ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Методика автоматизації парсингу фінансових новин на основі їх семантичного аналізу»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Олексій МАРТИНЕНКО
(підпис)

Виконав: здобувач вищої освіти групи ПД-62

Олексій МАРТИНЕНКО

Керівник: Максим КУКЛІНСЬКИЙ
канд. техн. наук

Рецензент: _____

Київ 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інженерії програмного забезпечення
_____ Ірина ЗАМРІЙ
«_____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Мартиненку Олексію Володимировичу

1. Тема кваліфікаційної роботи: «Методика автоматизації парсингу фінансових новин на основі їх семантичного аналізу»

керівник кваліфікаційної роботи Максим КУКЛІНСЬКИЙ, канд. техн. наук, доц.,

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» жовтня 2025 р. № 467.

2. Строк подання кваліфікаційної роботи «19» грудня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, відкриті джерела фінансових новин; готові моделі обробки природної мови для семантичного аналізу; програмні інструменти та бібліотеки для парсингу, попередньої обробки тексту й аналізу семантичної схожості; вимоги до точності тематичної класифікації та аналізу сентименту фінансових новин

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Дослідження методів автоматизованого парсингу фінансових новин з відкритих інформаційних джерел та аналіз форматів представлення новинних даних.
2. Аналіз сучасних підходів до семантичного аналізу текстів і використання готових NLP-моделей для обробки фінансових новин.
3. Розробка та експериментальна перевірка методики автоматизації парсингу і семантичного аналізу фінансових новин з оцінкою її ефективності та обмежень.

5. Перелік ілюстративного матеріалу: *презентація*

1. Мета, об’єкт та предмет дослідження.
2. Актуальність роботи.
3. Схема роботи методу.
4. Схема роботи методу (2).
5. Етапи семантичного аналізу.
6. Етапи семантичного аналізу (2).
7. Вебзастосунок аналізу фінансових новин.

6. Дата видачі завдання«31» жовтня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	31.10.25–06.11.2025	
2	Вивчення матеріалів для парсингу та семантичного аналізу	07.11.25–11.11.2025	
3	Дослідження методів аналізу фінансових новин	12.11.25–17.11.2025	
4	Аналіз особливостей семантичного аналізу фінансових новин	18.11.25–22.11.2025	
5	Дослідження технологій парсингу та методів обробки природньої мови	23.11.25–28.11.2025	
6	Застосування методів обробки природньої мови для аналізу фінансових новин	29.11.25–06.12.2025	
7	Оформлення роботи: вступ, висновки, реферат	07.12.25–12.12.2025	
8	Розробка демонстраційних матеріалів	13.12.25–16.12.2025	

9	Попередній захист роботи	17.12.25–19.12.2025	

Здобувач вищої освіти _____ Олексій МАРТИНЕНКО
(підпис)

Керівник кваліфікаційної роботи _____ Максим КУКЛІНСЬКИЙ
(підпис)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 72 стор., 1 табл., 9 рис., 30 джерел.

Мета роботи – підвищення ефективності автоматизованого аналізу фінансових новин шляхом застосування методів семантичного аналізу тексту та готових моделей обробки природної мови для витягу ключових тем і інформаційних характеристик новинних повідомлень.

Об'єкт дослідження – процес автоматизованої обробки фінансових новин як неструктурованих текстових даних з відкритих інформаційних джерел.

Предмет дослідження – методика автоматизації парсингу фінансових новин на основі їх семантичного аналізу з використанням сучасних NLP-інструментів та попередньо навчених мовних моделей..

У роботі використано методи автоматизованого парсингу веб-ресурсів і новинних стрічок, попередньої обробки текстів, семантичного аналізу, векторного представлення текстових даних, тематичної класифікації та аналізу тональності. Для реалізації дослідження застосовано готові моделі обробки природної мови, що дозволяють враховувати контекст і семантичні зв'язки у фінансових текстах без розробки власних моделей машинного навчання.

Проведено аналіз сучасних підходів до автоматизованого аналізу фінансових новин, розглянуто існуючі програмні рішення та інструменти, визначено їх переваги й обмеження з точки зору гнучкості та відтворюваності результатів. На основі проведеного аналізу розроблено методику автоматизації парсингу та семантичного аналізу фінансових новин, що поєднує збір даних з відкритих джерел, їх очищення та подальший змістовний аналіз.

У ході роботи проведено експериментальну перевірку запропонованої методики на вибірці фінансових новин, оцінено її придатність для тематичної

класифікації та аналізу інформаційного фону. Отримані результати підтверджують доцільність використання готових NLP-моделей для побудови інформаційно-аналітичних систем підтримки прийняття рішень у фінансовій сфері

КЛЮЧОВІ СЛОВА: ФІНАНСОВІ НОВИНИ, СЕМАНТИЧНИЙ АНАЛІЗ, ОБРОБКА ПРИРОДНОЇ МОВИ, NLP, АВТОМАТИЗОВАНИЙ ПАРСИНГ, АНАЛІЗ ТЕКСТУ, ІНФОРМАЦІЙНІ СИСТЕМИ.

ABSTRACT

Text part of the master's qualification work: 72 pages, 9 pictures, 1 table, 30 sources.

The purpose of the work is to improve the efficiency of automated analysis of financial news by applying semantic text analysis methods and pre-trained natural language processing models for extracting key topics and informational characteristics from news content.

Object of research – the process of automated processing of financial news as unstructured textual data obtained from open information sources.

Subject of research – the methodology for automating the parsing of financial news based on semantic analysis using modern NLP tools and pre-trained language models.

Summary of the work:

The master’s qualification work is devoted to the development and investigation of a methodology for automated parsing and semantic analysis of financial news. The study analyzes current approaches to processing unstructured financial text data and reviews existing software solutions used for news aggregation and analysis. The work focuses on the practical application of pre-trained natural language processing models without developing custom machine learning models, which allows ensuring reproducibility and reducing computational complexity.

The proposed methodology includes automated collection of financial news from open sources, text preprocessing, semantic representation of news content, and further thematic and sentiment analysis. Experimental evaluation was conducted on a dataset of financial news, demonstrating the applicability of the proposed approach for identifying key topics and assessing the informational background of financial markets. The results confirm that the use of pre-trained NLP models enables effective and scalable analysis of financial news and can serve as a foundation for information-analytical decision support systems in the financial domain.

KEYWORDS FINANCIAL NEWS, SEMANTIC ANALYSIS, NATURAL LANGUAGE PROCESSING, NLP, AUTOMATED PARSING, TEXT ANALYSIS, INFORMATION SYSTEMS

.

ЗМІСТ

ВСТУП

У сучасних умовах стрімкого розвитку інформаційних технологій та цифровізації суспільства фінансова інформація відіграє ключову роль у процесах прийняття управлінських, інвестиційних та стратегічних рішень. Щоденно у відкритих джерелах, таких як новинні портали, аналітичні платформи, фінансові блоги та офіційні ресурси компаній, публікуються тисячі фінансових новин, які містять дані про стан ринків, діяльність корпорацій, макроекономічні показники, валютні коливання та регуляторні зміни. Обсяг такої інформації постійно зростає, що унеможливлює її ефективний аналіз традиційними ручними методами.

Фінансові новини характеризуються високою динамічністю, тематичною різноманітністю та значною кількістю неструктурованих текстових даних. При цьому своєчасне виявлення ключових тем, подій та загального інформаційного тону новин є критично важливим для фінансових аналітиків, трейдерів, економістів, дослідників та автоматизованих систем підтримки прийняття рішень. У зв’язку з цим актуальним є завдання створення методик та інструментів, здатних автоматизувати процес збору, обробки та аналізу фінансових текстових даних з мінімальним втручанням людини.

Особливого значення набувають методи обробки природної мови (Natural Language Processing, NLP), які дозволяють здійснювати попередню обробку текстів, лематизацію, виділення ключових слів, іменованих сутностей, тематичну класифікацію та аналіз сентименту. Поєднання NLP-підходів із методами семантичного аналізу та векторного подання тексту створює передумови для більш глибокого розуміння змісту фінансових новин, а не лише їх поверхневої лексичної структури.

Семантичний аналіз тексту дає змогу враховувати контекст, смислові зв’язки між словами та тематичну спрямованість повідомлень. Застосування векторних моделей представлення тексту, зокрема методів на основі частотних характеристик та семантичних ембеддингів, дозволяє формалізувати текстову інформацію у вигляді числових векторів, придатних для подальшого автоматизованого аналізу,

порівняння та класифікації. Це є особливо важливим у фінансовій сфері, де навіть незначні зміни формулювань можуть суттєво впливати на інтерпретацію новин.

Актуальність даної магістерської роботи зумовлена необхідністю підвищення ефективності аналізу фінансових новин в умовах інформаційного перевантаження, а також потребою у створенні універсальної методики автоматизації парсингу та семантичного аналізу текстів, орієнтованої на фінансову тематику. Автоматизація таких процесів дозволяє скоротити час обробки інформації, зменшити вплив людського фактору, підвищити об'єктивність результатів аналізу та забезпечити масштабованість системи при роботі з великими обсягами даних.

Мета роботи – розробка та дослідження методики автоматизації парсингу фінансових новин на основі їх семантичного аналізу, яка забезпечує ефективний збір, обробку та інтерпретацію текстових даних з відкритих інформаційних джерел.

Об'єкт дослідження – процес автоматизованої обробки та аналізу фінансових текстових даних, що надходять з новинних інформаційних ресурсів.

Предмет дослідження – методи та алгоритми автоматизації парсингу фінансових новин, а також засоби семантичного аналізу текстів, включаючи методи попередньої обробки, векторного представлення та тематичного аналізу текстової інформації.

Для досягнення поставленої мети у роботі передбачається вирішення таких основних завдань:

1. Провести аналіз сучасних підходів до автоматизованого збору та парсингу фінансових новин з відкритих джерел, а також огляд існуючих методів обробки неструктурованих текстових даних.
2. Дослідити методи семантичного аналізу тексту та їх застосування до фінансових новин, зокрема підходи до виділення ключових тем, сутностей та загального інформаційного тону.
3. Розробити методику автоматизації парсингу фінансових новин, що включає етапи збору даних, попередньої обробки тексту та семантичного аналізу.

4. Реалізувати програмний прототип системи, який демонструє застосування запропонованої методики на практиці.
5. Провести експериментальну перевірку ефективності розробленої методики та проаналізувати отримані результати, визначивши її переваги, обмеження та можливі напрями подальшого вдосконалення.

Таким чином, результати магістерської роботи спрямовані на вирішення актуальної науково-практичної задачі автоматизованого аналізу фінансових новин і можуть бути використані як у дослідницькій діяльності, так і при створенні прикладних інформаційно-аналітичних систем у фінансовій сфері.

1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1. Аналіз фінансових новин як джерела даних

У сучасній фінансово-економічній системі новинна інформація є одним із ключових джерел даних для аналізу стану ринків, оцінки ризиків та формування прогнозів. Фінансові новини відображають як макроекономічні процеси (інфляція, процентні ставки, монетарна політика, геополітичні події), так і мікроекономічні фактори, пов’язані з діяльністю окремих компаній, галузей та фінансових інструментів. Саме через новинні повідомлення учасники ринку оперативно отримують інформацію, яка безпосередньо впливає на динаміку цін, інвестиційні очікування та поведінку ринку.

Особливістю фінансових новин є їх текстова, неструктурована природа, а також високий рівень термінологічної насиченості та контекстної залежності. Одна й та сама подія може бути описана різними джерелами з відмінними акцентами, що ускладнює їх порівняння та узагальнення. Крім того, фінансові тексти часто містять приховані смислові сигнали, наприклад, зміну тону повідомлення або використання специфічних формулювань, які можуть свідчити про позитивні або негативні очікування ринку.

Зі зростанням кількості онлайн-ресурсів, новинних стрічок та агрегаторів інформації виникає проблема інформаційного перевантаження. Людський аналіз великих обсягів новин стає неефективним і суб’єктивним, що зумовлює необхідність автоматизації процесів збору, фільтрації та аналізу фінансових текстових даних. У цьому контексті особливої актуальності набувають методи автоматизованого парсингу новинних джерел, а також семантичного аналізу, які дозволяють перейти від простого збору текстів до глибшого розуміння їх змісту.

Автоматизований аналіз фінансових новин використовується для вирішення широкого кола задач: тематичної класифікації новин, виявлення ключових подій, оцінки інформаційного фону навколо компаній або ринків, а також аналізу

сентименту. Наукові дослідження підтверджують, що текстові сигнали з фінансових новин можуть корелювати з ринковою динамікою та бути корисними для побудови аналітичних і прогнозних моделей [9]. Таким чином, фінансові новини є цінним, але складним джерелом даних, що потребує спеціалізованих методик обробки.

1.1.1. Роль семантичного аналізу у фінансових новинах

Семантичний аналіз є одним із ключових напрямів обробки природної мови, який спрямований на виявлення смислового наповнення тексту, а не лише його поверхневої лексичної структури. На відміну від традиційних частотних підходів, семантичний аналіз дозволяє враховувати контекст, взаємозв'язки між словами та загальну інтерпретацію текстового повідомлення. Для фінансових новин це має принципове значення, оскільки економічні тексти часто містять непрямі оцінки, припущення та умовні формулювання.

У фінансовій сфері семантичний аналіз широко застосовується для задач визначення тональності (sentiment analysis), тематичної класифікації та виявлення ключових концептів. Наприклад, одна і та сама лексема може мати різне смислове навантаження залежно від контексту, що особливо характерно для фінансової термінології. Саме тому використання доменно-орієнтованих підходів та спеціалізованих мовних моделей є більш ефективним порівняно з універсальними методами аналізу тексту [5].

Сучасні методи семантичного аналізу фінансових новин базуються на векторних поданнях тексту, які формуються за допомогою нейронних мереж трансформерного типу. Такі підходи дозволяють відображати текстові одиниці у багатовимірному семантичному просторі, де близькість векторів відповідає смисловій подібності текстів або окремих речень [1]. Це відкриває можливості для автоматичного групування новин за темами, порівняння повідомлень з різних джерел та виявлення дублюючого або близького за змістом контенту.

Водночас семантичний аналіз фінансових новин має низку обмежень, пов'язаних зі складністю економічної мови, динамічністю термінології та впливом

зовнішнього контексту. Результати такого аналізу залежать від якості попередньої обробки тексту, вибору моделей та навчальних даних. Тому семантичний аналіз доцільно розглядати як інструмент підтримки аналітичних рішень, який доповнює, але не замінює експертну оцінку.

Таким чином, застосування семантичного аналізу у фінансових новинах є важливим елементом сучасних інформаційно-аналітичних систем, що дозволяє підвищити глибину та об’єктивність аналізу фінансової інформації та створює основу для автоматизованих методик обробки новинних даних.

1.1.2. Використання готових NLP-моделей у фінансовому аналізі

Сучасний розвиток методів обробки природної мови значною мірою пов’язаний із появою глибоких нейронних архітектур, насамперед трансформерних моделей. Такі моделі дозволяють ефективно враховувати контекст, семантичні зв’язки та залежності між словами у тексті, що є критично важливим для аналізу фінансових новин. Водночас розробка та навчання власних моделей глибокого навчання потребує значних обчислювальних ресурсів, великих розмічених датасетів і суттєвих часових витрат.

З огляду на це, у практичних інформаційно-аналітичних системах все частіше застосовуються вже готові, попередньо навчені NLP-моделі, які можуть бути безпосередньо використані для розв’язання прикладних задач. Такі моделі навчаються на великих корпусах текстів і здатні забезпечувати високу якість результатів без необхідності повторного навчання з нуля. Прикладами є трансформерні архітектури, що формують семантичні векторні подання слів, речень або цілих документів [1].

У фінансовій галузі особливої популярності набули попередньо навчені мовні моделі, адаптовані до економічної та фінансової тематики. Вони дозволяють ефективніше аналізувати спеціалізовану термінологію, звітність, новинні тексти та аналітичні огляди. Такі моделі застосовуються для задач аналізу тональності фінансових новин, виявлення ключових фраз, класифікації текстів за темами та оцінки семантичної схожості між публікаціями [5].

Використання готових NLP-моделей у рамках даної роботи дозволяє зосередитись на побудові методики автоматизації парсингу та семантичного аналізу, не ускладнюючи систему етапами навчання власних моделей. Це підвищує відтворюваність результатів, спрощує інтеграцію рішень у веб-сервіси та забезпечує стабільну якість аналізу при роботі з фінансовими новинами з різних джерел.

1.1.3. Автоматизований аналіз текстових даних у фінансових системах

Автоматизований аналіз текстових даних є важливою складовою сучасних фінансових інформаційних систем. Його основною метою є перетворення неструктурованих текстів, зокрема фінансових новин, у формалізований вигляд, придатний для подальшого аналізу, агрегування та візуалізації. На відміну від алгоритмічної торгівлі, яка безпосередньо спрямована на виконання торгових операцій, автоматизований текстовий аналіз виконує допоміжну, але критично важливу аналітичну функцію.

Такі системи використовуються для моніторингу інформаційного фону навколо фінансових ринків, окремих компаній або галузей. Вони дозволяють у реальному або квазі-реальному часі відстежувати появу нових новин, визначати їх тематику, виявляти потенційно важливі події та оцінювати загальний тон інформаційного потоку. Це особливо актуально в умовах високої динаміки фінансових ринків, коли затримка в отриманні або інтерпретації інформації може призводити до значних втрат.

Автоматизований аналіз фінансових новин зазвичай включає кілька послідовних етапів: збір даних з відкритих джерел, очищення та нормалізацію тексту, семантичний аналіз і подальшу інтерпретацію результатів. На кожному з цих етапів використовуються спеціалізовані програмні інструменти та бібліотеки, що дозволяють масштабувати систему та обробляти великі обсяги текстової інформації без втрати продуктивності.

Важливою перевагою таких систем є зменшення впливу людського фактору та підвищення об’єктивності аналізу. Водночас результати автоматизованого

текстового аналізу не слід розглядати як остаточну основу для прийняття фінансових рішень. Вони виконують роль інформаційної підтримки аналітиків і можуть слугувати додатковим джерелом даних у комплексних фінансово-аналітичних моделях. Саме в такому контексті автоматизований семантичний аналіз фінансових новин розглядається у даній магістерській роботі як інструмент підвищення ефективності інформаційної обробки, а не як самостійна система прогнозування.

1.2. Аналіз програмного забезпечення

На сьогоднішній день на ринку інформаційно-аналітичних рішень представлено значну кількість програмних платформ і сервісів, орієнтованих на аналіз фінансових даних, у тому числі новинної інформації. Такі системи використовуються для моніторингу ринків, аналізу інформаційного фону, підтримки інвестиційних рішень та аналітичних досліджень. Водночас більшість із них є комерційними продуктами з обмеженою гнучкістю адаптації або орієнтовані на вузьке коло задач. У даному підрозділі розглянуто п’ять найбільш поширених аналогів, які частково вирішують задачі, подібні до тематики даної магістерської роботи.

1.2.1. Bloomberg Terminal

Bloomberg Terminal є одним із найбільш відомих і потужних інструментів у фінансовій індустрії. Платформа надає доступ до величезних обсягів фінансових даних, включаючи ринкові котирування, аналітичні звіти, макроекономічні показники та фінансові новини в реальному часі. Новинний модуль Bloomberg агрегує інформацію з великої кількості перевірених джерел та забезпечує швидкий доступ до актуальних подій.

Аналітичні можливості платформи включають базові інструменти фільтрації новин за компаніями, секторами або регіонами, а також пошук за ключовими словами. Проте внутрішні алгоритми аналізу тексту є закритими, а користувач не

має можливості налаштовувати або адаптувати семантичні моделі під власні потреби. Крім того, висока вартість ліцензії значно обмежує доступність платформи для навчальних та дослідницьких цілей.

Рис. 1.1. Інтерфейс Bloomberg Terminal

1.2.2. Refinitiv Eikon

Refinitiv Eikon є комплексною фінансово-аналітичною платформою, яка поєднує ринкові дані, фінансові звіти, аналітику та новинні стрічки. Система надає користувачам доступ до глобальних фінансових новин, включаючи корпоративні оголошення, економічні події та коментарі аналітиків.

Платформа підтримує інструменти аналізу текстових даних, зокрема пошук за ключовими словами та категоріями, а також базову оцінку новинного фону. Водночас семантичний аналіз у Refinitiv Eikon реалізований у вигляді готових функцій без можливості глибокої кастомізації. Система орієнтована передусім на кінцевих користувачів фінансових ринків, а не на розробників або дослідників, що ускладнює інтеграцію власних методик NLP.

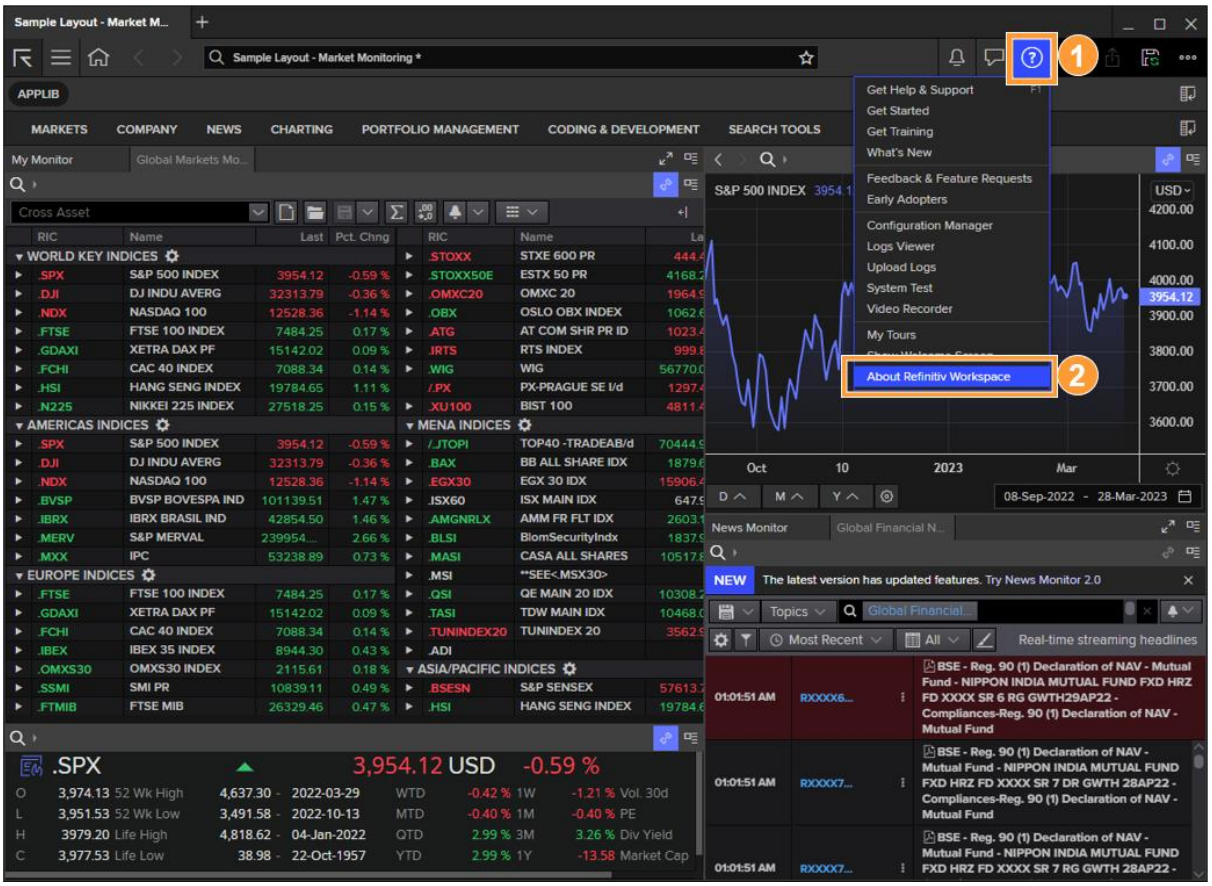


Рис. 1.2. Інтерфейс платформи Refinitiv Eikon

1.2.3. RavenPack

RavenPack є спеціалізованою платформою для аналізу фінансових новин і подій, орієнтованою на використання методів обробки природної мови та машинного навчання. Основною особливістю системи є перетворення новинних повідомлень у структуровані сигнали, які можуть бути використані в аналітичних або торгових стратегіях.

Платформа здійснює автоматичне визначення подій, компаній, тональності та рівня значущості новин. Однак більшість алгоритмів є пропрієтарними, а користувач отримує лише готові індикатори без доступу до внутрішньої логіки семантичного аналізу. Це обмежує можливість наукового дослідження або адаптації під конкретні завдання, зокрема експерименти з різними моделями NLP.

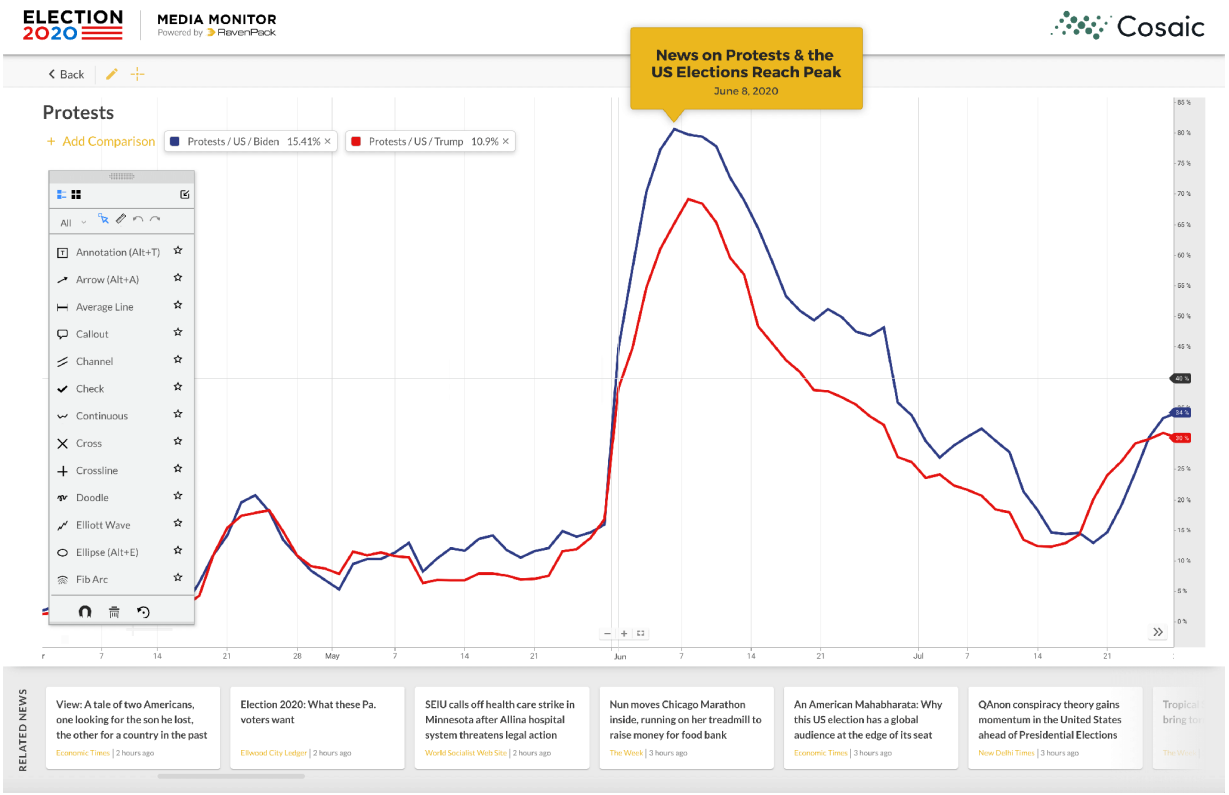


Рис. 1.3. Інтерфейс платформи RavenPack

1.2.4. AlphaSense

AlphaSense є аналітичною платформою, орієнтованою на пошук та аналіз текстової інформації з фінансових документів, звітів компаній, транскриптів конференц-дзвінків і новин. Система активно використовує NLP для покращення пошукових можливостей та контекстного аналізу.

Основною перевагою AlphaSense є потужний семантичний пошук, який дозволяє знаходити релевантну інформацію навіть за непрямими формулюваннями. Водночас платформа фокусується переважно на аналітичному пошуку, а не на повноцінному автоматизованому семантичному аналізі новинних потоків. Можливості автоматичного парсингу зовнішніх джерел та побудови власних аналітичних конвеєрів є обмеженими.

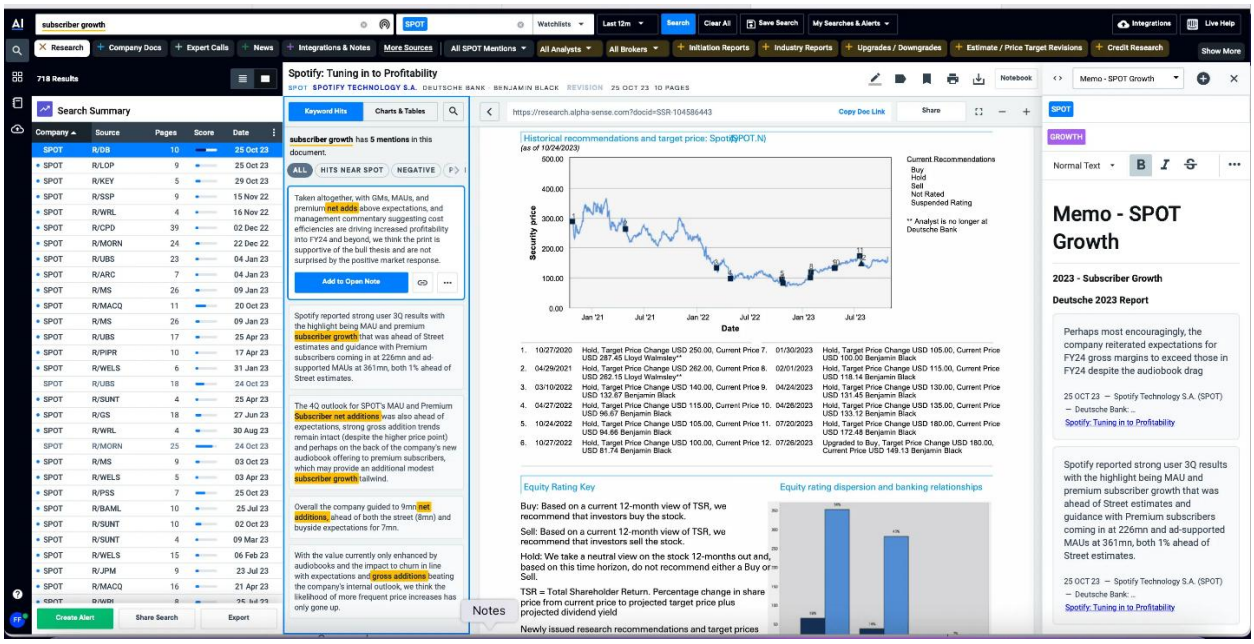


Рис. 1.4. Інтерфейс платформи AlphaSense

1.2.5. Google Cloud Natural Language API (у фінансових сценаріях)

Google Cloud Natural Language API є універсальним сервісом для аналізу текстових даних, який підтримує визначення тональності, класифікацію текстів та витяг іменованих сутностей. Хоча платформа не є спеціалізованим фінансовим продуктом, вона часто використовується для аналізу фінансових новин у прикладних рішеннях.

Перевагою даного підходу є висока масштабованість, стабільність та простота інтеграції через API. Разом з тим, універсальний характер сервісу знижує точність аналізу фінансової термінології, оскільки модель не адаптована під специфіку економічних текстів. Крім того, користувач не має контролю над архітектурою моделей і не може комбінувати різні підходи семантичного аналізу в межах одного конвеєра.

Зведені результати аналізу характеристик розглянутих додатків наведено у таблиці 1.1.

Таблиця 1.1

Зведені результати аналізу характеристик додатків

Назва	Автоматизація аналізу	Гнучкість та налаштовуваність аналізу	Доступність для досліджень та інтеграції
Bloomberg Terminal	Висока: повна агрегація та швидка обробка фінансових новин у реальному часі	Низька: закриті алгоритми, відсутність доступу до моделей	Низька: висока вартість, обмежене використання в наукових дослідженнях
Refinitiv Eikon	Висока: інтегровані новинні стрічки та фільтрація за фінансовими ознаками	Низька: фіксовані аналітичні інструменти без кастомізації	Середня: обмежений доступ для розробників, комерційна ліцензія
RavenPack	Висока: автоматичне виділення подій, сутностей і тональності	Середня: доступні готові сигнали, але без контролю внутрішніх моделей	Низька: пропрієтарне рішення, орієнтація на бізнес-користувачів
AlphaSense	Середня: фокус на аналізі документів і новин, а не потоковому парсингу	Середня: сильний семантичний пошук, обмежений NLP-конвеєр	Середня: доступна аналітика, але закрита архітектура

Продовження таблиці 1.1

Зведені результати аналізу характеристик додатків

Назва	Автоматизація аналізу	Гнучкість та налаштовуваність аналізу	Доступність для досліджень та інтеграції
Google Cloud Natural Language API	Середня: універсальний аналіз текстів, не спеціалізований на фінансах	Низька: відсутня доменна адаптація та контроль моделей	Висока: проста інтеграція через API, масштабованість

Порівняльний аналіз показує, що існуючі аналоги або забезпечують високу автоматизацію за рахунок закритих комерційних рішень, або пропонують універсальні NLP-інструменти без фінансової спеціалізації. Це підтверджує доцільність розробки методики автоматизації парсингу фінансових новин на основі готових NLP-моделей із можливістю гнучкого налаштування, розширення та використання у науково-дослідницьких і прикладних системах.

1.3. Аналіз інформаційних потоків фінансових новин

Фінансові новини формують окремий клас інформаційних потоків, які мають безпосередній і часто негайний вплив на поведінку учасників ринку, очікування інвесторів та загальний інформаційний фон. На відміну від структурованих фінансових показників, таких як ціни активів, обсяги торгів або макроекономічні індикатори, новинні дані характеризуються високою динамічністю, неструктурованістю та контекстною залежністю. Одна й та сама подія може інтерпретуватися по-різному залежно від джерела, формулювань і часу публікації, що значно ускладнює їх системний аналіз та узагальнення.

Інформаційні потоки фінансових новин мають виражену часову структуру. Новини з’являються нерівномірно, формуючи інформаційні сплески у відповідь на важливі події, такі як публікація фінансової звітності, рішення центральних банків або геополітичні події. Аналіз часової динаміки новин дозволяє виявляти періоди підвищеної інформаційної активності, оцінювати швидкість поширення інформації та її вплив на ринкові очікування. Для таких задач важливою є коректна фіксація часових міток та синхронізація новин з іншими джерелами фінансових даних.

Не менш важливою є тематична складова інформаційних потоків. Фінансові новини охоплюють широкий спектр тем — від корпоративних подій і фінансових результатів компаній до макроекономічної політики, регуляторних змін і глобальних ризиків. Тематична структура новинного потоку є динамічною та змінюється залежно від ринкової ситуації. Аналіз тематичних зсувів дозволяє виявляти домінуючі напрями інформаційного поля та відстежувати появу нових тем, що можуть мати значення для подальших аналітичних досліджень.

Ключовою особливістю фінансових новин є їх семантична насиченість. Тексти містять не лише фактичну інформацію, але й оціночні судження, прогнози та непрямі сигнали, які впливають на сприйняття подій ринковими учасниками. Саме тому аналіз інформаційних потоків неможливий без урахування семантичного контексту, у якому подається інформація. Методи обробки природної мови дозволяють формалізувати текстові дані, виділяти ключові смислові елементи та переводити новини у форму, придатну для подальшої аналітичної обробки [9].

Автоматизований аналіз новинних потоків забезпечує можливість безперервного моніторингу інформаційного середовища. Такі системи дозволяють оперативно виявляти значущі події, відстежувати зміни тематики та аналізувати еволюцію інформаційного тону в часі. У фінансовій сфері це створює передумови для побудови інформаційно-аналітичних систем, які підтримують прийняття рішень, забезпечують раннє виявлення потенційних ризиків та сприяють глибшому розумінню інформаційного фону ринку.

Таким чином, аналіз інформаційних потоків фінансових новин є складною багатовимірною задачею, що поєднує часовий, тематичний та семантичний аспекти. Ефективне вирішення цієї задачі можливе лише за умови використання автоматизованих методів обробки текстових даних, які дозволяють систематизувати великі обсяги новинної інформації та інтегрувати її у сучасні фінансово-аналітичні системи.

1.3.1. Тематична структура фінансових новин

Однією з ключових характеристик фінансових новин є їх виражена тематична різноманітність, що відображає складну та багатовимірну природу фінансових ринків. Новинні повідомлення можуть стосуватися макроекономічних показників, таких як інфляція, процентні ставки або рівень безробіття, діяльності окремих компаній і галузей, змін у регуляторному та податковому середовищі, публікації фінансової звітності, процесів злиття і поглинання, а також геополітичних подій, що опосередковано впливають на фінансові ринки. Така різноманітність тематик призводить до формування складного інформаційного поля, в якому одночасно співіснують кілька домінуючих і другорядних тем.

Автоматичне визначення тематичної належності фінансових новин є важливим етапом аналізу інформаційних потоків, оскільки воно дозволяє структурувати новинний контент, зменшити інформаційний шум та спростити подальший аналітичний процес. Без тематичної сегментації новинні потоки сприймаються як суцільний масив текстів, що ускладнює виявлення ключових тенденцій і подій. Тематична класифікація, навпаки, дає змогу групувати новини за смисловою ознакою та фокусувати увагу на релевантних аспектах інформаційного середовища.

Для виявлення тематичної структури фінансових новин традиційно застосовуються методи тематичного моделювання та класифікації текстів. Класичні підходи ґрунтуються на статистичному аналізі розподілу термінів у корпусі документів. Зокрема, модель латентного розміщення Діріхле дозволяє автоматично виділяти приховані теми та оцінювати внесок кожної теми у

конкретний документ [13]. Такі методи є відносно простими в реалізації та не потребують розмічених даних, проте вони мають обмежену здатність враховувати контекст і семантичні зв’язки між словами.

Сучасні підходи до тематичного аналізу фінансових новин поєднують ідеї тематичного моделювання з використанням семантичних векторних подань текстів. Представлення новин у вигляді контекстних ембеддингів дозволяє враховувати смислову близькість між документами навіть за відсутності спільних ключових слів. Це особливо важливо для фінансових новин, де різні джерела можуть описувати одну й ту саму подію з використанням різних формулювань. Поєднання семантичних моделей із методами кластеризації або тематичної класифікації забезпечує більш стійке та інтерпретоване групування новин за темами.

Тематична сегментація новинних потоків має важливе прикладне значення. Вона використовується для побудови аналітичних панелей, фільтрації інформації за обраними напрямами, формування тематичних добірок новин та автоматизованого моніторингу окремих сегментів фінансового ринку. Крім того, аналіз тематичної структури дозволяє порівнювати інформаційний фон різних новинних джерел, виявляти зміни у пріоритетах фінансового дискурсу та відстежувати еволюцію ключових тем у часі.

Таким чином, тематична структура фінансових новин є фундаментальною характеристикою новинних потоків, яка визначає можливості їх подальшого аналізу. Автоматичне виявлення та аналіз тем створює основу для систематизації інформаційного простору, підвищує ефективність аналітичних досліджень і є невід’ємною складовою сучасних методик автоматизації аналізу фінансових новин.

1.3.2. Семантична подібність та агрегація новин

Фінансові новини часто дублюються або перефразовуються різними інформаційними агентствами, що зумовлено високою конкуренцією між джерелами та необхідністю оперативного висвітлення подій. Одна й та сама інформаційна подія може бути опублікована у десятках варіацій, які відрізняються

стилістикою, деталізацією та формулюваннями, але мають спільний смисловий зміст. Це призводить до надмірності даних, ускладнює аналіз великих новинних потоків і може спотворювати загальну картину інформаційного фону. У таких умовах одним із ключових завдань стає виявлення семантично подібних або близьких за змістом повідомлень з метою їх подальшої агрегації та зменшення інформаційного шуму.

Семантична подібність текстів у фінансовому контексті визначається не лише наявністю спільних ключових слів, але й схожістю описуваних подій, суб'єктів та причинно-наслідкових зв'язків. Класичні лексичні підходи, засновані на простому зіставленні термінів, часто виявляються недостатніми, оскільки не враховують контекст і не здатні коректно обробляти перефразовані тексти. Саме тому в сучасних системах аналізу фінансових новин використовуються семантичні методи, що базуються на векторних представленнях текстів.

Для розв'язання задачі оцінки семантичної подібності кожна новина подається у вигляді числового вектора у багатовимірному просторі. Такі векторні подання формуються за допомогою контекстних мовних моделей і відображають смисловий зміст тексту. Подібність між новинами оцінюється за допомогою математичних метрик, найпоширенішою з яких є косинусна міра, що дозволяє визначити кут між векторами і, відповідно, ступінь їх семантичної близькості [26]. Чим вищим є значення подібності, тим більш імовірно, що тексти описують одну й ту саму подію або мають спільну тематику.

Застосування семантичної подібності дає змогу вирішувати низку практичних задач, зокрема виявлення дублікатів і так званих near-duplicate документів, групування новин за подіями та формування узагальнених інформаційних зведень. Агрегація новин дозволяє об'єднувати публікації з різних джерел в один інформаційний блок, що значно спрощує аналіз і зменшує кількість повторюваної інформації. Це є особливо важливим у фінансовій сфері, де оперативне сприйняття суті події часто важливіше за кількість окремих повідомлень.

Методи агрегації новин на основі семантичної подібності широко використовуються в автоматизованих інформаційно-аналітичних системах, оскільки вони знижують навантаження на аналітиків і підвищують інформативність результатів. Замість перегляду великої кількості схожих новин користувач отримує структуровані групи публікацій, що відображають ключові події та тенденції. Крім того, такі підходи застосовуються для побудови інформаційних стрічок, тематичних оглядів та систем раннього виявлення значущих подій.

Важливо зазначити, що точність агрегації новин залежить від якості векторних представлень текстів і правильно обраних порогів подібності. Занадто низький поріг може призводити до об’єднання різних подій в одну групу, тоді як надто високий — до фрагментації однієї події на кілька кластерів. Тому ефективне використання семантичної подібності потребує балансування між чутливістю та узагальненням. Дослідження методів виявлення near-duplicate документів підтверджують доцільність семантичних підходів для роботи з великими текстовими колекціями, зокрема у фінансовому домені [23].

Таким чином, аналіз семантичної подібності та агрегація новин є важливою складовою автоматизованого аналізу інформаційних потоків фінансових новин. Ці методи дозволяють ефективно структурувати новинний контент, зменшувати надмірність даних і створювати основу для подальшої аналітичної обробки у сучасних фінансово-аналітичних системах.

1.3.3. Часова динаміка новинного фону

Фінансові новини мають чітко виражену часову природу, оскільки їх інформативність, вплив та актуальність суттєво змінюються з плином часу. На відміну від статичних інформаційних ресурсів, новинний контент формується у вигляді безперервного потоку повідомлень, інтенсивність якого залежить від поточних економічних, політичних та ринкових подій. Аналіз часової динаміки новинного фону дозволяє досліджувати частоту публікацій, виявляти піки

новинної активності та простежувати зміну тематичних акцентів у різні часові періоди.

Однією з характерних особливостей фінансового новинного потоку є нерівномірність розподілу новин у часі. У періоди стабільності кількість публікацій може бути відносно низькою та однорідною за тематикою, тоді як у моменти криз, публікації макроекономічних звітів або важливих корпоративних подій спостерігаються різкі сплески новинної активності. Аналіз таких піків дозволяє ідентифікувати ключові інформаційні події та оцінювати їх значущість у загальному інформаційному полі.

Часовий аналіз новинних потоків широко використовується для виявлення інформаційних подій, які можуть бути пов’язані зі змінами на фінансових ринках або трансформацією очікувань інвесторів. Зростання кількості публікацій за певною тематикою або поява нових домінуючих тем у короткий проміжок часу часто свідчить про формування важливого інформаційного сигналу. У цьому контексті часова динаміка розглядається не лише як статистична характеристика, але й як індикатор змін у фінансовому середовищі.

Важливим аспектом аналізу часової динаміки є життєвий цикл новини. Кожне новинне повідомлення має фазу появи, активного поширення та поступової втрати актуальності. Автоматизований аналіз дозволяє відстежувати, як швидко певна інформація набирає увагу, скільки часу вона залишається у фокусі новинного потоку та коли втрачає свою значущість. Це особливо актуально для фінансових новин, де навіть незначна затримка в аналізі може призвести до втрати релевантності інформації.

Поєднання часових характеристик із результатами семантичного аналізу відкриває можливість побудови комплексних інформаційних моделей, які відображають еволюцію фінансового дискурсу у часі. Наприклад, аналіз динаміки тематичних кластерів дозволяє простежити, як змінюється увага до певних економічних питань, компаній або ринкових сегментів. Семантична агрегація новин у поєднанні з часовими мітками дає змогу аналізувати не лише окремі події, але й довготривалі інформаційні тенденції.

Часова динаміка новинного фону також є важливою з точки зору побудови автоматизованих систем моніторингу. Безперервний аналіз новин у реальному або квазі-реальному часі дозволяє оперативно виявляти зміни інформаційного середовища, фіксувати появу нових тем та оцінювати інтенсивність інформаційних потоків. Це створює передумови для використання новинного аналізу у системах підтримки прийняття рішень та аналітичних інструментах фінансової сфери.

Таким чином, аналіз часової динаміки новинного фону є важливим елементом дослідження інформаційних потоків фінансових новин. Він доповнює тематичний та семантичний аналіз, дозволяючи враховувати еволюцію інформації у часі та її вплив на загальний інформаційний контекст. Сукупний аналіз часових, тематичних і семантичних характеристик створює теоретичне та методологічне підґрунтя для побудови автоматизованих методик парсингу та семантичного аналізу фінансових новин із використанням готових NLP-моделей [1].

2. АНАЛІЗ МЕТОДІВ ПАРСИНГУ НА ОСНОВІ СЕМАНТИЧНОГО АНАЛІЗУ

2.1. Аналіз існуючих методів парсингу та семантичної обробки фінансових новин

Фінансові новини є одним із найбільш оперативних і водночас складних для аналізу джерел інформації про стан фінансових ринків, діяльність компаній, макроекономічні процеси та регуляторні зміни. Вони швидко реагують на події реального світу та часто випереджають офіційні статистичні дані. Саме через новинні повідомлення формується інформаційний фон, який впливає на очікування інвесторів, ринкові настрої та ухвалення фінансових рішень. На відміну від структурованих наборів даних, таких як котирування, фінансові звіти або макроекономічні індикатори, новини представлені у вигляді неструктурованого тексту, що ускладнює їх формальний аналіз і автоматизовану обробку.

Особливістю фінансових новин є висока варіативність подання інформації. Одна й та сама подія може бути описана різними джерелами з використанням різної лексики, стилістики та акцентів. Крім того, фінансові тексти насичені спеціалізованою термінологією, умовними формулюваннями та непрямими оцінками, що потребує врахування контексту. Через це ключовим завданням стає побудова методів, які забезпечують стабільний та відтворюваний збір текстів (парсинг) і подальшу семантичну інтерпретацію в межах єдиного обробного конвеєра (pipeline).

Під парсингом фінансових новин у даному контексті розуміється сукупність процедур автоматизованого отримання новинного контенту із зовнішніх інформаційних джерел, зокрема новинних порталів та агрегаторів. Цей процес включає доступ до джерела, завантаження новинних записів, вилучення релевантних фрагментів (заголовок, дата публікації, основний текст, автор, посилання), очищення від службових елементів і нормалізацію формату.

Результатом парсингу є уніфіковане представлення новини, придатне для збереження в структурованому вигляді та подальшої автоматизованої обробки.

Семантична обробка текстів фінансових новин є наступним етапом після парсингу та виконує функцію переходу від «сирого» тексту до формалізованих інформаційних представлень. На цьому етапі застосовуються методи обробки природної мови, які дозволяють виділяти ключові смислові характеристики тексту, визначати його тематичну належність, виявляти іменовані сутності та аналізувати загальний інформаційний тон. Семантичний аналіз є необхідною умовою для автоматичної класифікації новин, їх групування за подіями або темами, а також для побудови узагальнених інформаційних зведень.

У сучасних підходах до семантичної обробки фінансових новин все більшого поширення набувають методи, що базуються на векторних представленнях тексту. Такі представлення дозволяють відобразити зміст тексту у багатовимірному числовому просторі, де відстань між векторами відповідає семантичній близькості текстів. Основою цих підходів є трансформерні архітектури, які забезпечують контекстне розуміння тексту та ефективно працюють із довгими послідовностями слів [1]. Подальший розвиток цих ідей реалізовано в моделях типу BERT, які формують універсальні мовні подання та широко застосовуються для аналізу фінансових і економічних текстів [2].

Важливо зазначити, що в межах цієї роботи не виконувалась розробка або навчання власної NLP-моделі. Натомість застосовувалися готові попередньо навчені моделі та бібліотеки, що забезпечують отримання семантичних ембеддингів і базових текстових ознак. Такий підхід відповідає сучасній практиці використання трансформерних моделей як універсального інструменту представлення текстів, коли основна увага зосереджується не на оптимізації архітектури моделі, а на побудові методики її застосування та інтеграції у прикладний сервіс.

Застосування готових NLP-рішень дозволяє суттєво знизити складність системи, скоротити вимоги до обчислювальних ресурсів і забезпечити відтворюваність результатів. Це є особливо важливим у задачах автоматизованого

аналізу фінансових новин, де ключову роль відіграє стабільність процесу обробки та можливість масштабування системи при зростанні обсягу новинних даних. Таким чином, аналіз існуючих методів парсингу та семантичної обробки підтверджує доцільність використання інтегрованого підходу, який поєднує автоматизований збір фінансових новин із сучасними семантичними методами обробки тексту.

2.1.1. Джерела фінансових новин і методи доступу до контенту

Першим і одним із найбільш критичних етапів автоматизації аналізу фінансових новин є вибір джерел даних та механізмів доступу до контенту. Якість, повнота та стабільність новинного корпусу безпосередньо впливають на ефективність подальшого парсингу та семантичної обробки. Фінансові новини публікуються великою кількістю джерел, які відрізняються форматом подання, частотою оновлення, ступенем структурованості та доступністю контенту. Тому при побудові автоматизованих систем аналізу необхідно використовувати такі методи доступу, які забезпечують баланс між надійністю, повнотою інформації та складністю реалізації.

На практиці для отримання фінансових новин застосовуються три основні підходи, кожен з яких має власні особливості, переваги та обмеження.

1. Стандартизовані новинні канали (RSS)

   Багато новинних ресурсів, фінансових порталів і агрегаторів надають доступ до свого контенту через RSS або Atom-канали. Ці формати є стандартизованими механізмами синдикації, що дозволяють отримувати оновлення у структурованому вигляді. Кожен елемент RSS-стрічки, як правило, містить заголовок новини, короткий опис, посилання на повний текст, дату публікації та унікальний ідентифікатор.

   Ключовою перевагою використання RSS є стабільність і простота інтеграції. Оскільки формат даних стандартизований, процес збору новин не залежить від змін дизайну веб-сторінок, структури HTML або рекламних елементів. Це значно знижує ризик збоїв у роботі парсера та спрощує

підтримку системи. З методологічної точки зору RSS є надійним інструментом для регулярного поповнення бази новин та організації безперервного моніторингу інформаційного середовища [21], [22].

Водночас обмеженням цього підходу є те, що RSS-канали часто містять лише короткі анотації або уривки тексту. Для задач глибокого семантичного аналізу цього може бути недостатньо, що зумовлює необхідність комбінування RSS з іншими методами доступу до повного контенту.

2. Парсинг HTML-сторінок

У випадках, коли RSS-канали недоступні, містять урізаний текст або не охоплюють усі необхідні джерела, застосовується прямий парсинг HTML-сторінок новинних сайтів. Цей підхід передбачає завантаження HTML-коду сторінки, аналіз її структури та вилучення релевантних блоків, зокрема заголовка, основного тексту, підзаголовків, тегів та метаданих.

Парсинг HTML-сторінок забезпечує доступ до повного тексту новини, що є важливою перевагою для семантичної інтерпретації, витягу ключових слів і аналізу контексту. На практиці для реалізації такого підходу використовуються HTTP-клієнти для завантаження сторінок та бібліотеки для розбору HTML-дерева. У середовищі Python поширеним є поєднання бібліотек для HTTP-запитів і парсингу HTML, що дозволяє ефективно працювати з веб-контентом [19], [20].

Разом з тим HTML-парсинг є більш вразливим до змін верстки сайтів. Навіть незначні оновлення структури сторінки можуть призвести до некоректної роботи парсера. Тому такий підхід потребує реалізації механізмів підтримки, зокрема регулярного оновлення селекторів, обробки помилок завантаження, редиректів та нестандартних форматів сторінок. Це підвищує складність системи, але забезпечує максимальну повноту даних.

3. API-платформи та агрегатори

Третім підходом до доступу фінансових новин є використання спеціалізованих API, які надаються новинними агрегаторами або аналітичними платформами. Такі сервіси зазвичай повертають дані у

стандартизованих форматах (JSON або XML), що значно спрощує інтеграцію та обробку інформації. Основною перевагою API є стабільність інтерфейсу та чітко визначена структура даних, що зменшує залежність від змін на стороні постачальника контенту.

Однак API-рішення мають і суттєві обмеження. До них належать обмеження швидкості запитів, вимоги до автентифікації, платні ліцензії та залежність від зовнішнього сервісу. Крім того, багато API надають лише метадані та короткі уривки новин, тоді як повний текст може бути недоступним або захищеним авторськими правами. Це обмежує можливості глибокого семантичного аналізу та зменшує гнучкість системи

Таким чином, джерела фінансових новин і методи доступу до контенту суттєво відрізняються за своїми характеристиками. Практика побудови автоматизованих систем аналізу новин показує, що найбільш ефективним є комбінований підхід, який поєднує використання RSS-каналів для стабільного виявлення нових публікацій, HTML-парсинг для отримання повного тексту та, за потреби, API-платформи для доступу до додаткових метаданих. Така стратегія дозволяє забезпечити повноту, надійність і масштабованість системи збору фінансових новин у межах єдиного аналітичного конвеєра.

2.1.2. Попередня підготовка тексту як основа семантичного аналізу

Сирі тексти фінансових новин, навіть після їх вилучення з HTML-сторінок або RSS-каналів, не є безпосередньо придатними для семантичного аналізу. Вони можуть містити технічні артефакти, різні формати подання, шумові елементи та мовні особливості, які ускладнюють подальшу автоматизовану обробку. Тому попередня підготовка тексту є обов’язковим етапом NLP-конвеєра та виконує роль фундаменту для всіх наступних методів семантичної інтерпретації.

На практиці попередня підготовка охоплює такі етапи:

— очищення від службових символів, зайвих пробілів, рекламних вставок;
— приведення до єдиного кодування та формату;
— токенізацію (розбиття на токени);

— лематизацію/стемінг для зведення словоформ до базової форми;
— видалення стоп-слів або фільтрацію нерелевантних токенів;
— виділення іменованих сутностей (організації, країни, персони, валюти).

Очищення тексту від службових символів і сторонніх вставок спрямоване на усунення елементів, які не несуть семантичного навантаження, але можуть впливати на результати аналізу. До таких елементів належать HTML-артефакти, спеціальні символи, повторювані пробіли, рекламні блоки та навігаційні фрагменти. Видалення цього шуму дозволяє зосередитися виключно на змістовній частині новини.

Приведення тексту до єдиного кодування та формату забезпечує коректну роботу подальших NLP-алгоритмів. У фінансових новинних потоках часто зустрічаються тексти з різних джерел, які можуть використовувати різні кодування або форматування. Уніфікація цих аспектів є необхідною умовою стабільної автоматизованої обробки.

Токенізація є базовою операцією, що полягає у розбитті тексту на окремі одиниці — токени, якими можуть бути слова, числа або спеціальні символи. Саме на рівні токенів здійснюється подальший аналіз тексту, побудова векторних представлень та обчислення семантичних ознак. Коректна токенізація є особливо важливою для фінансових новин, де часто використовуються скорочення, числові показники та спеціалізовані терміни.

Лематизація або стемінг застосовуються для зведення різних словоформ до базової форми. Це дозволяє зменшити розмір словника та підвищити узагальнювальну здатність моделей, оскільки різні граматичні форми одного й того ж слова розглядаються як єдина сутність. Для фінансових текстів це особливо актуально через велику кількість варіацій термінів і формулювань.

Видалення стоп-слів або фільтрація нерелевантних токенів спрямовані на усунення слів, які часто зустрічаються в тексті, але не несуть самостійного смислового навантаження. До таких слів належать загальномовні сполучники, прийменники та службові частини мови. Водночас у фінансовому домені важливо

обережно підходити до формування списків стоп-слів, оскільки деякі загальні слова можуть мати специфічне значення в економічному контексті.

Виділення іменованих сутностей є важливим компонентом попередньої обробки, що дозволяє автоматично ідентифікувати ключові об’єкти, такі як назви компаній, фінансових інститутів, країн, валют або окремих осіб. Для фінансових новин це створює основу для подальшого аналізу взаємозв’язків між суб’єктами, агрегації новин за компаніями чи регіонами та побудови більш структурованого представлення інформації.

Усі перелічені операції є типовими для сучасних NLP-пайплайнів і формують базу для подальшої тематизації, класифікації, аналізу семантичної подібності або сентименту. Практична реалізація попередньої підготовки текстів у прикладних системах часто ґрунтується на готових інструментах, зокрема на pipeline-підході бібліотеки spaCy, яка забезпечує токенізацію, лематизацію та розпізнавання іменованих сутностей у межах єдиного узгодженого процесу [16], [17], [18]. Важливим аспектом при цьому є коректний вибір мовних моделей відповідно до мови джерела та адаптація до специфіки фінансової термінології, що безпосередньо впливає на якість подальшого семантичного аналізу.

2.1.3. Семантичне представлення текстів: від ознак до ембеддингів

Ключовим завданням семантичного аналізу фінансових новин є перехід від неструктурованого тексту до числового подання, яке може бути використане для подальших обчислень, порівнянь, класифікації та агрегації. Якість цього подання безпосередньо визначає ефективність усіх наступних етапів NLP-конвеєра, зокрема аналізу подібності, тематизації, кластеризації та виявлення дублікатів. У процесі розвитку методів обробки природної мови сформувалося кілька класів підходів до семантичного представлення текстів, кожен з яких має свої особливості та область застосування.

1. Статистичні (частотні) представлення

   Традиційні підходи до представлення тексту ґрунтуються на частотному аналізі термінів, зокрема моделях типу bag-of-words або TF-IDF.

У таких моделях документ подається у вигляді вектора, елементи якого відповідають частотам слів або зваженим значенням термінів у корпусі. Перевагою цих методів є простота реалізації, інтерпретованість та відносно низькі обчислювальні витрати.

Однак частотні представлення мають суттєві обмеження з точки зору семантики. Вони не враховують порядок слів, контекст та синонімію, що є критичним для фінансових новин. Одна й та сама подія може бути описана різними формулюваннями, використовуючи різні терміни або стилістичні конструкції. У таких випадках частотні моделі не здатні виявити семантичну близькість між текстами, що знижує їхню ефективність у задачах агрегації новин або виявлення дублікатів.

2. Трансформерні контекстні подання

Суттєвий прорив у семантичному аналізі текстів відбувся з появою моделей сімейства Transformer. Їх ключовою ідеєю є механізм self-attention, який дозволяє моделі враховувати взаємозв’язки між усіма словами в тексті незалежно від їх позиції [1]. Завдяки цьому значення кожного слова формується з урахуванням контексту, що принципово відрізняє трансформерні підходи від класичних частотних моделей.

Подальший розвиток цих ідей привів до появи моделей типу BERT, які навчаються на великих корпусах текстів у режимі попереднього навчання і можуть бути використані як універсальна основа для широкого спектра NLP-задач [2]. Для фінансових новин це має особливе значення, оскільки тексти часто містять складні синтаксичні конструкції, багатозначні терміни та залежності між подіями, які важко коректно інтерпретувати без урахування контексту.

У прикладних системах автоматизованого аналізу фінансових новин широко застосовуються sentence embeddings, які дозволяють представляти не окремі токени, а цілі речення або документи у вигляді векторів фіксованої розмірності. Такі подання є зручними для обчислення семантичної подібності, кластеризації новин та побудови агрегованих зведень. Важливим

напрямом розвитку є також створення мульти- та монолінгвальних ембеддингів із переносом знань, що дозволяє працювати з новинами різними мовами в єдиному векторному просторі [3]. Практичним стандартом для реалізації таких підходів є бібліотека Sentence-Transformers, яка надає набір готових моделей та інструментів для побудови семантичних векторних подань текстів [4].

3. Використання спеціалізованих фінансових мовних моделей

Загальномовні трансформерні моделі демонструють високу універсальність, проте не завжди оптимально відображають специфіку фінансового домену. Фінансова лексика має власні семантичні особливості, де окремі терміни можуть мати значення, відмінні від повсякденного мовлення. Наприклад, слова liability, debt, volatility або downgrade у фінансовому контексті мають чітко визначене професійне значення, яке може бути некоректно інтерпретоване загальними мовними моделями.

Для зменшення цього ефекту застосовуються спеціалізовані мовні моделі, адаптовані до фінансових текстів. Такі моделі попередньо навчаються або донавчаються на корпусах фінансових новин, звітів та економічної документації, що дозволяє їм краще відображати доменну семантику. Прикладом такого підходу є FinBERT, який спеціалізований на аналізі фінансових текстів і широко використовується у задачах сентимент-аналізу та семантичної інтерпретації новин [5].

Таким чином, еволюція методів семантичного представлення текстів демонструє перехід від простих статистичних ознак до контекстно-орієнтованих ембеддингів. У контексті автоматизованого парсингу фінансових новин найбільш практично обґрунтованим є використання готових трансформерних моделей та sentence embeddings, які забезпечують баланс між глибиною семантичної інтерпретації, масштабованістю та зручністю інтеграції в прикладні інформаційно-аналітичні системи.

2.1.4. Методи витягу ключових слів і тематизації

Після формування уніфікованого текстового подання фінансової новини та виконання попередньої обробки виникає задача виділення тих фрагментів тексту, які найбільш повно репрезентують її зміст. Витяг ключових слів і тематизація є важливими етапами семантичного аналізу, оскільки саме вони дозволяють перейти від суцільного тексту до компактного й інтерпретованого опису новини. У контексті фінансових новин це дає змогу швидко ідентифікувати сутність події, її основних учасників та тематичну спрямованість.

Найпоширеніші підходи до витягу ключових слів і тематизації включають такі методи:

— Графові методи ранжування ключових слів (TextRank): будуються графи співвзустрічальності термінів, після чого виконується ранжування вузлів [14].
— BERT-based keyword extraction (KeyBERT): ключові слова визначаються на основі семантичної близькості кандидатів до векторного представлення документа, що дозволяє враховувати контекст [15].
— Тематичне моделювання (LDA): дозволяє виділяти латентні теми в колекції документів і групувати новини за тематичними кластерами [13].

Графові методи, зокрема TextRank, ґрунтуються на ідеї представлення тексту у вигляді графа, де вершини відповідають словам або фразам, а ребра відображають їх спільну появу в контексті. Подальше застосування алгоритмів ранжування дозволяє визначити найбільш «впливові» терміни, які займають центральне місце в структурі тексту. Перевагою таких методів є їх незалежність від навчальних даних і відносна простота реалізації. Водночас вони здебільшого спираються на лексичні зв’язки й можуть бути менш чутливими до глибокого семантичного контексту, що є важливим для фінансових текстів з великою кількістю специфічних формулювань.

Підхід KeyBERT належить до сучасних методів витягу ключових слів, які використовують семантичні векторні представлення. Його основна ідея полягає в тому, що ключові слова або фрази мають бути семантично близькими до загального векторного подання документа. Це дозволяє враховувати контекст і зміст тексту, а

не лише частоту або позицію слів. Для фінансових новин такий підхід є особливо корисним, оскільки дозволяє виділяти терміни, які мають змістовну значущість навіть за відносно низької частоти появи.

Тематичне моделювання на основі LDA орієнтоване на аналіз колекції документів загалом, а не окремих новин. Модель дозволяє автоматично виявляти латентні теми та оцінювати їх внесок у кожен документ. Це створює можливість групування фінансових новин за тематичними кластерами, наприклад за напрямами макроекономіки, корпоративних подій або фінансових ринків. Перевагою LDA є її здатність виявляти глобальну тематичну структуру корпусу, проте модель має обмежену здатність враховувати контекст і часто чутлива до якості попередньої обробки текстів.

У прикладних системах аналізу фінансових новин зазначені підходи зазвичай не використовуються ізольовано. Їх комбінування дозволяє отримати більш повне та стійке представлення інформації. Зокрема, статистичне тематичне моделювання може забезпечувати загальну структуру корпусу новин і визначати основні напрями інформаційного потоку, тоді як ембеддинг-орієнтовані методи, такі як KeyBERT, дозволяють виконувати контекстно чутливий витяг ключових термінів для окремих публікацій. Такий комбінований підхід є практично обґрунтованим для роботи з фінансовими новинами, оскільки поєднує інтерпретованість, масштабованість і глибину семантичного аналізу.

2.1.5. Сентимент-аналіз у фінансових новинах

Сентимент-аналіз у фінансовій сфері розглядається як процедура визначення емоційного або оціночного тону тексту з метою виявлення позитивних, негативних або нейтральних сигналів. На відміну від загальнотематичних текстів, у фінансових новинах сентимент має контекстно-залежний характер і часто пов’язаний не з емоційною забарвленістю мови, а з очікуваннями ринку, оцінкою ризиків, перспектив зростання або погіршення фінансового стану. Тому інтерпретація тональності фінансових новин потребує врахування доменних особливостей і специфіки економічної термінології [9].

Сентимент фінансової новини може відображати ставлення до корпоративних результатів, макроекономічних показників, регуляторних рішень або геополітичних подій. Навіть формально нейтральні формулювання можуть містити приховані негативні або позитивні сигнали для ринку. У зв’язку з цим методи сентимент-аналізу у фінансовій сфері суттєво відрізняються від підходів, які застосовуються для аналізу соціальних мереж або загальних новинних текстів. Основні групи підходів:

1. Словникові методи

   Словникові методи ґрунтуються на використанні заздалегідь сформованих списків слів із відомою полярністю. Для фінансових текстів застосування загальних сентимент-лексиконів є обмеженим, оскільки багато термінів у фінансовому контексті мають інше значення, ніж у повсякденному мовленні. Наприклад, слова на кшталт liability або debt у фінансових документах не завжди мають негативну конотацію, а є нейтральними описовими поняттями.

   У зв’язку з цим у фінансовому домені використовуються спеціалізовані словники, зокрема словники Loughran–McDonald, які були розроблені на основі аналізу фінансової звітності та економічних текстів [6], [7]. Ці словники містять категорії позитивних, негативних, невизначених та модальних термінів, що дозволяє більш коректно оцінювати тональність фінансових новин.

   Перевагами словникових методів є їхня простота, пояснюваність результатів та відсутність потреби у навчанні моделей. Вони легко інтегруються в автоматизовані конвеєри аналізу новин і можуть використовуватися для швидкої оцінки інформаційного фону. Водночас такі підходи є чутливими до контексту й не завжди здатні враховувати складні мовні конструкції, заперечення або іронію, що обмежує точність інтерпретації.

2. Моделі на основі трансформерів

Сучасні підходи до сентимент-аналізу у фінансовій сфері активно використовують трансформерні мовні моделі, які забезпечують контекстну інтерпретацію тексту. Завдяки механізму self-attention такі моделі здатні враховувати взаємозв'язки між словами та їх оточенням, що є критично важливим для аналізу складних фінансових формулювань.

Прикладом доменно-орієнтованого підходу є FinBERT, який було адаптовано до фінансових текстів і який демонструє підвищену чутливість до специфічної лексики та стилістики фінансових новин [5]. Такі моделі здатні краще розрізняти нейтральні описові твердження та дійсно негативні або позитивні сигнали, що робить їх більш придатними для аналізу новинного фону.

Разом з тим використання трансформерних моделей супроводжується підвищеними обчислювальними витратами та складністю інтеграції. Крім того, ефективність таких моделей значною мірою залежить від корпусів, на яких вони були попередньо навчені. У практичних системах це потребує компромісу між точністю семантичної інтерпретації та продуктивністю обробки новинних потоків.

У межах даної роботи акцент робиться саме на використанні готових попередньо навчених моделей і словникових ресурсів, без розробки або донавчання власних моделей. Такий підхід дозволяє зосередитися на побудові методики автоматизованого парсингу фінансових новин, інтеграції модулів семантичного та сентимент-аналізу в єдиний сервіс і забезпеченні відтворюваності результатів. Використання готових рішень є практично обґрунтованим для прикладних інформаційно-аналітичних систем, де ключову роль відіграє стабільність, масштабованість і швидкість обробки новинних даних.

2.1.6. Виявлення дублікатів і близьких за змістом новин

Окремою важливою прикладною задачею при роботі з великими потоками фінансових новин є усунення дублювання та надмірності інформації. У сучасному інформаційному середовищі одна й та сама подія часто висвітлюється різними медіа, новинними агентствами або агрегаторами з мінімальними редакційними відмінностями. Це може бути перефразування заголовків, зміна порядку подання фактів або додавання коротких коментарів. Без спеціальних механізмів фільтрації такі публікації створюють значний інформаційний шум і ускладнюють подальший аналіз.

Основною метою виявлення дублікатів є ідентифікація новин, що описують одну й ту саму подію, з подальшою агрегацією або відбором репрезентативних публікацій. Для цього використовують:

— методи near-duplicate detection, які оцінюють схожість документів на основі їх подання та ознак [23], [24];

— косинусну схожість ембеддингів, що дозволяє виявляти семантично близькі тексти навіть при різному формулюванні [26].

Методи near-duplicate detection історично розроблялися для пошуку майже ідентичних документів у великих колекціях текстів. Вони базуються на припущенні, що дублікати мають високу лексичну або структурну подібність. Такі підходи ефективні для виявлення новин, які були скопійовані або змінені незначною мірою. Їх перевагою є висока швидкодія та можливість масштабування на великі корпуси. Водночас вони менш ефективні у випадках, коли тексти описують одну подію, але використовують суттєво різні формулювання.

Сучасні підходи дедалі частіше використовують семантичні ембеддинги, які дозволяють перейти від поверхневого порівняння слів до аналізу змісту тексту. У такому випадку кожна новина представляється у вигляді вектора у багатовимірному просторі, після чого подібність між новинами оцінюється за допомогою косинусної міри або інших метрик. Цей підхід дає змогу виявляти близькі за змістом тексти навіть тоді, коли вони мають різний словниковий склад або стилістичні особливості.

Перевагою ембеддинг-орієнтованого підходу є його контекстна чутливість та здатність працювати з перефразованими текстами. Це особливо актуально для фінансових новин, де різні джерела можуть по-різному інтерпретувати події, зберігаючи при цьому однакову семантичну основу. Разом з тим точність такого підходу залежить від якості векторних представлень і вибору порогових значень схожості, що потребує ретельного налаштування в прикладних системах.

З практичного погляду, детекція дублікатів і близьких за змістом новин має низку важливих переваг. По-перше, вона дозволяє уникати повторного врахування однієї події при побудові тематичної статистики або аналізі інформаційного фону. По-друге, агрегація новин за подіями дає змогу формувати компактні та інформативні зведення, що зменшує навантаження на аналітиків і користувачів системи. По-третє, зменшення обсягу дубльованих даних позитивно впливає на продуктивність наступних етапів аналізу, зокрема тематизації та сентимент-аналізу.

Таким чином, виявлення дублікатів є невід’ємним компонентом автоматизованих систем парсингу та семантичного аналізу фінансових новин. Поєднання класичних методів near-duplicate detection із семантичними підходами на основі ембеддингів дозволяє досягти балансу між швидкодією, масштабованістю та глибиною аналізу, що є критично важливим для роботи з великими та динамічними новинними потоками.

2.2. Аналіз переваг та недоліків методів парсингу на основі семантичного аналізу

Методи парсингу фінансових новин із використанням семантичного аналізу формують основу сучасних інформаційно-аналітичних систем, орієнтованих на роботу з великими обсягами неструктурованих текстових даних. Їх ефективність визначається здатністю забезпечувати стабільний збір контенту, коректну інтерпретацію змісту та адаптацію до динамічних змін у новинному середовищі. Водночас кожен клас методів має власні переваги та обмеження, які необхідно

враховувати при побудові прикладних систем автоматизації аналізу фінансової інформації.

Однією з ключових переваг методів парсингу, інтегрованих із семантичним аналізом, є можливість роботи з великими масивами новинних даних у автоматизованому режимі. На відміну від ручного аналізу, такі підходи дозволяють обробляти сотні й тисячі новин за короткий проміжок часу, забезпечуючи оперативність доступу до актуальної інформації. Це особливо важливо для фінансової сфери, де швидкість реагування на події має вирішальне значення.

Суттєвою перевагою семантичного підходу є врахування контексту та смислових зв’язків між словами і фразами. Використання контекстних мовних моделей дозволяє зменшити залежність від конкретних формулювань і виявляти схожі за змістом новини, навіть якщо вони подані різними джерелами у відмінній лексичній формі. Це створює передумови для ефективної агрегації новин, виявлення інформаційних хвиль та зменшення дублювання контенту [1].

Ще однією перевагою є універсальність семантичних представлень тексту. Векторні подання новин можуть бути використані для широкого спектра задач: тематичної класифікації, витягу ключових слів, пошуку подібних статей або аналізу інформаційного фону. Це дозволяє будувати гнучкі системи, у яких результати одного етапу аналізу можуть бути повторно використані в інших модулях без необхідності додаткової обробки.

Водночас методи парсингу на основі семантичного аналізу мають і низку обмежень. Одним із основних недоліків є залежність від якості вхідних даних. Помилки на етапі парсингу, зокрема некоректне вилучення тексту, наявність рекламних або службових фрагментів, можуть негативно впливати на результати семантичного аналізу. Таким чином, стабільність і точність всієї системи значною мірою залежать від надійності механізмів збору та попередньої обробки текстів.

Іншим важливим обмеженням є обчислювальна складність семантичних методів, особливо тих, що базуються на трансформерних моделях. Обчислення векторних представлень для великої кількості новин може вимагати значних ресурсів, що ускладнює використання таких підходів у режимі реального часу або

при обмежених обчислювальних можливостях. Це потребує оптимізації конвеєра обробки та раціонального використання вже обчислених результатів [4].

Також варто враховувати, що готові попередньо навчені мовні моделі не завжди повною мірою відображають специфіку фінансової термінології. Універсальні NLP-моделі можуть некоректно інтерпретувати окремі економічні поняття або їхнє контекстне значення, що частково знижує точність семантичної інтерпретації. Для зменшення цього ефекту застосовуються доменно-орієнтовані моделі або додаткові словникові підходи, проте це ускладнює архітектуру системи [5].

Окремим недоліком є обмежена пояснюваність результатів семантичного аналізу. Векторні представлення та обчислення подібності між текстами не завжди мають прозору інтерпретацію з точки зору кінцевого користувача. Це може ускладнювати аналіз причинно-наслідкових зв’язків та вимагати додаткових механізмів візуалізації або пояснення результатів.

Таким чином, аналіз переваг і недоліків методів парсингу фінансових новин на основі семантичного аналізу показує, що ці підходи є потужним інструментом автоматизованої роботи з текстовими даними, але потребують зваженого використання. Ефективність таких методів досягається за умови поєднання стабільного механізму збору даних, якісної попередньої обробки тексту та раціонального застосування семантичних моделей у межах єдиного аналітичного конвеєра.

2.2.1. Методи парсингу новинних джерел

Методи парсингу фінансових новин можна умовно поділити на дві основні групи: підходи, що базуються на використанні стандартизованих каналів синдикації контенту, та методи прямого аналізу веб-сторінок новинних ресурсів. Кожна з цих груп має власні особливості, сфери застосування та обмеження, які безпосередньо впливають на якість і повноту подальшого семантичного аналізу.

Використання RSS каналів є одним із найбільш поширених і стабільних підходів до автоматизованого збору новин. Основною перевагою цього методу є

чітко визначена структура даних, у якій кожен запис містить стандартизований набір полів: заголовок, короткий опис, посилання на повну статтю, дату публікації та, у деяких випадках, унікальний ідентифікатор. Такий формат значно спрощує процес парсингу, зменшує кількість помилок під час збору даних і робить систему менш залежною від змін у зовнішньому вигляді веб-ресурсів [21], [22].

Крім того, RSS-канали є зручним інструментом для організації регулярного моніторингу новин, оскільки дозволяють швидко визначати появу нових публікацій без необхідності повторного завантаження всього контенту сайту. Наявність часових міток у записах RSS є важливою перевагою для аналізу часової динаміки новинного фону, побудови хронологічних зрізів та дослідження інформаційних хвиль у фінансовому середовищі.

Водночас суттєвим обмеженням RSS-підходу є те, що більшість каналів надає лише анотації або скорочені версії текстів. У фінансовій журналістиці саме повний текст статті часто містить ключові деталі, контекстні уточнення та формулювання, необхідні для коректного семантичного аналізу. Тому використання RSS-каналів у чистому вигляді не завжди дозволяє виконати повноцінну семантичну обробку без додаткового звернення до першоджерела.

Альтернативним підходом є парсинг HTML-сторінок новинних ресурсів, який передбачає безпосередній аналіз структури веб-сторінок і вилучення текстового контенту з HTML-документів. Цей метод забезпечує доступ до повного тексту новини, включаючи підзаголовки, цитати, таблиці та інші елементи, що розширює можливості семантичної інтерпретації та витягу інформації. Для фінансових новин це має особливе значення, оскільки навіть незначні деталі формулювань можуть впливати на загальне розуміння події.

Разом з тим HTML-парсинг є більш складним з технічної точки зору. Він сильно залежить від структури веб-сторінки, яка може змінюватися без попередження, що потребує регулярного оновлення правил парсингу та селекторів. Крім того, необхідно враховувати обробку помилок, редиректів, різних форматів верстки, а також можливі обмеження доступу з боку сайтів [19], [20]. Це ускладнює підтримку системи та підвищує вимоги до її надійності.

Таким чином, вибір методу парсингу фінансових новин є компромісом між стабільністю та повнотою даних. На практиці ефективним підходом є поєднання обох методів: використання RSS-каналів для швидкого виявлення нових публікацій та застосування HTML-парсингу для отримання повного тексту новин, необхідного для подальшого семантичного аналізу. Така комбінована стратегія дозволяє підвищити якість зібраних даних і забезпечити більш глибоку та коректну інтерпретацію фінансового новинного контенту

2.2.2. Семантичні методи аналізу текстів

Семантичний аналіз фінансових новин має суттєві переваги порівняно з поверхневими лексичними підходами, що базуються виключно на частоті слів або простих правилах зіставлення термінів. На відміну від таких методів, семантичний підхід орієнтований на розуміння смислового наповнення тексту та врахування контексту, в якому використовуються слова і фрази. Це є особливо важливим для фінансових новин, де одні й ті самі терміни можуть мати різне значення залежно від економічної ситуації, типу події або ринкового середовища.

Використання контекстних мовних моделей дозволяє враховувати смислові зв’язки між словами, реченнями та абзацами, а також коректно інтерпретувати складні фінансові формулювання. Такі моделі здатні розрізняти позитивні та негативні конотації термінів залежно від контексту, що є критично важливим для аналізу фінансових новин, аналітичних оглядів та економічних прогнозів. Особливо ефективними в цьому аспекті є трансформерні архітектури, які використовують механізм уваги для побудови універсальних векторних представлень текстів та забезпечують високу якість семантичного узагальнення [1], [2].

Семантичні методи аналізу дозволяють вирішувати широкий спектр прикладних задач, зокрема тематичну класифікацію новин, витяг ключових слів і фраз, групування текстів за подіями, а також визначення семантичної подібності між статтями з різних джерел. Представлення текстів у вигляді векторів у багатовимірному просторі створює можливість застосування математичних

методів порівняння, що є основою для побудови аналітичних і рекомендаційних компонентів у фінансових інформаційних системах.

Використання готових попередньо навчених моделей є значною перевагою з практичної точки зору. Такий підхід дозволяє інтегрувати семантичний аналіз у прикладні системи без необхідності навчання власних моделей, збору великих розмічених корпусів та налаштування складної обчислювальної інфраструктури. Це суттєво знижує поріг входу для реалізації семантичного аналізу та робить можливим його застосування у веб-орієнтованих сервісах і дослідницьких проєктах. Крім того, використання готових моделей спрощує відтворюваність результатів і прискорює процес розробки систем автоматизації аналізу фінансових новин [4].

Разом з тим, застосування готових семантичних моделей має і певні обмеження. Узагальнені мовні моделі, навчені на великих корпусах загальномовних текстів, не завжди оптимально відображають специфіку фінансового домену. Це може призводити до втрати важливих нюансів або некоректної інтерпретації окремих термінів і виразів, що мають специфічне значення в економічному контексті. Для зменшення цього ефекту застосовуються доменно-орієнтовані моделі, адаптовані до фінансових текстів, однак навіть вони залишаються залежними від якості та структури корпусів, на яких були попередньо навчені [5].

Окремим аспектом є обчислювальна складність семантичних моделей. Трансформерні архітектури потребують значних ресурсів для обробки великих обсягів текстів, що може впливати на масштабованість системи, особливо при аналізі потоків новин у квазі-реальному часі. Це зумовлює необхідність оптимізації процесів обробки, повторного використання обчислених векторних представлень та раціонального проєктування архітектури системи.

Таким чином, семантичні методи аналізу текстів є потужним інструментом для роботи з фінансовими новинами, оскільки забезпечують глибшу інтерпретацію змісту порівняно з традиційними підходами. Водночас їх ефективне застосування потребує усвідомлення обмежень, пов’язаних із доменною специфікою,

обчислювальною складністю та залежністю від попередньо навчених моделей, що має бути враховано при розробці автоматизованих систем аналізу фінансової інформації.

2.2.3. Методи витягу інформації та агрегації новин

Методи витягу інформації та агрегації фінансових новин відіграють ключову роль у зменшенні інформаційного шуму та формуванні структурованого уявлення про новинний потік. За умов постійного зростання кількості публікацій з різних джерел саме ці методи дозволяють перейти від хаотичного набору текстів до впорядкованих тематичних блоків, що відображають сутність фінансових подій та процесів. Основними напрямами в цій області є витяг ключових слів, тематична класифікація та семантична агрегація новин.

Витяг ключових слів і фраз дозволяє виділити найбільш інформативні елементи тексту, які репрезентують його основний зміст. У фінансових новинах це можуть бути назви компаній, фінансові інструменти, макроекономічні показники або події. Графові методи, зокрема підходи на основі аналізу зв’язків між словами, дозволяють визначати важливість термінів з урахуванням їхньої ролі в структурі тексту. Водночас ембеддинг-орієнтовані методи використовують семантичну близькість між текстом новини та кандидатами ключових слів, що дає змогу враховувати контекст і уникати виділення формально частотних, але змістовно другорядних термінів [14], [15].

Тематична класифікація та кластеризація новин спрямовані на групування публікацій за спільною тематикою або подіями. Такі методи дозволяють ідентифікувати основні напрями інформаційного потоку, наприклад новини про фінансові ринки, корпоративні звіти, монетарну політику або геополітичні фактори. Представлення новин у вигляді семантичних векторів створює можливість застосування алгоритмів кластеризації та класифікації, що значно спрощує аналіз великих корпусів текстів і забезпечує більш цілісне бачення фінансового інформаційного середовища.

Важливою перевагою методів витягу інформації та агрегації є їхня гнучкість і можливість комбінування з іншими компонентами NLP-конвеєра. Результати витягу ключових слів можуть використовуватися для побудови тематичних міток, кластеризація — для групування новин за подіями, а семантичні вектори — для подальшого аналізу подібності. Такі підходи добре масштабуються на великі масиви текстів і можуть бути використані для створення зведень, аналітичних панелей або інформаційних оглядів, що є актуальним для фінансової аналітики.

Разом з тим результати витягу інформації значною мірою залежать від якості попередньої обробки текстів, коректності нормалізації та вибору параметрів моделей. У разі використання неоднорідних джерел новин можливі ситуації, коли одна й та сама тема розбивається на кілька кластерів через різницю у формулюваннях або стилі подання інформації. З іншого боку, надмірне узагальнення може призводити до об’єднання різних подій в одну групу, що знижує точність тематичної агрегації.

Окремою важливою складовою семантичного підходу є виявлення дублікатів і близьких за змістом публікацій. Фінансові новини часто повторюються або перефразовуються різними агентствами, що створює надлишковість у новинному потоці. Методи виявлення near-duplicate документів і семантичної подібності дозволяють агрегувати такі публікації та розглядати їх як одну інформаційну подію, що зменшує навантаження на аналітичні системи та підвищує інформативність результатів [23], [24]. Точність такого групування залежить від стабільності векторних представлень текстів і правильно обраних метрик подібності, зокрема косинусної міри [26].

Узагальнюючи, можна зазначити, що методи витягу інформації та агрегації новин є невід’ємною частиною сучасних систем аналізу фінансових новин. Поєднання автоматизованого парсингу з семантичним аналізом на основі готових NLP-моделей є практично обґрунтованим підходом для роботи з великими обсягами фінансового текстового контенту. Водночас жоден з розглянутих методів не є універсальним, а ефективність системи в цілому визначається балансом між стабільністю збору даних, глибиною семантичної інтерпретації та можливостями

масштабування. Саме ці аспекти мають бути враховані при подальшій розробці та вдосконаленні методики автоматизації парсингу фінансових новин.

3. РОЗРОБКА МЕТОДИКИ АВТОМАТИЗАЦІЇ ПАРСИНГКУ ТА СЕМАНТИЧНОГО АНАЛІЗУ ФІНАНСОВИХ НОВИН

3.1. Загальна схема методики та постановка задачі

Запропонована у даній роботі методика автоматизації парсингу та семантичного аналізу фінансових новин спрямована на вирішення задачі ефективної обробки великих обсягів неструктурованої текстової інформації, що надходить з відкритих новинних джерел. Основною ідеєю методики є побудова цілісного автоматизованого конвеєра, який забезпечує безперервний збір фінансових новин, їх попередню обробку, семантичну інтерпретацію та подальше представлення результатів у зручному для аналізу вигляді.

На відміну від підходів, орієнтованих на навчання власних моделей машинного навчання, у даній методиці зроблено акцент на використанні готових попередньо навчених NLP-моделей. Такий підхід дозволяє зменшити складність реалізації, забезпечити відтворюваність результатів та зосередитися безпосередньо на задачах автоматизації збору і семантичного аналізу фінансових текстів. Використання трансформерних архітектур як бази для семантичного представлення текстів є обґрунтованим з огляду на їхню здатність ефективно враховувати контекст і смислові залежності у природній мові [1], [2].

Методика розглядає фінансову новину як багатокомпонентний об’єкт, що включає заголовок, основний текст, часову мітку, джерело та допоміжні метадані. На початковому етапі здійснюється автоматизований парсинг новин з відкритих ресурсів із використанням стандартизованих каналів синдикації або аналізу веб-сторінок. Отримані тексти приводяться до уніфікованого формату, що забезпечує коректну подальшу обробку незалежно від джерела походження.

Ключовим елементом методики є семантичний аналіз, який базується на побудові векторних представлень текстів. Такі представлення дозволяють

формалізувати зміст фінансових новин у вигляді числових векторів та виконувати операції порівняння, групування і тематичної сегментації. Застосування sentence-level ембеддингів дає змогу аналізувати новини як цілісні смислові одиниці, що є особливо важливим для подійно-орієнтованого фінансового контенту [4].

Запропонована методика також передбачає використання допоміжних процедур семантичної обробки, зокрема витягу ключових слів, тематичної класифікації та визначення семантичної подібності між новинами. Це дозволяє структурувати новинний потік, зменшувати інформаційний шум і формувати узагальнені інформаційні блоки для подальшого аналізу. У фінансовому контексті такі підходи широко застосовуються для роботи з новинами та економічними текстами, що підтверджується сучасними оглядами методів фінансового текстового аналізу [9].

Таким чином, загальна концепція методики полягає у поєднанні автоматизованого парсингу фінансових новин з сучасними семантичними методами обробки тексту на основі готових NLP-моделей. Це забезпечує практичну придатність рішення, його масштабованість та можливість інтеграції у веб-орієнтовані інформаційно-аналітичні системи без необхідності розробки складних моделей машинного навчання з нуля.

3.2. Архітектура та логіка роботи методики автоматизованого парсингу і семантичного аналізу

Запропонована методика автоматизації парсингу та семантичного аналізу фінансових новин реалізується у вигляді багатокомпонентної системи, побудованої за принципом послідовного обробного конвеєра (pipeline). Основною метою такої архітектури є забезпечення масштабованого, керованого та відтворюваного процесу роботи з новинними потоками — від моменту ініціації збору до отримання структурованих семантичних результатів.

Загальна логіка роботи методики базується на розділенні відповідальності між окремими модулями системи. Кожен модуль виконує чітко визначену функцію: ініціацію процесів, збір даних, збереження статей, семантичний аналіз та обчислення подібності. Такий підхід відповідає сучасним принципам побудови інформаційно-аналітичних систем і дозволяє гнучко розширювати або модифікувати окремі етапи без впливу на всю систему.

На рисунку 3.1 представлено узагальнену діаграму послідовності взаємодії компонентів системи під час виконання парсингу та NLP-аналізу фінансових новин.

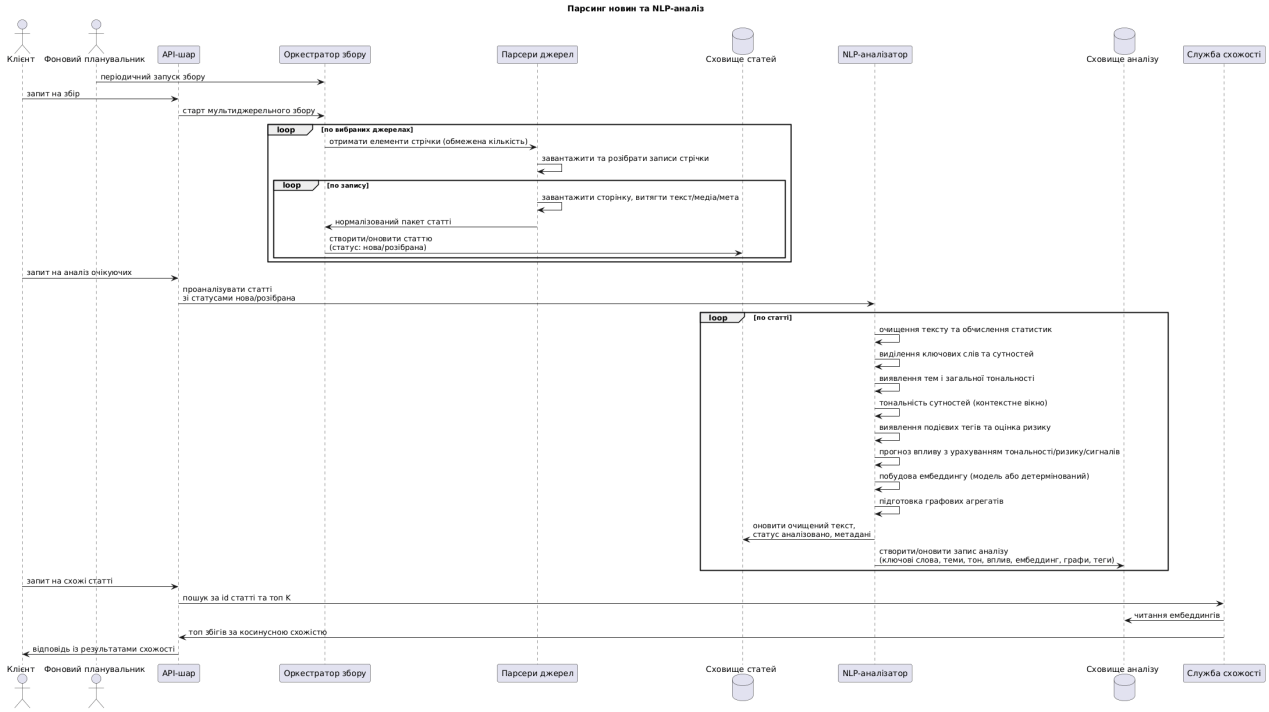


Рис. 3.1. Діаграма процесу парсингу фінансових новин та NLP-аналізу

Діаграма відображає повний життєвий цикл обробки новин: від ініціації збору клієнтом або фоновим планувальником до отримання результатів семантичного аналізу та пошуку схожих статей. Вона демонструє як послідовну, так і циклічну природу обробки даних, що є характерною для потоків новин з декількох джерел.

3.2.1. Процес автоматизованого збору та збереження фінансових новин

Початковим етапом методики є автоматизований збір фінансових новин. Ініціація цього процесу може здійснюватися як безпосередньо клієнтом через API, так і фоновим планувальником у режимі періодичного запуску. Такий підхід дозволяє поєднувати ручний та автоматичний сценарії використання системи.

Після ініціації запиту API-шар передає керування оркестратору збору, який відповідає за запуск мультиджерельного парсингу. Оркестратор послідовно або паралельно опрацьовує перелік налаштованих джерел новин. Для кожного джерела отримуються елементи стрічки (RSS/Atom або HTML-індекси) з обмеженням на кількість записів, що дозволяє контролювати навантаження та обсяг оброблюваних даних [21], [22].

Далі для кожного запису виконується завантаження повної сторінки новини, з якої витягується текстовий вміст, метадані та допоміжна інформація. На цьому етапі формується нормалізований пакет статті, який приводиться до уніфікованої структури незалежно від джерела походження. Після цього стаття створюється або оновлюється у сховищі статей із відповідним статусом (наприклад, «нова» або «розібрана»).

Такий підхід забезпечує:

— відокремлення логіки збору від логіки аналізу;
— можливість повторної обробки статей;
— накопичення історичного корпусу фінансових новин для подальших досліджень.

3.2.2. Процес семантичного аналізу та обробки результатів

Другим ключовим етапом методики є семантичний аналіз фінансових новин, який виконується для статей зі статусами «нова» або «розібрана». Ініціація аналізу також здійснюється через API або фонові механізми, що дозволяє асинхронно обробляти накопичені дані.

У межах NLP-аналізу для кожної статті послідовно виконуються такі операції:

— очищення та нормалізація тексту;
— обчислення базових текстових статистик;
— виділення ключових слів та іменованих сутностей;
— визначення тематики та загальної тональності;
— аналіз контекстної тональності сутностей;
— формування семантичних ембеддингів тексту.

Застосування векторних подань дозволяє надалі використовувати результати аналізу для обчислення семантичної подібності між статтями, зокрема за допомогою косинусної міри [26]. Для цього реалізовано окрему службу схожості, яка виконує пошук топ-K найбільш близьких за змістом новин на основі попередньо збережених ембеддингів.

Результати семантичного аналізу зберігаються у сховищі аналізу у вигляді структурованого запису, що включає очищений текст, ключові слова, теми, показники тональності, ембеддинги та допоміжні агреговані дані. Це дозволяє використовувати отримані результати як для візуалізації в інтерфейсі застосунку, так і для подальших аналітичних або дослідницьких задач.

Таким чином, об’єднана архітектура, представлена на рис. 3.1, забезпечує логічно завершений та масштабований процес автоматизованого парсингу і семантичного аналізу фінансових новин, у якому готові NLP-моделі виступають інструментом інтерпретації текстів, а не об’єктом навчання чи оптимізації [1], [4].

3.3. Вибір програмних засобів та обґрунтування технологічного стеку

Реалізація методики автоматизації парсингу та семантичного аналізу фінансових новин потребує використання програмних засобів, які забезпечують ефективну роботу з неструктурованими текстовими даними, підтримують сучасні підходи до обробки природної мови та дозволяють інтегрувати окремі компоненти в єдину веб-орієнтовану систему. Вибір технологічного стеку у даній роботі

здійснювався з урахуванням практичної доцільності, поширеності інструментів, наявності якісної документації та можливості масштабування рішення.

Основною мовою програмування для реалізації серверної частини системи обрано Python. Це зумовлено тим, що Python є де-факто стандартом у сфері обробки природної мови та аналізу текстових даних. Його екосистема містить широкий набір бібліотек для парсингу веб-ресурсів, роботи з текстами та інтеграції сучасних NLP-моделей. Крім того, Python забезпечує високу швидкість розробки та зручність прототипування, що є важливим фактором при реалізації дослідницьких і прикладних інформаційно-аналітичних систем.

Для реалізації механізмів збору фінансових новин з веб-джерел використовуються бібліотеки для виконання HTTP-запитів та розбору HTML-структур. Такий підхід дозволяє здійснювати як роботу зі стандартизованими новинними стрічками, так і прямий парсинг веб-сторінок фінансових ресурсів [19], [20]. Обрані інструменти є кросплатформними, стабільними та широко застосовуються у промислових проєктах.

Попередня обробка текстів та базові NLP-операції реалізуються з використанням бібліотеки spaCy, яка надає готові мовні pipeline для токенізації, лематизації та виділення іменованих сутностей. Використання spaCy дозволяє стандартизувати процес обробки тексту та забезпечити коректну роботу з великими корпусами новинних даних [16]–[18].

Семантичний аналіз текстів у межах методики ґрунтується на використанні попередньо навчених трансформерних моделей, інтегрованих через бібліотеку Sentence-Transformers. Даний інструмент дозволяє отримувати контекстні векторні представлення речень і документів без необхідності навчання моделей з нуля, що відповідає концепції роботи та суттєво знижує складність реалізації [4]. Для роботи з трансформерними моделями також використовується екосистема Hugging Face, яка забезпечує зручний доступ до готових моделей та їх уніфіковану інтеграцію [27].

Для побудови прикладного веб-сервісу та взаємодії між клієнтом і серверною частиною застосовано фреймворк Django у поєднанні з Django REST Framework.

Такий вибір обумовлений можливістю швидкого створення API, чітким розділенням логіки додатку та підтримкою масштабованих архітектур. REST-інтерфейс забезпечує зручний доступ до функціональності системи, зокрема ініціації збору новин, запуску аналізу та отримання результатів [29], [30].

Зберігання статей та результатів аналізу реалізується з використанням серверного сховища даних, структура якого адаптована до збереження як первинних текстів, так і похідних семантичних ознак. Такий підхід дозволяє розділяти дані різних етапів обробки та повторно використовувати їх у подальших аналітичних операціях.

Отже, обраний технологічний стек на основі Python, сучасних NLP-бібліотек та веб-фреймворків забезпечує ефективну реалізацію запропонованої методики автоматизації парсингу та семантичного аналізу фінансових новин. Його використання є обґрунтованим як з практичної, так і з науково-дослідної точки зору, оскільки поєднує гнучкість, масштабованість та відповідність сучасним підходам до аналізу текстової інформації.

3.4. Реалізація системи автоматизованого аналізу

Реалізація системи автоматизованого аналізу фінансових новин виконана відповідно до розробленої методики та обраного технологічного стеку. Система побудована як модульний веб-орієнтований сервіс, у якому кожен функціональний компонент відповідає за окремий етап обробки новинних даних. Такий підхід забезпечує чітке розмежування відповідальностей, спрощує підтримку коду та дозволяє масштабувати систему залежно від обсягу оброблюваних даних.

Серверна частина системи реалізована у вигляді REST-сервісу, який надає зовнішнім клієнтам доступ до основних функцій: ініціації збору новин, запуску семантичного аналізу та отримання результатів. Логіка обробки даних організована у вигляді послідовного конвеєра, що відображає етапи, описані у попередніх підрозділах. Кожен етап може виконуватися незалежно, що дозволяє асинхронно обробляти великі обсяги новин та уникати блокування системи.

Модуль збору новин відповідає за отримання фінансових публікацій з визначених джерел. Він взаємодіє з оркестратором збору, який керує процесом обходу джерел, контролює кількість оброблюваних записів та забезпечує повторюваність виконання. На цьому етапі формується уніфіковане представлення статей, що включає основний текст, заголовок, часову мітку та метадані. Отримані статті зберігаються у сховищі з відповідним статусом, що відображає стадію їх обробки.

Модуль семантичного аналізу активується для статей, які готові до обробки. У межах цього модуля виконується попередня підготовка тексту, включаючи очищення, нормалізацію та видалення нерелевантних елементів. Далі застосовуються готові NLP-моделі для отримання семантичних ознак: ключових слів, тематичних характеристик, загальної та контекстної тональності. Для кожної статті обчислюється векторне представлення, яке використовується як універсальна форма семантичного опису тексту [4].

Окремим компонентом системи є модуль обчислення семантичної подібності. Він використовує збережені векторні представлення статей для пошуку схожих новин за допомогою метрик подібності. Це дозволяє виявляти близькі за змістом публікації, агрегувати новини за подіями та зменшувати інформаційний шум у новинному потоці [26]. Результати подібності зберігаються у вигляді зв’язків між статтями, що можуть бути використані для подальшої аналітики або візуалізації.

Усі результати семантичного аналізу та допоміжних обчислень зберігаються у структурованому вигляді у сховищі аналізу. Така організація даних дозволяє повторно використовувати результати без необхідності повторного запуску обчислювально складних процедур. Крім того, збереження окремих етапів обробки забезпечує можливість розширення системи, наприклад шляхом додавання нових типів аналізу або інтеграції з іншими аналітичними сервісами.

Таким чином, реалізована система автоматизованого аналізу фінансових новин є практичним втіленням запропонованої методики. Вона поєднує автоматизований парсинг, семантичний аналіз на основі готових NLP-моделей та

сервісну архітектуру, що забезпечує гнучкість, масштабованість і придатність до використання у прикладних інформаційно-аналітичних системах.

3.5. Опис інтерфейсу розробленого застосунку

Інтерфейс розробленого застосунку виконує роль користувацького рівня доступу до функціональності системи автоматизованого парсингу та семантичного аналізу фінансових новин. Його основне призначення полягає у забезпеченні зручної взаємодії з результатами аналізу, візуалізації оброблених даних та керуванні процесами збору й аналізу новин без необхідності безпосереднього доступу до серверної логіки або бази даних.

Інтерфейс побудований за принципом клієнт–серверної взаємодії та працює через REST API, що дозволяє відокремити візуальний рівень від обчислювальних модулів системи. Такий підхід забезпечує гнучкість у подальшому розвитку застосунку та можливість інтеграції з іншими клієнтськими рішеннями.

Початковою екранною формою є сторінка списку новин (Newsroom), на якій відображається агрегований перелік зібраних фінансових статей у вигляді таблиці з основними атрибутами новини: ідентифікатором, заголовком, джерелом, датою публікації та статусом обробки. Статус *ANALYZED* свідчить про завершення повного циклу семантичної обробки відповідної публікації.

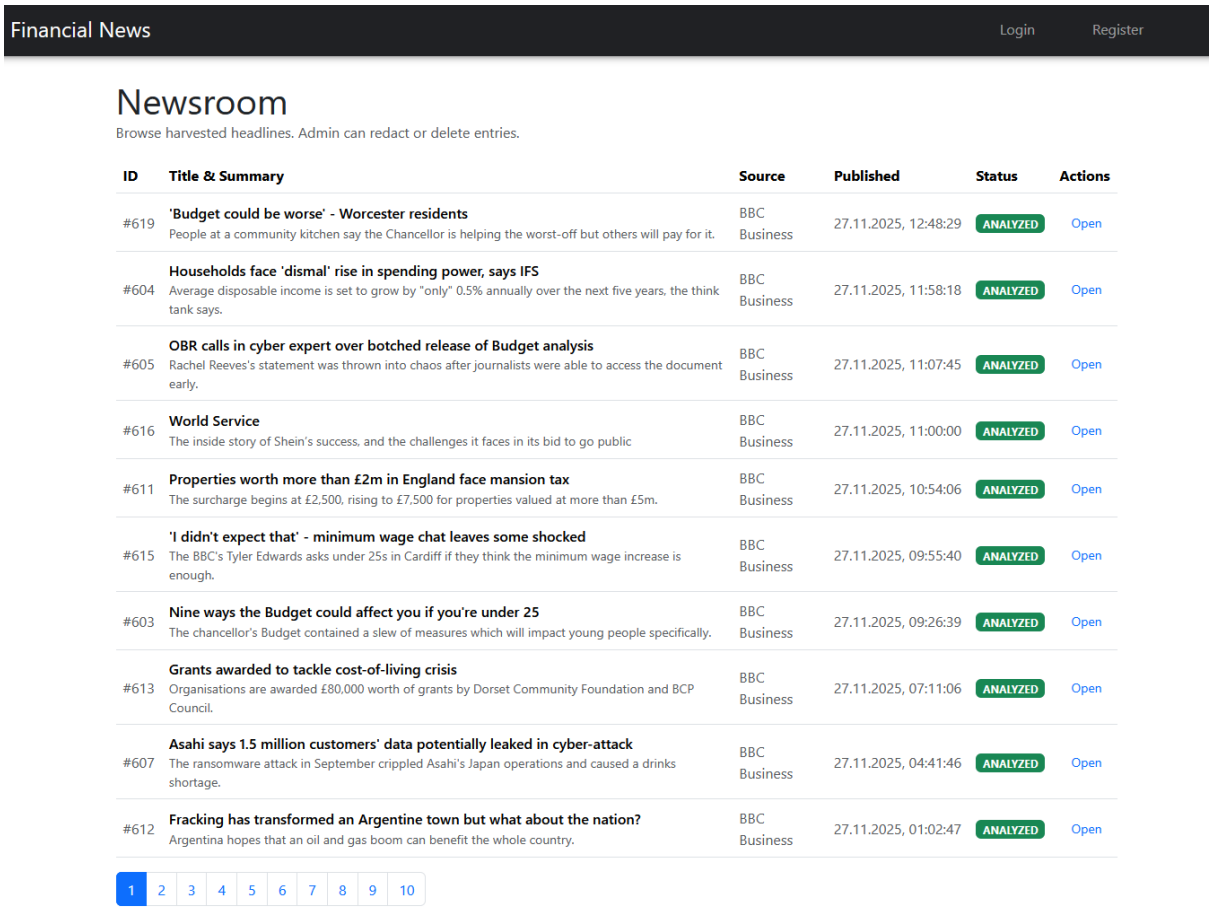


Рис. 3.2. Сторінка списку фінансових новин

Табличне представлення дозволяє швидко переглядати великі обсяги новин, виконувати навігацію між сторінками списку за допомогою механізму пагінації та переходити до детального перегляду вибраної статті через дію *Open*.

Наступною екранною формою є сторінка детального перегляду новини, яка містить розширену інформацію про обрану публікацію. На цій сторінці відображається заголовок статті, джерело, дата та час публікації, статус аналізу, а також посилання на оригінальний матеріал. Центральним елементом є повний текст новини, що дозволяє зіставити вихідний контент із результатами автоматичного аналізу.

Рис. 3.3. Сторінка детального перегляду фінансової новини

Окремим інформаційним блоком на сторінці детального перегляду представлено узагальнений AI-insight, який містить ключові результати семантичної обробки: оцінку впливу (impact), основні тематичні мітки (event tags) та коротке автоматично сформоване резюме змісту новини. Такий блок дозволяє швидко оцінити загальний характер публікації без необхідності повного прочитання тексту.

Важливою складовою інтерфейсу є секція Similar articles, у якій наведено перелік семантично подібних новин із зазначенням коефіцієнтів схожості.

Подібність обчислюється на основі контекстних ембеддингів і косинусної міри, що забезпечує виявлення дублікатів та близьких за змістом публікацій з різних джерел.

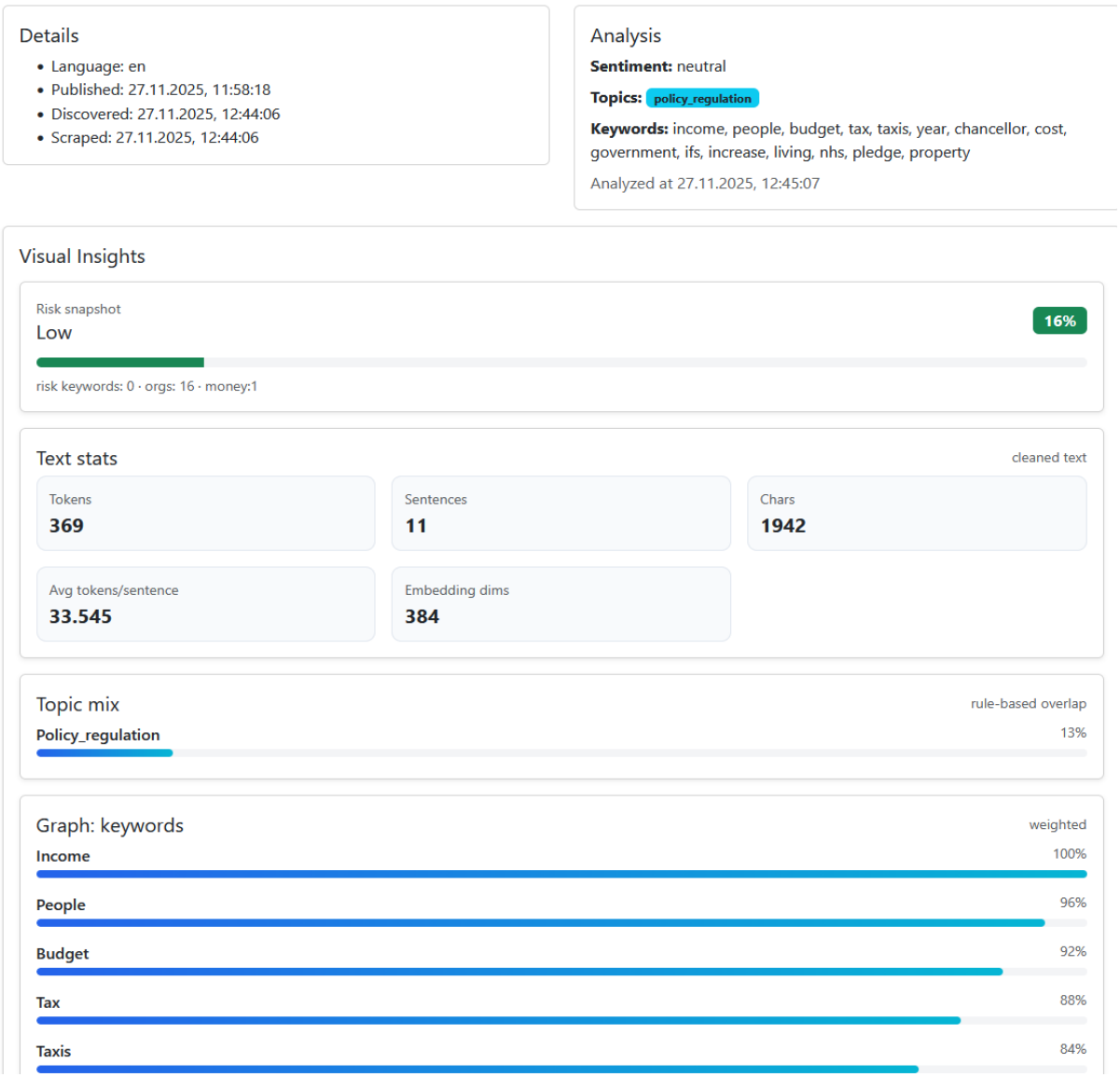


Рис. 3.4. Візуальні показники та текстові статистики фінансової новини

Окремо відображається тематичний розподіл (topic mix), який кількісно характеризує домінування основної теми у тексті новини на основі rule-based overlap. Це дозволяє оцінити, наскільки однорідним є зміст публікації з точки зору тематики.

У розділі Keywords ключові слова подано у вигляді тегів та графічних індикаторів їхньої ваги, що спрощує візуальне сприйняття та дозволяє швидко ідентифікувати основні поняття, навколо яких побудовано текст.

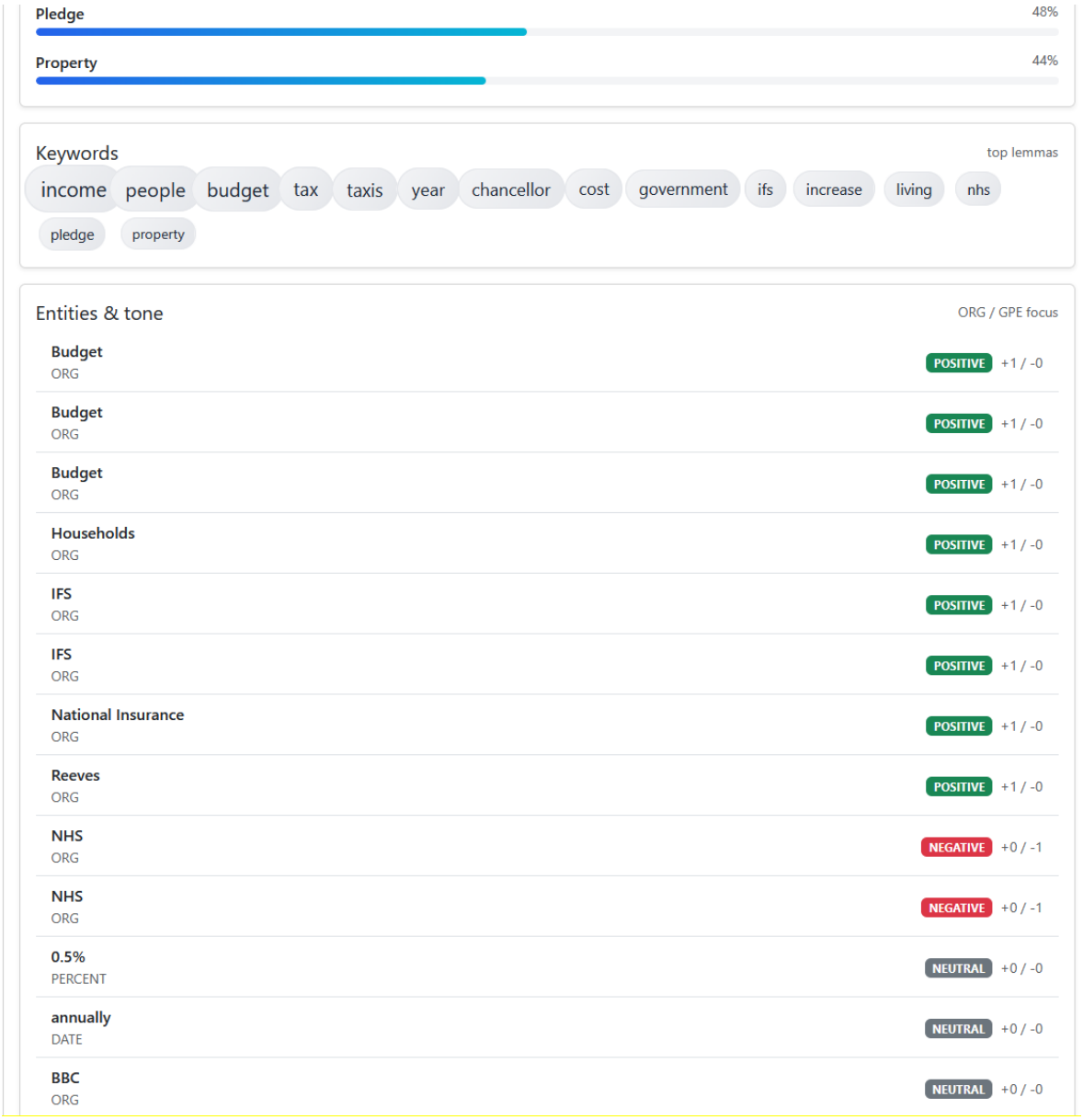


Рис. 3.5. Візуалізація ключових слів та їх відносної значущості, виявлені іменовані сутності та оцінка тональності їх згадування

Завершальним елементом сторінки є секція Entities & tone, у якій відображаються виявлені іменовані сутності (організації, географічні об’єкти, числові показники) разом із локальною оцінкою тональності згадування. Це дозволяє аналізувати ставлення тексту до конкретних економічних суб’єктів та оцінювати репутаційні аспекти фінансових новин.

З точки зору користувацького досвіду інтерфейс орієнтований на простоту та інформативність. Він не перевантажений технічними деталями реалізації NLP-

алгоритмів, а зосереджується на поданні результатів у структурованому та зрозумілому вигляді. Це робить застосунок придатним як для дослідницького використання, так і для прикладного аналізу фінансових новин.

Таким чином, інтерфейс розробленого застосунку є логічним завершенням запропонованої методики автоматизації парсингу та семантичного аналізу. Він забезпечує ефективний доступ до результатів обробки фінансових новин і виступає важливим елементом інтеграції системи у практичні інформаційно-аналітичні сценарії.

ВИСНОВКИ

Досліджено сучасні підходи до автоматизації парсингу та семантичного аналізу фінансових новин, а також проаналізовано роль текстових новин як джерела неструктурованих даних у фінансово-аналітичних системах. Розглянуто основні методи збору новинної інформації, підходи до попередньої обробки текстів і семантичної інтерпретації, що дозволило виявити ключові обмеження ручного аналізу та необхідність застосування автоматизованих NLP-рішень для роботи з великими обсягами фінансового контенту.

Проаналізовано існуючі програмні аналоги та платформи для аналізу фінансових новин, визначено їх функціональні можливості та обмеження. Встановлено, що більшість наявних рішень є або закритими комерційними системами з обмеженою гнучкістю, або універсальними інструментами обробки тексту без глибокої орієнтації на фінансову тематику. Це підтвердило доцільність розробки власної методики, орієнтованої на автоматизацію парсингу та семантичного аналізу фінансових новин із можливістю адаптації під конкретні аналітичні задачі.

Розроблено методику автоматизації парсингу та семантичного аналізу фінансових новин, яка базується на використанні готових попередньо навчених NLP-моделей і реалізована у вигляді послідовного обробного конвеєра. Запропонована методика охоплює етапи збору новин з відкритих джерел, попередньої обробки тексту, формування семантичних векторних представлень, витягу ключових слів, тематичної сегментації та аналізу семантичної подібності між статтями. Такий підхід дозволяє структурувати новинний потік і зменшити інформаційний шум.

Реалізовано програмну систему автоматизованого аналізу фінансових новин на основі обраного технологічного стеку, що включає Python, сучасні NLP-бібліотеки та веб-орієнтовану архітектуру. Система забезпечує асинхронний збір і обробку новин, збереження результатів семантичного аналізу та надання доступу до них через REST-інтерфейс. Розроблений користувацький інтерфейс спрощує

взаємодію з результатами аналізу та дозволяє наочно переглядати семантичні характеристики фінансових новин.

Підвищено рівень ефективності роботи з фінансовими новинними потоками за рахунок автоматизації процесів парсингу та семантичної обробки текстів. Запропонована методика дозволяє зменшити часові витрати на аналіз фінансових новин, підвищити об’єктивність інтерпретації текстових даних та створює основу для подальшого розвитку інформаційно-аналітичних систем у фінансовій сфері, зокрема шляхом розширення набору семантичних ознак або інтеграції з іншими джерелами даних.

Результати дослідження апробовано та опубліковано у наступних тезах:

1. Мартиненко О.В., Золотухіна О.А., Розробка методу витягу семантично значущої інформації з фінансових новин на основі парсингу та аналізу текстів. Всеукраїнська науково-практична конференція «Застосування програмного забезпечення в ІКТ», 24 квітня 2025 р., Київ, Державний університет інформативно-комунікаційних технологій. Збірник тез. К.: ДУІКТ,2025.С.489.

2. Мартиненко О.В., Золотухіна О.А., Огляд методів обробки текстової інформації в контексті методики автоматизації парсингу фінансових новин та їх семантичного аналізу для витягу ключових тем. Всеукраїнська науково-практична конференція «Застосування програмного забезпечення в ІКТ», 24 квітня 2025 р., Київ, Державний університет інформативно-комунікаційних технологій. Збірник тез. К.: ДУІКТ,2025.С. 591.

ПЕРЕЛІК ПОСИЛАНЬ

1. Vaswani A., Shazeer N., Parmar N. et al. Attention Is All You Need. NeurIPS, 2017. URL: https://arxiv.org/abs/1706.03762 (дата звернення: 16.12.2025).

2. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv, 2018. URL: https://arxiv.org/abs/1810.04805 (дата звернення: 16.12.2025).

3. Reimers N., Gurevych I. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. EMNLP, 2020. URL: https://arxiv.org/abs/2004.09813 (дата звернення: 16.12.2025).

4. Sentence-Transformers. SentenceTransformers Documentation. URL: https://sbert.net/ (дата звернення: 16.12.2025).

5. Araci D. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. arXiv, 2019. URL: https://arxiv.org/abs/1908.10063 (дата звернення: 16.12.2025).

6. Loughran T., McDonald B. When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. *Journal of Finance*, 2011. URL: https://www.uts.edu.au/globalassets/sites/default/files/adg_cons2015_loughran-mcdonald-je-2011.pdf (дата звернення: 16.12.2025).

7. Loughran-McDonald. Master Dictionary (Sentiment Word Lists). URL: https://sraf.nd.edu/loughranmcdonald-master-dictionary/ (дата звернення: 16.12.2025).

8. Malo P., Sinha A., Takala P., Korhonen P., Wallenius J. Good Debt or Bad Debt: Detecting Semantic Orientations in Economic Texts. arXiv, 2013/2014. URL: https://arxiv.org/abs/1307.5336 (дата звернення: 16.12.2025).

9. Du K. та ін. Financial Sentiment Analysis: Techniques and Applications (survey). 2024. URL: https://w.sentic.net/financial-sentiment-analysis-survey.pdf (дата звернення: 16.12.2025).

11. Inserte P. R. та ін. Large Language Model Adaptation for Financial Sentiment Analysis. ACL Anthology (FinNLP), 2023. URL: https://aclanthology.org/2023.finnlp-2.1.pdf (дата звернення: 16.12.2025).

12. Yathongkhum W. Hybrid approach for economic and financial news classification. *Information Development*, 2024. URL: https://journals.sagepub.com/doi/10.3233/IDA-237373 (дата звернення: 16.12.2025).

13. Blei D. M., Ng A. Y., Jordan M. I. Latent Dirichlet Allocation. *JMLR*, 2003. URL: https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf (дата звернення: 16.12.2025).

14. Mihalcea R., Tarau P. TextRank: Bringing Order into Text. EMNLP, 2004. URL: https://aclanthology.org/W04-3252/ (дата звернення: 16.12.2025).

Grootendorst M. KeyBERT: Minimal keyword extraction with BERT (репозиторій). URL: https://github.com/MaartenGr/KeyBERT (дата звернення: 16.12.2025).

15. Grootendorst M. Keyword Extraction with BERT (KeyBERT). URL: https://maartengrootendorst.com/blog/keybert/ (дата звернення: 16.12.2025).

16. spaCy. Industrial-strength Natural Language Processing in Python (офіційний сайт). URL: https://spacy.io/ (дата звернення: 16.12.2025).

17. spaCy. Language Processing Pipelines (tokenization/lemmatization/NER pipeline). URL: https://spacy.io/usage/processing-pipelines (дата звернення: 16.12.2025).

18. spaCy. Trained Models & Pipelines. URL: https://spacy.io/models (дата звернення: 16.12.2025).

19. Beautiful Soup. Beautiful Soup 4 Documentation. URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (дата звернення: 16.12.2025).

20. Requests. Requests: HTTP for Humans (Documentation). URL: https://requests.readthedocs.io/ (дата звернення: 16.12.2025).

21. IETF. RFC 4287: The Atom Syndication Format. URL: https://datatracker.ietf.org/doc/html/rfc4287 (дата звернення: 16.12.2025).

22. RSS Advisory Board. RSS 2.0 Specification. URL: https://www.rssboard.org/rss-specification (дата звернення: 16.12.2025).

23. Broder A. Z. On the Resemblance and Containment of Documents. 1997. URL: https://www.cs.princeton.edu/courses/archive/spring13/cos598C/broder97resemblance.pdf (дата звернення: 16.12.2025).

24. Broder A. Z. Identifying and Filtering Near-Duplicate Documents. 2007 (огляд/методики near-duplicate). URL: https://cs.brown.edu/courses/cs253/papers/nearduplicate.pdf (дата звернення: 16.12.2025).

25. scikit-learn. SGDClassifier — документація. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html (дата звернення: 16.12.2025).

26. scikit-learn. cosine_similarity — документація. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html (дата звернення: 16.12.2025).

27. Hugging Face. Transformers Documentation. URL: https://huggingface.co/docs/transformers/en/index (дата звернення: 16.12.2025).

28. Hugging Face Datasets. financial_phrasebank (dataset card). URL: https://huggingface.co/datasets/takala/financial_phrasebank (дата звернення: 16.12.2025).

29. Django. Django documentation. URL: https://docs.djangoproject.com/en/6.0/ (дата звернення: 16.12.2025).

30. Django REST framework. Official documentation (Home). URL: https://www.django-rest-framework.org/ (дата звернення: 16.12.2025).

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Магістерська робота

«Методика автоматизації парсингу фінансових новин на основі їх семантичного аналізу»

Виконав: студент групи ПДМ-62 Олексій МАРТИНЕНКО

Керівник: канд. техн. наук, професор кафедри ІТ Максим КУКЛІНСЬКИЙ

Київ - 2025

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: підвищення точності та автоматизація процесу опрацювання фінансових новин на основі методів семантичного аналізу тексту.

Об'єкт дослідження: процес автоматизованого аналізу фінансових текстових новин.

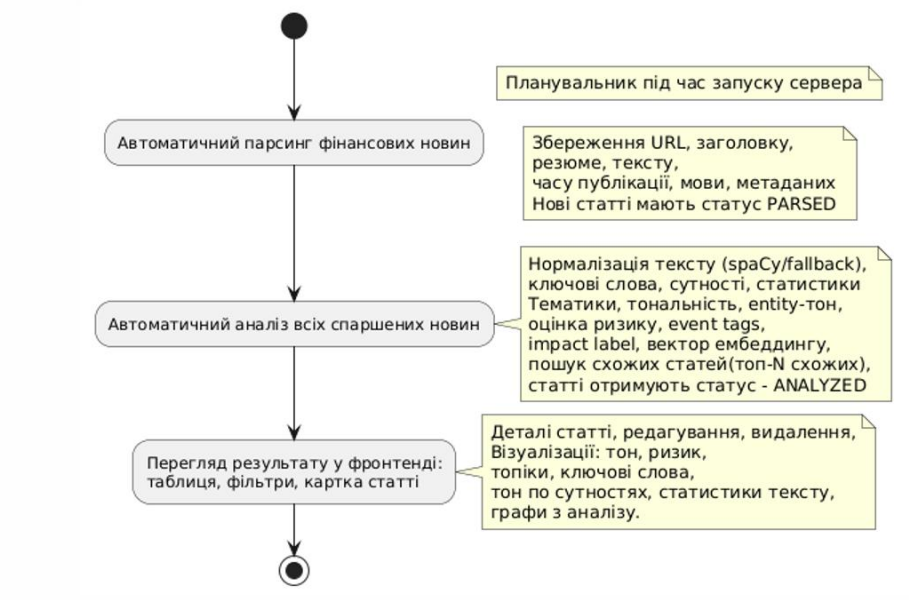
Предмет дослідження: комбінований метод для опрацювання фінансових новин, що поєднує автоматизований парсинг інформаційних джерел та семантичний аналіз тексту на основі сучасних моделей природної мови.

АКТУАЛЬНІСТЬ РОБОТИ

Метод	Переваги	Ключові недоліки
Ручний моніторинг та експертний аналіз новин	Висока глибина розуміння контексту, урахування специфіки ринку, можливість складних інтерпретацій	Низька масштабованість, велика трудомісткість, людський фактор, затримки в отриманні аналітики
Класичні підходи до автоматизованого аналізу тексту (ключові слова, TF-IDF + SVM/LogReg)	Висока швидкість, простота реалізації, відносна інтерпретованість ознак	"Semantic gap": ігнорування глибокого контексту, чутливість до формулювань та синонімів, велика кількість хибних спрацьовувань/пропусків, складність адаптації під нові джерела та мови
Сучасні глибокі моделі (BERT/FinBERT, LLM-API)	Найвища точність семантичного аналізу, розуміння контексту, можливість одночасного визначення тем, тональності, сутностей	Висока ресурсомісткість (GPU/хмарні сервіси), модель як "чорна скринька", складність вбудовування в повний конвеєр "парсинг → очищення → аналіз → агреговані індикатори", відсутність доменно-спеціалізованої методики для фінансових новин

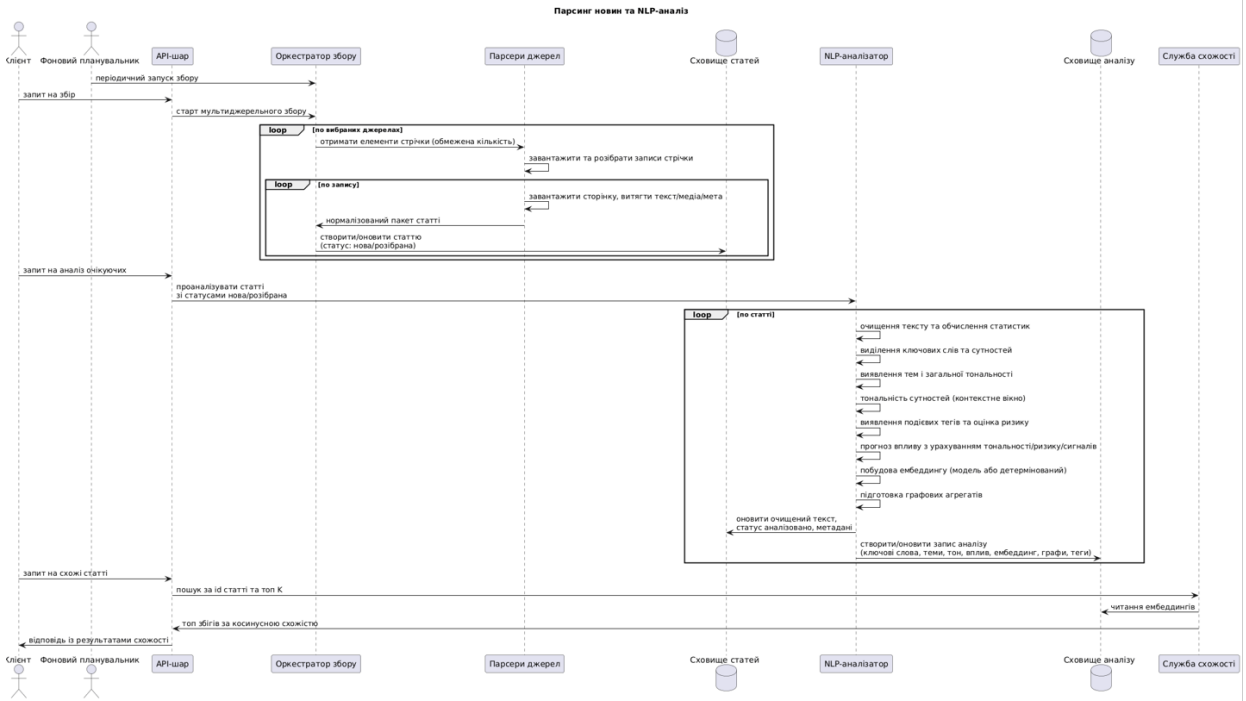
3

СХЕМА РОБОТИ МЕТОДУ



4

СХЕМА РОБОТИ МЕТОДУ(2)



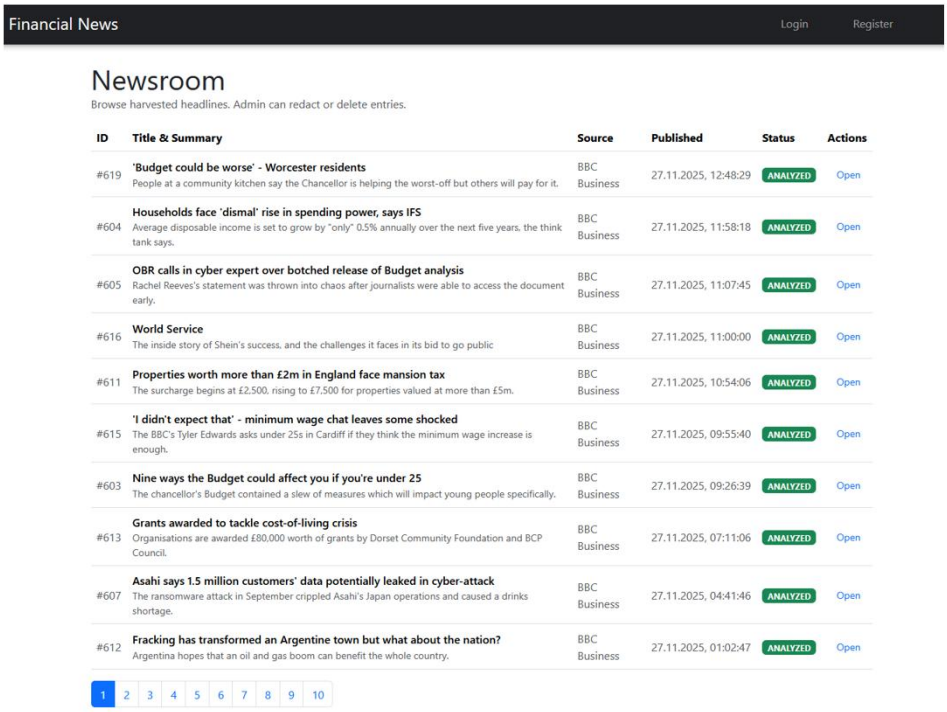
ЕТАПИ СЕМАТИЧНОГО АНАЛІЗУ

Етап	Технічні деталі	Вихідний результат
1. Нормалізація тексту	Очищення тексту від зайвих символів, уніфікація пробілів, приведення до нижнього регістру, підготовка до лінгвістичного аналізу	Нормалізований текст, готовий до подальшої обробки
2. Лематизація та ключові слова	Аналіз слів, зведення їх до початкової форми, відбір інформативних іменників та власних назв, частотний аналіз	Набір ключових лем, що описують зміст новини
3. Виділення іменованих сутностей	Виявлення компаній, країн, валют, відсотків, грошових сум та інших важливих сутностей у тексті	Список сутностей з їх типами та позиціями в тексті
4. Статистичний опис тексту	Підрахунок токенів, речень, символів, оцінка середньої довжини речення та інших базових метрик	Статистичні характеристики тексту (обсяг, структура, щільність інформації)

ЕТАПИ СЕМАТИЧНОГО АНАЛІЗУ(2)

Етап	Технічні деталі	Вихідний результат
5. Визначення тематик новини	Порівняння ключових слів із заздалегідь заданими словниками тематик (ринок, макроекономіка, компанії, валюти, сировина, регуляція тощо)	Перелік тематик із оцінками релевантності для кожної новини
6. Оцінка тональності	Аналіз позитивних і негативних слів для визначення загального тону тексту та локальної тональності довкола ключових сутностей (компаній, країн)	Загальна тональність новини та тональність ключових сутностей
7. Оцінка ризику та прогноз впливу	Облік наявності ризикової лексики, фінансових сутностей і співвідношення позитивних/негативних сигналів для оцінки ризику та очікуваного впливу	Рівень ризику (low/medium/high) та прогнозований вплив новини (positive/neutral/negative)
8. Семантичне подання та дані для візуалізації	Побудова векторного подання тексту, агрегування тематик, ключових слів, тональності, ризику та статистики для подальших графіків	Семантичний вектор новини та структуровані дані для візуалізації й збереження результатів аналізу

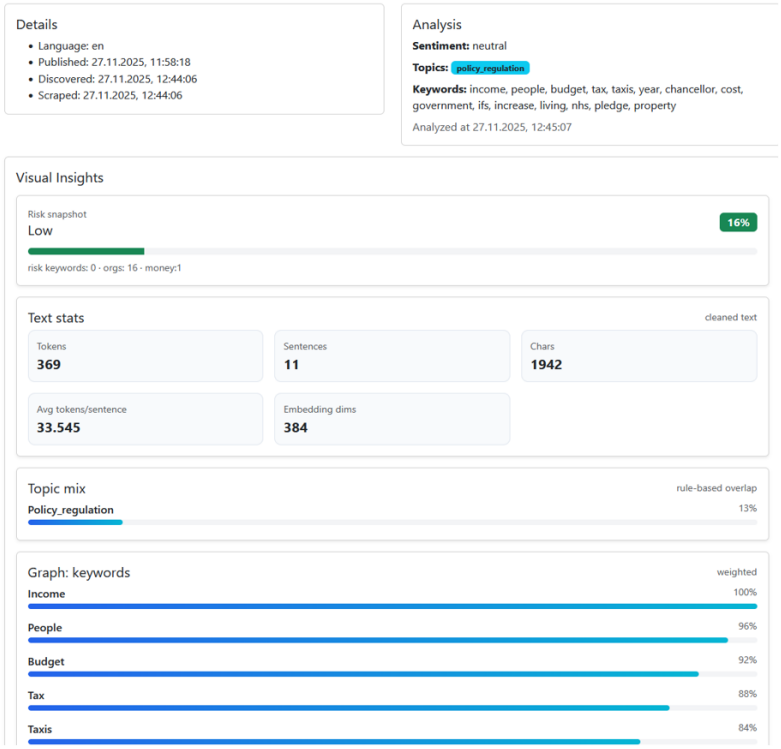
ВЕБЗАСТОСУНОК АНАЛІЗУ ФІНАНСОВИХ НОВИН



ВЕБЗАСТОСУНОК АНАЛІЗУ ФІНАНСОВИХ НОВИН

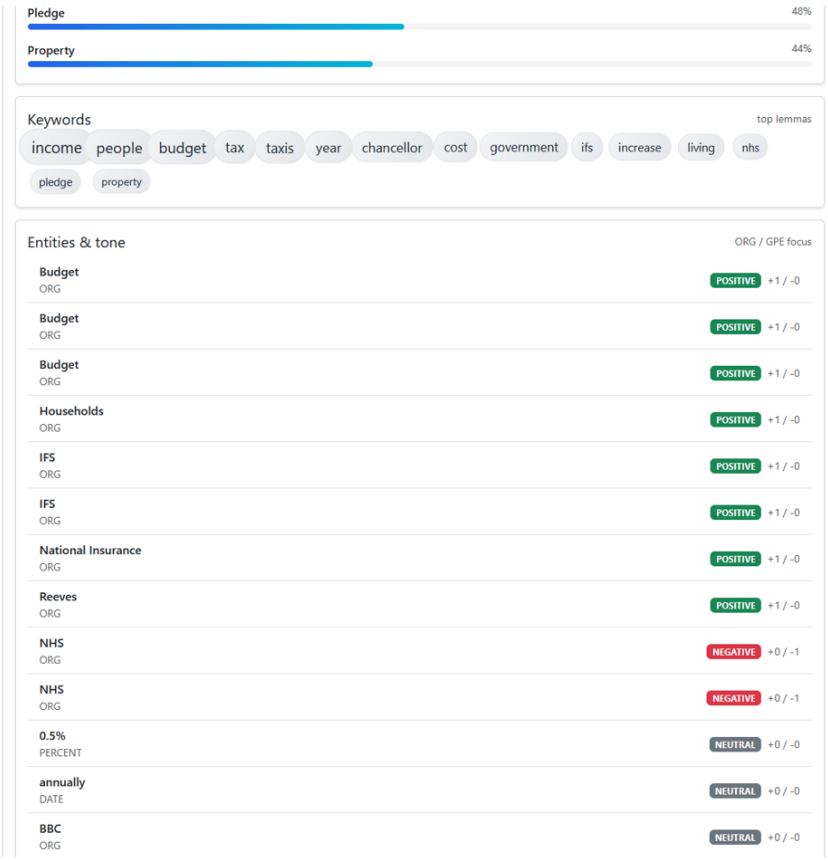
9

ВЕБЗАСТОСУНОК АНАЛІЗУ ФІНАНСОВИХ НОВИН



10

ВЕБЗАСТОСУНОК АНАЛІЗУ ФІНАНСОВИХ НОВИН



11

ВИСНОВКИ

1. Досліджено та проаналізовано сучасні підходи до автоматизованої обробки фінансових новин на основі методів обробки природної мови. Виявлено обмеження ручового аналізу та базових лексичних підходів, зокрема залежність від джерела, суб'єктивність оцінок і недостатню масштабованість при зростанні обсягів новинних даних.
2. Сформульовано вимоги та розроблено методику автоматизації парсингу фінансових новин, що поєднує нормалізацію тексту, лематизацію, витяг ключових слів і іменованих сутностей, тематичну класифікацію та визначення тональності, доповнені обчисленням базових текстових статистик.
3. Реалізовано багатоступеневий семантичний аналіз, який включає правило-орієнтоване визначення тематик і сентименту, локальну оцінку тональності для компаній і країн на основі контекстного вікна, а також оцінку ризику фінансових новин за лексичними ознаками та типами сутностей.
4. Запропоновано використання векторних семантичних подань для аналізу схожості фінансових новин, що дозволяє виявляти пов'язані публікації, групувати матеріали за змістом та підвищувати узгодженість аналітичних результатів у межах новинного потоку.
5. Створено програмний прототип веб-сервісу, який реалізує запропоновану методику та забезпечує збереження, агрегацію й візуалізацію результатів аналізу. Проведені експериментальні дослідження підтвердили працездатність підходу та показали підвищення стабільності оцінки тематик і ризиків фінансових новин порівняно з базовими лексичними методами.

12

ПУБЛІКАЦІЇ ТА АПРОБАЦІЯ РОБОТИ

Тези доповідей:

1. 1Мартиненко О.В., Золотухіна О.А., Розробка методу витягу семантично значущої інформації з фінансових новин на основі парсингу та аналізу текстів. Всеукраїнська науково-практична конференція «Застосування програмного забезпечення в ІКТ», 24 квітня 2025 р., Київ, Державний університет інформативно-комунікаційних технологій. Збірник тез. К.: ДУІКТ,2025.С.489.

2. Мартиненко О.В., Золотухіна О.А., Огляд методів обробки текстової інформації в контексті методики автоматизації парсингу фінансових новин та їх семантичного аналізу для витягу ключових тем. Всеукраїнська науково-практична конференція «Застосування програмного забезпечення в ІКТ», 24 квітня 2025 р., Київ, Державний університет інформативно-комунікаційних технологій. Збірник тез. К.: ДУІКТ,2025.С. 591.

ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ

```
"""
Multi-source financial news scraping.
"""

import logging
import re
from dataclasses import dataclass
from datetime import datetime
from email.utils import parsedate_to_datetime
from typing import Any, Callable, Dict, Iterable, List, Optional
from xml.etree import ElementTree

import requests
from bs4 import BeautifulSoup
from django.utils import timezone

from .models import Article, ArticleStatus

logger = logging.getLogger(__name__)

DEFAULT_HEADERS = {
    "User-Agent": (
        "Mozilla/5.0 (Windows NT 10.0; Win64; x64) "
        "AppleWebKit/537.36 (KHTML, like Gecko) "
        "Chrome/124.0 Safari/537.36"
    )
}

REQUEST_TIMEOUT = 15

session = requests.Session()
session.headers.update(DEFAULT_HEADERS)


@dataclass
class ScrapedArticle:
    source_name: str
    url: str
    title: str
    summary: str = ""
    body: str = ""
    raw_html: str = ""
    published_at: Optional[datetime] = None
    image_url: Optional[str] = None
    language: str = "en"
    metadata: Optional[Dict[str, Any]] = None
```

```
@dataclass(frozen=True)
class NewsSource:
    key: str
    name: str
    scraper: Callable[[int], Iterable[ScrapedArticle]]


def _strip_html(text: str) -> str:
    soup = BeautifulSoup(text or "", "html.parser")
    return soup.get_text(" ", strip=True)


def _parse_datetime(value: Optional[str]) -> Optional[datetime]:
    if not value:
        return None

    try:
        return parsedate_to_datetime(value)
    except Exception:
        pass

    try:
        return datetime.fromisoformat(value.replace("Z", "+00:00"))
    except Exception:
        return None


def _aware(dt: Optional[datetime]) -> Optional[datetime]:
    if not dt:
        return None
    return timezone.make_aware(dt) if timezone.is_naive(dt) else dt


def _fetch(url: str) -> requests.Response:
    response = session.get(url, timeout=REQUEST_TIMEOUT)
    response.raise_for_status()
    return response


def _persist_article(scraped: ScrapedArticle, stats: Dict[str, Any]) -> None:
    stats["checked"] += 1

    published_at = _aware(scraped.published_at) if scraped.published_at else None
    metadata = scraped.metadata.copy() if scraped.metadata else {}

    existing = Article.objects.filter(url=scraped.url).first()
    if existing:
        if existing.status in {ArticleStatus.PARSED, ArticleStatus.ANALYZED} and not
scraped.body:
            stats["skipped_existing"] += 1
            return
```

```
        existing.title = scraped.title or existing.title
        existing.summary = scraped.summary or existing.summary
        if scraped.body:
            existing.parsed_text = scraped.body
            existing.scraped_at = timezone.now()
            if existing.status != ArticleStatus.ANALYZED:
                existing.status = ArticleStatus.PARSED
        existing.raw_html = scraped.raw_html or existing.raw_html
        if published_at and not existing.published_at:
            existing.published_at = published_at
        existing.language = scraped.language or existing.language
        meta = existing.scrape_metadata or {}
        meta.update(metadata)
        existing.scrape_metadata = meta
        existing.save()
        stats["updated"] += 1
        return

    Article.objects.create(
        source_name=scraped.source_name,
        url=scraped.url,
        title=scraped.title,
        summary=scraped.summary,
        parsed_text=scraped.body or scraped.summary or scraped.title,
        raw_html=scraped.raw_html,
        published_at=published_at,
        discovered_at=timezone.now(),
        scraped_at=timezone.now() if scraped.body else None,
        status=ArticleStatus.PARSED if scraped.body else ArticleStatus.NEW,
        language=scraped.language,
        scrape_metadata=metadata or None,
    )
    stats["created"] += 1


# ---------------------------------------------------------------------------
# BBC Business
# ---------------------------------------------------------------------------

BBC_RSS_URL = "https://feeds.bbci.co.uk/news/business/rss.xml"


def _parse_bbc_article(url: str) -> tuple[str, str, str, Optional[str]]:
    response = _fetch(url)
    soup = BeautifulSoup(response.text, "html.parser")

    title_el = soup.find("h1")
    title = title_el.get_text(strip=True) if title_el else ""
```

```
    paragraphs = []
    for block in soup.select("article div[data-component='text-block']"):
        text = block.get_text(" ", strip=True)
        if text:
            paragraphs.append(text)
    body = "\n\n".join(paragraphs)

    image_url = None
    image_meta = soup.find("meta", property="og:image")
    if image_meta and image_meta.get("content"):
        image_url = image_meta["content"]

    return title, body, response.text, image_url or None


def scrape_bbc_business(max_items: int) -> Iterable[ScrapedArticle]:
    rss_response = _fetch(BBC_RSS_URL)
    feed = ElementTree.fromstring(rss_response.content)
    for item in feed.findall("./channel/item")[:max_items]:
        link = item.findtext("link")
        if not link:
            continue

        title = _strip_html(item.findtext("title") or "")
        summary = _strip_html(item.findtext("description") or "")
        published_at = _parse_datetime(item.findtext("pubDate"))

        article_title, body, raw_html, image_url = _parse_bbc_article(link)
        yield ScrapedArticle(
            source_name="BBC Business",
            url=link,
            title=article_title or title,
            summary=summary or article_title or title,
            body=body,
            raw_html=raw_html,
            published_at=published_at,
            image_url=image_url,
            metadata={"rss_title": title, "rss_summary": summary},
        )


# ---------------------------------------------------------------------------
# CNBC
# ---------------------------------------------------------------------------

CNBC_RSS_URL = "https://www.cnbc.com/id/10000664/device/rss/rss.html"  # Finance
section
```

```
def _parse_cnbc_article(url: str) -> tuple[str, str, str, Optional[str],
Optional[datetime]]:
    response = _fetch(url)
    soup = BeautifulSoup(response.text, "html.parser")

    title_el = soup.find("h1")
    title = title_el.get_text(strip=True) if title_el else ""

    body = ""
    body_el = soup.select_one("div[itemprop='articleBody']") or
soup.select_one("div.ArticleBody-articleBody")
    if body_el:
        parts = [p.get_text(" ", strip=True) for p in body_el.find_all("p")]
        body = "\n\n".join(p for p in parts if p)

    image_url = None
    image_meta = soup.find("meta", property="og:image")
    if image_meta and image_meta.get("content"):
        image_url = image_meta["content"]

    published_at = None
    published_meta = soup.find("meta", {"itemprop": "datePublished"})
    if published_meta and published_meta.get("content"):
        published_at = _parse_datetime(published_meta["content"])

    return title, body, response.text, image_url, published_at


def scrape_cnbc_finance(max_items: int) -> Iterable[ScrapedArticle]:
    rss_response = _fetch(CNBC_RSS_URL)
    feed = ElementTree.fromstring(rss_response.content)

    for item in feed.findall("./channel/item")[:max_items]:
        link = item.findtext("link")
        if not link:
            continue

        title = _strip_html(item.findtext("title") or "")
        summary = _strip_html(item.findtext("description") or "")
        published_at = _parse_datetime(item.findtext("pubDate"))

        article_title, body, raw_html, image_url, published_from_page =
_parse_cnbc_article(link)
        yield ScrapedArticle(
            source_name="CNBC Finance",
            url=link,
            title=article_title or title,
            summary=summary or article_title or title,
            body=body,
            raw_html=raw_html,
```

```
            published_at=published_from_page or published_at,
            image_url=image_url,
            metadata={"rss_title": title, "rss_summary": summary},
        )


# ---------------------------------------------------------------------------
# Public API
# ---------------------------------------------------------------------------

AVAILABLE_SOURCES: Dict[str, NewsSource] = {
    "bbc": NewsSource(key="bbc", name="BBC Business", scraper=scrape_bbc_business),
    "cnbc": NewsSource(key="cnbc", name="CNBC Finance", scraper=scrape_cnbc_finance),
}


def parse_financial_news(
    max_items_per_source: int = 9999,
    sources: Optional[List[str]] = None,
) -> Dict[str, Any]:
    """
    Parse financial news from multiple sources and save into Article records.
    """
    if max_items_per_source <= 0:
        max_items_per_source = 1

    selected_keys = sources or list(AVAILABLE_SOURCES.keys())
    selected = []
    for key in selected_keys:
        source = AVAILABLE_SOURCES.get(key)
        if source:
            selected.append(source)

    stats: Dict[str, Any] = {
        "checked": 0,
        "created": 0,
        "updated": 0,
        "skipped_existing": 0,
        "errors": [],
        "sources": {},
    }

    for source in selected:
        source_stats = {
            "checked": 0,
            "created": 0,
            "updated": 0,
            "skipped_existing": 0,
            "errors": [],
        }
```

```
        try:
            for article in source.scraper(max_items_per_source):
                _persist_article(article, source_stats)
        except Exception as exc:
            logger.exception("Failed to parse %s", source.name)
            source_stats["errors"].append({"source": source.name, "error": str(exc)})

        stats["checked"] += source_stats["checked"]
        stats["created"] += source_stats["created"]
        stats["updated"] += source_stats["updated"]
        stats["skipped_existing"] += source_stats["skipped_existing"]
        stats["errors"].extend(source_stats["errors"])
        stats["sources"][source.key] = source_stats

    return stats

# services/article_analysis.py

from __future__ import annotations

import re
from typing import List, Dict, Any, Iterable

from django.utils import timezone

from .models import Article, ArticleAnalysis, ArticleStatus
from .text_processing import build_text_metadata
from .services.embeddings import get_article_embedding


# ---------------------------------------------------------------------------
# Rule-based topic detection and sentiment seed vocabularies
# ---------------------------------------------------------------------------

TOPIC_KEYWORDS = {
    "markets": {
        "market",
        "stock",
        "equity",
        "share",
        "index",
        "dow",
        "nasdaq",
        "s&p",
        "bond",
        "yield",
    },
    "macro": {
        "inflation",
        "interest",
```

```
        "rate",
        "central bank",
        "fed",
        "ecb",
        "gdp",
        "economy",
        "economic",
        "recession",
        "unemployment",
        "growth",
    },
    "companies": {
        "company",
        "profit",
        "earnings",
        "revenue",
        "quarter",
        "q1",
        "q2",
        "q3",
        "q4",
        "results",
        "forecast",
        "guidance",
        "merger",
        "acquisition",
        "m&a",
        "ipo",
    },
    "currencies": {
        "currency",
        "dollar",
        "euro",
        "pound",
        "yen",
        "exchange rate",
        "forex",
        "fx",
    },
    "commodities": {
        "oil",
        "gas",
        "gold",
        "silver",
        "commodity",
        "brent",
        "wti",
        "energy",
    },
    "policy_regulation": {
```

```
        "regulation",
        "ban",
        "sanction",
        "tariff",
        "tax",
        "policy",
        "law",
        "rules",
    },
}

POSITIVE_WORDS = {
    "gain",
    "gains",
    "rise",
    "rises",
    "surge",
    "soar",
    "soars",
    "boost",
    "beat",
    "record",
    "profit",
    "profits",
    "growth",
    "expand",
    "expansion",
    "strong",
    "rebound",
    "recover",
    "recovery",
    "upbeat",
    "optimistic",
    "bullish",
    "outperform",
    "robust",
    "rally",
    "improve",
    "improves",
    "improved",
}

NEGATIVE_WORDS = {
    "loss",
    "losses",
    "fall",
    "falls",
    "drop",
    "drops",
    "plunge",
```

```
    "slump",
    "decline",
    "cut",
    "cuts",
    "miss",
    "warning",
    "slowdown",
    "crisis",
    "recession",
    "weak",
    "collapse",
    "bearish",
    "selloff",
    "layoff",
    "layoffs",
    "shutdown",
    "strike",
    "protest",
    "lawsuit",
    "fine",
}

POSITIVE_CUES = {
    "beat",
    "beats",
    "record",
    "surge",
    "soar",
    "upgrade",
    "growth",
    "increase",
    "rally",
    "gain",
    "gains",
    "profit",
    "profits",
}

NEGATIVE_CUES = {
    "miss",
    "misses",
    "cut",
    "cuts",
    "downgrade",
    "decline",
    "drop",
    "drops",
    "fall",
    "falls",
    "plunge",
```

```
    "slump",
    "loss",
    "losses",
    "fine",
    "lawsuit",
}

POSITIVE_CONTEXT = POSITIVE_WORDS | POSITIVE_CUES | {"turnaround", "buoyant"}
NEGATIVE_CONTEXT = NEGATIVE_WORDS | NEGATIVE_CUES | {"recall", "meltdown",
"crackdown", "antitrust", "regulatory", "penalty"}

RISK_KEYWORDS = {
    "crisis",
    "recession",
    "default",
    "bankruptcy",
    "sanction",
    "collapse",
    "risk",
    "fraud",
    "downturn",
    "meltdown",
}

EVENT_PATTERNS = {
    "earnings": {"earnings", "profit", "loss", "revenue", "guidance", "forecast",
"q1", "q2", "q3", "q4"},
    "merger_acquisition": {"merger", "acquisition", "m&a", "buyout", "deal",
"purchase"},
    "regulatory": {"regulation", "ban", "sanction", "tariff", "fine", "lawsuit",
"antitrust", "probe"},
    "macroeconomic": {"inflation", "rate hike", "rate cut", "gdp", "recession",
"unemployment"},
    "product": {"launch", "product", "unveil", "release"},
}


def build_graph_data(
    meta: Dict[str, Any],
    topics: List[Dict[str, Any]],
    keywords: List[str],
    entity_sentiments: Dict[str, Any],
    risk_info: Dict[str, Any],
) -> Dict[str, Any]:
    """
    Prepare lightweight graph-friendly data for the frontend.
    """
    topic_bars = [{"label": t.get("label"), "value": float(t.get("score") or 0) *
100} for t in topics]
```

```
    keyword_weights: List[Dict[str, Any]] = []
    for idx, kw in enumerate(keywords[:20]):
        weight = max(10, 100 - idx * 4)
        keyword_weights.append({"label": kw, "value": weight})

    sentiment_counts = {"positive": 0, "negative": 0, "neutral": 0}
    for data in entity_sentiments.values():
        sentiment = data.get("sentiment", "neutral")
        if sentiment in sentiment_counts:
            sentiment_counts[sentiment] += 1

    stats = meta.get("stats") or {}

    return {
        "topics": topic_bars,
        "keywords": keyword_weights,
        "entity_sentiments": sentiment_counts,
        "risk": {"score": risk_info.get("score"), "level": risk_info.get("level")},
        "text_stats": {
            "tokens": stats.get("token_count", 0),
            "sentences": stats.get("sentence_count", 0),
            "avg_tokens_per_sentence": stats.get("avg_tokens_per_sentence", 0.0),
        },
    }


def detect_topics(meta: Dict[str, Any]) -> List[Dict[str, Any]]:
    """
    Identify coarse topics using keyword overlap.
    Returns [{label, score, matched_keywords}].
    """
    keywords = meta.get("keywords", [])
    if not keywords:
        return []

    kw_set = set(keywords)
    topic_scores: List[Dict[str, Any]] = []

    for topic, topic_words in TOPIC_KEYWORDS.items():
        intersect = kw_set & topic_words
        if not intersect:
            continue
        score = len(intersect) / len(topic_words)
        topic_scores.append(
            {
                "label": topic,
                "score": round(score, 3),
                "matched_keywords": sorted(intersect),
            }
        )
```

```
    topic_scores.sort(key=lambda x: x["score"], reverse=True)
    return topic_scores


def detect_sentiment(meta: Dict[str, Any]) -> str:
    """
    Light-weight sentiment from keyword buckets.
    """
    keywords = meta.get("keywords", [])
    if not keywords:
        return "neutral"

    pos = sum(1 for w in keywords if w in POSITIVE_WORDS)
    neg = sum(1 for w in keywords if w in NEGATIVE_WORDS)

    if pos > neg:
        return "positive"
    if neg > pos:
        return "negative"
    return "neutral"


def compute_entity_sentiment(
    cleaned_text: str,
    entities: List[Dict[str, Any]],
    window: int = 8,
    default_sentiment: str | None = None,
) -> Dict[str, Any]:
    """
    Rule-of-thumb sentiment for entities (ORG/GPE) based on a token window.
    Returns mapping: entity_text -> {"label": ..., "sentiment": ..., "pos": int,
"neg": int}.
    If there is no local context, falls back to the article-level sentiment when
provided.
    """
    if not cleaned_text or not entities:
        return {}

    result: Dict[str, Any] = {}
    tokens = re.findall(r"\b\w+\b", cleaned_text.lower())

    fallback_default = default_sentiment if default_sentiment in {"positive",
"negative"} else None

    # Expand the context window to capture adjectives/verbs a bit farther from the
entity mention.
    window = max(window, 8)

    def _sentiment_from_counts(pos_count: int, neg_count: int) -> str:
```

```
        if pos_count > neg_count:
            return "positive"
        if neg_count > pos_count:
            return "negative"
        return "neutral"

    for ent in entities:
        label = ent.get("label")
        if label not in {"ORG", "GPE"}:
            continue

        text = (ent.get("text") or "").lower()
        if not text:
            continue

        words = text.split()
        occurrences: List[int] = []
        for idx, tok in enumerate(tokens):
            if tok == words[0] and tokens[idx : idx + len(words)] == words:
                occurrences.append(idx)

        if not occurrences:
            continue

        pos_total = 0
        neg_total = 0
        for idx in occurrences:
            start = max(0, idx - window)
            end = min(len(tokens), idx + len(words) + window)
            window_tokens = tokens[start:end]
            pos_total += sum(1 for t in window_tokens if t in POSITIVE_CONTEXT)
            neg_total += sum(1 for t in window_tokens if t in NEGATIVE_CONTEXT)

        sentiment = _sentiment_from_counts(pos_total, neg_total)
        if sentiment == "neutral" and pos_total == 0 and neg_total == 0 and
fallback_default:
            sentiment = fallback_default
        result[text] = {
            "label": label,
            "sentiment": sentiment,
            "pos": pos_total,
            "neg": neg_total,
        }

    # If there is only one focal entity and its tone is still neutral, bias it toward
the article-level tone.
    if len(result) == 1 and fallback_default:
        only_key = next(iter(result))
        if result[only_key]["sentiment"] == "neutral":
            result[only_key]["sentiment"] = fallback_default
```

```
    return result


def estimate_risk(meta: Dict[str, Any]) -> Dict[str, Any]:
    """
    Estimate a lightweight risk score in [0,1] based on keywords and entity
distribution.
    """
    cleaned = meta.get("cleaned_text") or ""
    entities: Iterable[Dict[str, Any]] = meta.get("entities") or []
    stats = meta.get("stats") or {}

    tokens = re.findall(r"\b\w+\b", cleaned.lower())
    token_count = max(1, stats.get("token_count") or len(tokens) or 1)

    risk_kw_hits = sum(1 for t in tokens if t in RISK_KEYWORDS)

    label_counts: Dict[str, int] = {}
    for ent in entities:
        label = ent.get("label")
        if not label:
            continue
        label_counts[label] = label_counts.get(label, 0) + 1

    money_factor = min(label_counts.get("MONEY", 0), 10) * 0.02
    org_factor = min(label_counts.get("ORG", 0), 10) * 0.01
    percent_factor = min(label_counts.get("PERCENT", 0), 10) * 0.02
    keyword_factor = min(risk_kw_hits / token_count * 3.0, 1.0)

    score = min(keyword_factor + money_factor + org_factor + percent_factor, 1.0)

    if score >= 0.66:
        level = "high"
    elif score >= 0.33:
        level = "medium"
    else:
        level = "low"

    return {
        "score": round(score, 3),
        "level": level,
        "details": {
            "risk_keywords": risk_kw_hits,
            "token_count": token_count,
            "entities": label_counts,
        },
    }
```

```python
def predict_impact(
    sentiment: str,
    risk_score: float,
    pos_hits: int = 0,
    neg_hits: int = 0,
    event_tags: List[str] | None = None,
) -> str:
    """
    Map sentiment, risk, cues, and event tags to a coarse impact label.
    """
    event_tags = event_tags or []

    if risk_score >= 0.66 or "regulatory" in event_tags:
        return "negative"

    if neg_hits >= pos_hits + 2:
        return "negative"
    if pos_hits >= neg_hits + 2:
        return "positive"

    if sentiment == "positive":
        return "positive"
    if sentiment == "negative":
        return "negative"

    if "earnings" in event_tags:
        if pos_hits > neg_hits:
            return "positive"
        if neg_hits > pos_hits:
            return "negative"

    if risk_score >= 0.5:
        return "negative"
    return "neutral"


def detect_event_tags(meta: Dict[str, Any]) -> List[str]:
    """
    Detect coarse event tags based on cleaned text keywords.
    """
    text = (meta.get("cleaned_text") or "").lower()
    keywords = set(meta.get("keywords") or [])

    tags: List[str] = []
    for tag, patterns in EVENT_PATTERNS.items():
        for token in patterns:
            if token in keywords or token in text:
                tags.append(tag)
                break
```

```
    # Deduplicate while preserving order
    seen = set()
    unique_tags = []
    for t in tags:
        if t not in seen:
            unique_tags.append(t)
            seen.add(t)
    return unique_tags


def _count_sentiment_tokens(text: str) -> Dict[str, int]:
    tokens = re.findall(r"\b\w+\b", text.lower())
    pos_hits = sum(1 for t in tokens if t in POSITIVE_WORDS or t in POSITIVE_CUES)
    neg_hits = sum(1 for t in tokens if t in NEGATIVE_WORDS or t in NEGATIVE_CUES)
    return {"pos": pos_hits, "neg": neg_hits}


# ---------------------------------------------------------------------------
# Article analysis pipeline
# ---------------------------------------------------------------------------

def analyze_article(article: Article) -> ArticleAnalysis:
    """
    Perform text analysis for a single article.
    - build text metadata;
    - derive topics/sentiment;
    - compute entity-level sentiment, risk, embedding, and predictive tags;
    - persist Article and ArticleAnalysis records idempotently.
    """
    meta = build_text_metadata(article.parsed_text)

    keywords = meta["keywords"]
    entities = meta.get("entities", [])
    stats = meta.get("stats", {})

    topics = detect_topics(meta)
    sentiment = detect_sentiment(meta)
    entity_sentiments = compute_entity_sentiment(
        meta["cleaned_text"],
        entities,
        window=8,
        default_sentiment=sentiment,
    )
    risk_info = estimate_risk(meta)
    sentiment_counts = _count_sentiment_tokens(meta["cleaned_text"])
    impact = predict_impact(
        sentiment=sentiment,
        risk_score=risk_info.get("score", 0.0),
        pos_hits=sentiment_counts["pos"],
        neg_hits=sentiment_counts["neg"],
```

```
        event_tags=[],
    )
    event_tags = detect_event_tags(meta)
    impact = predict_impact(
        sentiment=sentiment,
        risk_score=risk_info.get("score", 0.0),
        pos_hits=sentiment_counts["pos"],
        neg_hits=sentiment_counts["neg"],
        event_tags=event_tags,
    )
    embedding = get_article_embedding(meta["cleaned_text"])
    graphs = build_graph_data(meta, topics, keywords, entity_sentiments, risk_info)

    article.cleaned_text = meta["cleaned_text"]
    article.status = ArticleStatus.ANALYZED

    extra_meta = article.scrape_metadata or {}
    extra_meta["entities"] = entities
    extra_meta["text_stats"] = stats
    extra_meta["auto_topics"] = topics
    extra_meta["auto_sentiment"] = sentiment
    extra_meta["entity_sentiments"] = entity_sentiments
    extra_meta["risk"] = risk_info
    article.scrape_metadata = extra_meta

    article.save(update_fields=["cleaned_text", "status", "scrape_metadata",
"updated_at"])

    analysis, created = ArticleAnalysis.objects.get_or_create(
        article=article,
        defaults={
            "keywords": keywords,
            "topics": topics,
            "sentiment": sentiment,
            "embedding": embedding,
            "impact": impact,
            "event_tags": event_tags,
            "graphs": graphs,
            "analyzed_at": timezone.now(),
        },
    )

    if not created:
        analysis.keywords = keywords
        analysis.topics = topics
        analysis.sentiment = sentiment
        analysis.embedding = embedding
        analysis.impact = impact
        analysis.event_tags = event_tags
        analysis.graphs = graphs
```

```
        analysis.analyzed_at = timezone.now()
        analysis.save(
            update_fields=["keywords", "topics", "sentiment", "embedding", "impact",
"event_tags", "graphs", "analyzed_at"]
        )

    return analysis

# text_processing.py

from __future__ import annotations

import re
from collections import Counter
from typing import List, Dict, Any, Optional

import spacy
from spacy.language import Language
from spacy.tokens import Doc

# ---------------------------------------------------------------------------
# spaCy bootstrap (lazy load with a lightweight fallback to avoid runtime crashes)
# ---------------------------------------------------------------------------

_NLP: Optional[Language] = None


def _build_fallback_nlp() -> Language:
    """
    Build a minimal English pipeline when the full model is unavailable.
    Adds a sentencizer and a tiny EntityRuler so unit tests remain stable.
    """
    nlp = spacy.blank("en")
    if "sentencizer" not in nlp.pipe_names:
        nlp.add_pipe("sentencizer")

    ruler = nlp.add_pipe("entity_ruler")
    ruler.add_patterns(
        [
            {"label": "ORG", "pattern": "Apple"},
            {"label": "ORG", "pattern": "Microsoft"},
            {"label": "GPE", "pattern": "London"},
            {"label": "GPE", "pattern": "USA"},
        ]
    )
    return nlp


def get_nlp() -> Language:
    """
```

```
    Return a cached spaCy Language pipeline.
    Falls back to a lightweight blank model if the standard model is missing.
    """
    global _NLP
    if _NLP is None:
        try:
            _NLP = spacy.load("en_core_web_sm")
        except Exception:
            _NLP = _build_fallback_nlp()
    return _NLP


# ---------------------------------------------------------------------------
# Text normalization helpers
# ---------------------------------------------------------------------------

def clean_text(text: str) -> str:
    """
    Basic normalization:
    - replace NBSP/zero-width characters;
    - collapse whitespace.
    """
    if not text:
        return ""

    text = text.replace("\xa0", " ").replace("\u200b", "")
    text = re.sub(r"\s+", " ", text, flags=re.MULTILINE)
    return text.strip()


def preprocess_text(text: str) -> str:
    """
    Prepare text for analysis:
    - basic cleanup;
    - lowercase for keyword extraction.
    """
    text = clean_text(text)
    return text.lower()


# ---------------------------------------------------------------------------
# Keyword extraction
# ---------------------------------------------------------------------------

def _extract_candidate_tokens(doc: Doc) -> List[str]:
    """
    Collect candidate lemmas for keywords:
    - alphabetic tokens only;
    - no stop words;
    - nouns/proper nouns with length >= 3.
```

```
    """
    candidates: List[str] = []
    for token in doc:
        if not token.is_alpha:
            continue
        if token.is_stop:
            continue
        if token.pos_ not in {"NOUN", "PROPN"}:
            continue

        lemma = token.lemma_.lower()
        if len(lemma) < 3:
            continue
        candidates.append(lemma)
    return candidates


def extract_keywords_from_doc(doc: Doc, top_n: int = 15) -> List[str]:
    """
    Return the top-N frequent candidate lemmas from a spaCy Doc.
    """
    candidates = _extract_candidate_tokens(doc)
    if not candidates:
        return []

    freq = Counter(candidates)
    sorted_items = sorted(freq.items(), key=lambda x: (-x[1], x[0]))
    return [word for word, _ in sorted_items[:top_n]]


def extract_keywords(text: str, top_n: int = 15) -> List[str]:
    """
    Convenience wrapper for keyword extraction from raw text.
    """
    nlp = get_nlp()
    preprocessed = preprocess_text(text)
    doc = nlp(preprocessed)
    return extract_keywords_from_doc(doc, top_n=top_n)


# ---------------------------------------------------------------------------
# Entities
# ---------------------------------------------------------------------------

def extract_entities_from_doc(doc: Doc) -> List[Dict[str, Any]]:
    """
    Extract entities as dictionaries with offsets.
    """
    entities: List[Dict[str, Any]] = []
    for ent in doc.ents:
```

```
        entities.append(
            {
                "text": ent.text,
                "label": ent.label_,
                "start": int(ent.start_char),
                "end": int(ent.end_char),
            }
        )
    return entities


def extract_entities(text: str) -> List[Dict[str, Any]]:
    """
    Convenience wrapper for entity extraction from raw text.
    """
    nlp = get_nlp()
    preprocessed = clean_text(text)
    doc = nlp(preprocessed)
    return extract_entities_from_doc(doc)


# ------------------------------------------------------------------------
# Full metadata assembly
# ------------------------------------------------------------------------

def build_text_metadata(text: str, top_n_keywords: int = 15) -> Dict[str, Any]:
    """
    Build normalized text metadata:
        - cleaned_text: normalized lowercase text
        - keywords: top lemmas (lowercase)
        - entities: list of {text,label,start,end}
        - stats: token_count, sentence_count, char_count, avg_tokens_per_sentence
    """
    nlp = get_nlp()
    cleaned_lower = preprocess_text(text)
    cleaned_original = clean_text(text)

    if not cleaned_original:
        return {
            "cleaned_text": "",
            "keywords": [],
            "entities": [],
            "stats": {"token_count": 0, "sentence_count": 0, "char_count": 0,
"avg_tokens_per_sentence": 0.0},
        }

    doc_for_keywords = nlp(cleaned_lower)
    doc_for_entities = nlp(cleaned_original)

    keywords = extract_keywords_from_doc(doc_for_keywords, top_n=top_n_keywords)
```

```
    entities = extract_entities_from_doc(doc_for_entities)

    tokens = [t for t in doc_for_keywords if not t.is_space]
    sentences = list(doc_for_keywords.sents)
    sentence_count = len(sentences)
    token_count = len(tokens)
    avg_tokens_per_sentence = token_count / sentence_count if sentence_count else 0.0

    stats = {
        "token_count": token_count,
        "sentence_count": sentence_count,
        "char_count": len(cleaned_original),
        "avg_tokens_per_sentence": round(avg_tokens_per_sentence, 3),
    }

    return {
        "cleaned_text": cleaned_lower,
        "keywords": keywords,
        "entities": entities,
        "stats": stats,
    }
```