 2), яка дозволяє визначити, наскільки джерела схожі між собою з погляду словникового запасу. Це є першою базовою метрикою, яка демонструє потенційну тематичну або семантичну близькість каналів і надалі використовується у кластеризації або побудові графу зв’язків.

$$z(s) = \frac{1}{n_s} \sum_{k=1}^{n_s} v(x_k) \in \mathbb{R}^d. \tag{1}$$

Тоді для пари джерел s_1, s_2 косинусна подібність дорівнює:

$$cs(s_i, s_j) = \frac{z(s_i)^T z(s_j)}{\|z(s_i)\|_2 \, \|z(s_j)\|_2} \in [-1, 1]\,, \tag{2}$$

де *z* – лексичний вектор для відповідного повідомлення s_i.

Алгоритм *TSI* є інноваційним підходом, що дозволяє виявити часовий вплив одного джерела на інше. Його мета це знайти інформаційні хвилі, тобто подібні повідомлення, які з'являються в різних джерелах у близький проміжок часу. Якщо певне джерело стабільно публікує ключові теми раніше за інші, і його лексика згодом повторюється в інших джерелах - це свідчить про його впливовість. Навіть якщо загальна подібність між каналами низька (за Cosine Similarity), TSI дозволяє виявити прихований зв'язок між джерелами. Результатом роботи є матриця впливу, де показано напрям та інтенсивність інформаційного впливу між парами каналів (див. формулу 3).

$$TSI = \frac{sm(s_i, s_j)}{1 + \alpha \times \Delta t}, \tag{3}$$

де α – коефіцієнт впливу часу, *Δt* – різниця між часом публікацій 2 повідомлень, *sm* – функція косинусної подібності.

TSI визначає вплив між каналами через появу семантично схожих повідомлень у близьких часових вікнах. На відміну від Hawkes, TSI працює не лише з часом, а й зі змістом, що знижує кількість хибних зв'язків і добре інтегрується в NLP-модуль. На основі запропонованого алгоритму формується набір комбінацій параметрів – коефіцієнта впливу часу α, функції подібності між 2 повідомленнями, порогу дозволеної семантичної схожості та діапазону охоплення повідомлень для порівняння. Для кожної комбінації обчислюється матриця впливу між каналами, після чого виконується оптимізація з метою вибору параметрів, що забезпечують найбільшу дисперсію елементів матриці. Висока дисперсія вказує на кращу розрізнювальну здатність моделі, тобто більш чітку ідентифікацію напрямів та інтенсивності інформаційного впливу між джерелами.

Переваги TSI щодо Hawkes.

– Семантично зумовлений вплив: TSI відбирає лише схожі за змістом пари постів у заданому вікні часу, тож зв'язок менш чутливий до шумових співпадінь активності.

– Легка інтеграція у вашу систему: TSI обчислюється на вже підготовлених векторах/ембеддингах.

Недоліки TSI щодо Hawkes.

– Параметри (поріг схожості, форма затухання, ширина вікна) задаються емпірично. Без додаткових статистичних тестів складніше робити формальні висновки про причинність.

– TSI не є повною стохастичною моделлю генерації подій. Це радше оцінювач впливу на основі контенту й часу.

Hostile Language Detection. Цей підмодуль реалізує класифікацію повідомлень за ознаками ворожості на основі спеціально підготовленого датасету. Навчальний набір містить приклади маркованих повідомлень із зазначенням, чи є вони ворожими, і використовується для тренування моделі. Для розпізнавання ворожості використовується модифікована модель BERT, адаптована під специфіку лінгвістичних особливостей зібраного корпусу ворожих повідомлень. Після навчання модель застосовується до всієї вибірки, і кожному повідомленню присвоюється коефіцієнт ворожості. На основі цих оцінок обчислюється середній рівень ворожості для кожного джерела, що стає ще однією метрикою для подальшого аналізу [8]. Це особливо корисно при виявленні джерел з агресивною риторикою або маніпулятивними повідомленнями, і може вказувати на координацію в межах групи джерел. На основі завчасно невеликої кількості помічених каналів як ворожих і неворожих здійснюється класифікація джерел, під час якої модель оцінює рівень ворожості кожного повідомлення. Для оцінки якості класифікації застосовуються стандартні метрики машинного навчання – accuracy, recall, precision та F1-score.

Clusterization. Після того як для всіх джерел сформовано векторні представлення (лексичний профіль, ворожість, TSI-вплив), виконується кластеризація джерел. Мета цього підмодуля об'єднати джерела з дуже подібними риторичними або тематичними характеристиками в групи. Застосовуються стандартні алгоритми кластеризації, зокрема DBSCAN [9] або KMeans [10], які дають змогу працювати з високовимірними просторами.

Кластери, отримані в результаті, ідентифікують інформаційні коаліції, групи джерел, які діють узгоджено або розповсюджують схожі наративи.

Source Graph. Фінальним етапом є побудова графа зв'язків між джерелами. Цей підмодуль об'єднує всі раніше обчислені метрики: косинусну подібність, TSI, рівень ворожості, статистичну активність, для формування орієнтованого графа, де вузлами є інформаційні джерела, а ребрами - наявність і напрямок впливу. Вага ребра відображає силу взаємозв'язку: вона може базуватись на TSI, ворожій риториці чи тематичній близькості. Граф дає змогу візуалізувати структуру інформаційного поля, знайти ізольовані або надто впливові джерела, а також виявити потенційні центри координації.

Усі результати аналізу як проміжні (вектори, метрики, оцінки), так і фінальні (матриці, кластери, граф) зберігаються у базі даних Mongo [11]. Це дозволяє повторно їх використовувати для візуалізацій, нових запитів, побудови графіків або порівнянь між часовими зрізами, не повторюючи обчислення.

Запропонована система реалізує повний цикл збору, обробки й аналізу текстових повідомлень з Telegram-каналів та інших джерел. Архітектура побудована модульно й відмовостійко, підтримує масштабування, автоматичні оновлення та сповіщення. Центральний NLP-модуль об'єднує кілька підходів: Cosine Similarity, нову метрику TSI, класифікацію риторики та кластеризацію каналів. Саме TSI поєднує часову близькість і семантичну подібність, дозволяючи будувати матрицю впливів і виявляти приховані зв'язки навіть при низькій лексичній схожості.

Практична цінність полягає у швидкому отриманні висновків про джерела: тип, теми, динаміку зв'язків і позицій у графі. Поєднання TSI з іншими метриками допомагає виявляти координацію та інформаційні хвилі. Завдяки Kafka й асинхронній обробці система стабільно працює з великими потоками даних, легко доповнюється новими метриками й придатна для OSINT, редакційних чи дослідницьких завдань.

СПИСОК ЛІТЕРАТУРИ

1. Ye J., Skiena S. "MediaRank: Computational ranking of online news sources". *arXiv*. 2019. DOI: https://doi.org/10.48550/arXiv.1903.07581.

2. Horne B. D., Dron W., Khedr S., Adalı S. "Assessing the news landscape: A multi-module toolkit for evaluating the credibility of news". *Companion Proceedings of The Web Conference (WWW'18). ACM*. Lyon, France. 2018. p. 235–238. DOI: https://doi.org/10.1145/3184558.3186987.

3. Cann T. J. B., Dennes B., Coan T., O'Neill S., Williams H. T. P. "Using semantic similarity and text embedding to measure the social media echo of strategic communications". *arXiv*. 2023. DOI: https://doi.org/10.48550/arXiv.2303.16694.

4. Worrall J., Browning R., Wu P., Mengersen K. "Fifty years later: New directions in Hawkes processes". *SORT*. 2022; 46 (1): 3–38. DOI: https://doi.org/10.2436/20.8080.02.116.

5. Valdivia J. A., Lora-Gonzalez A., Limon X., Verdin K. C. "Patterns related to microservice architecture: A multivocal literature review". *Programming and Computer Software*. 2024; 46 (8): 594–608. DOI: https://doi.org/10.1134/S0361768820080253.

6. Serbout S., El Malki A., Pautasso C., Zdun U. "API rate limit adoption – A pattern collection". P*roceedings of the 28th European Conference on Pattern Languages of Programs (EuroPLoP'23). ACM*. Irsee, Germany. 2023. p. 1–20. DOI: https://doi.org/10.1145/3628034.3628039.

7. Pellicano N., Parisi L., Ragusa A. "FastSpell: An efficient method for multilingual language identification and spelling correction". *arXiv*. 2024. – URL: https://arxiv.org/abs/2404.07567.

8. Jahan M. S., Oussalah M. "A systematic review of hate speech automatic detection using natural language processing". *Neurocomputing*. 2023; 546: 126232. DOI: https://doi.org/10.1016/j.neucom.2023.126232.

9. Celebi M. E., Aslandogan Y. A. "A survey of density based clustering algorithms". *Knowledge-Based Systems*. 2020; 212: 106598. DOI: https://doi.org/10.1007/s11704-019-9059-3.

10. Singh J., Singh D. “A comprehensive review of clustering techniques in artificial intelligence for knowledge discovery: Taxonomy, challenges, applications and future prospects”. *Advanced Engineering Informatics*. 2024; 62: 102799. DOI: https://doi.org/10.1016/j.aei.2024.102799.

11. Rathore M., Bagui S. “MongoDB: Meeting the dynamic needs of modern applications”. *Encyclopedia*. 2024; 4: 1433–1453. DOI: https://doi.org/10.3390/encyclopedia4040093.

DOI: https://doi.org/10.15276/ict.02.2025.14
UDC 004.93

Method for implementing a complex modular system for analyzing information sources

Dmytro I. Uhrin[1)]
Doctor of Engineering Sciences, Professor of the Department of Computer Sciences
ORCID: https://orcid.org/0000-0003-4858-4511; d.ugryn@chnu.edu.ua. Scopus Author ID: 57163746300
Yuriy A. Ushenko[1)]
Doctor of Physics and Mathematics, Professor of the Department of Computer Sciences
ORCID: https://orcid.org/0000-0003-1767-1882; y.ushenko@chnu.edu.ua. Scopus Author ID: 6701840218
Artem D. Kalancha[1)]
PhD Student of the Department of Computer Systems Software
ORCID: https://orcid.org/0009-0004-1451-7470; kalancha.artem@chnu.edu.ua
[1)] Yuriy Fedkovich Chernivtsi National University, 2, Kotsyubinsky Str. Chernivtsi, 58012, Ukraine

ABSTRACT

The article presents a method for building a modular information system for collecting, processing and analyzing text messages from open sources, in particular Telegram channels. The system combines procedures for cleaning and normalizing text with the calculation of metrics: Cosine Similarity, time-semantic influence (TSI), hostile rhetoric classification, clustering and building a graph of relationships. To ensure stability, an asynchronous processor with a message queue and a Circuit Breaker template are used, and the results are stored in MongoDB. TSI allows you to detect cross-channel influence even with low lexical similarity, while the hostile language module analyzes rhetoric at the message and source levels. The system generates automatic conclusions for each channel and supports visualization of results, which increases the speed and convenience of analytics. The developed solution has applied value for OSINT, information security and research tasks.

Keywords: Timed Semantic Influence; cosine similarity; hostile speech; source clustering; information influence graph; automated data collection