МАГІСТЕРСЬКА РОБОТА

МР. ІПм - 20.00.00.000 ПЗ

Група ІПм-24-1

Сінітович Олег

2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Сінітович Олег Васильович

(прізвище, ім'я, по батькові)

УДК 004.9

(індекс)

МАГІСТЕРСЬКА РОБОТА

Моделі та методи сервісів архівування з відстеженням змін в веб-сторінках

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Сінітович О.В.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник Крихівський Михайло Васильович, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту

Завідувач кафедри

доц. Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

Нормоконтроль

доц. Вовк Р.Б.

(посада) (підпис) (дата) (ініціали та прізвище)

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

Івано-Франківськ – 2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедрою **ІПЗ**

доц. В.В. Бандура

“ 04 ” вересня 2025 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Сінітовичу Олегу Васильовичу

(прізвище, ім’я, по-батькові)

1. Тема магістерської роботи “ Моделі та методи сервісів архівування з відстеженням змін в веб-сторінках”

керівник проекту (роботи) Крихівський М.В., к.т.н., доцент

затверджені наказом закладу вищої освіти від “ 05 ” листопада 2025 р. № 695/7

2. Строк подання студентом проекту (роботи) 15 грудня 2025 р.

3. Вихідні дані до проекту (роботи) Теоретичні концепції та формальні моделі функціонування інформаційних технологій моніторингу змін на веб-ресурсах

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

1. Дослідження актуальності проблеми відстеження змін у веб-сторінках
2. Дослідження методів та алгоритмів сервісів архівування з відстеженням змін в веб-сторінках
3. Представлення архітектури та методології пошуку змін у веб-сторінках та архівах
4. Огляд функціональності навігаційної панелі веб-архівів та пропозиції щодо її вдосконалення

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

1. Принцип роботи системи Apache Lucene (рис. 2.1)
2. Спрощена архітектура Solr (рис. 2.2)
3. Короткий перелік форматів що підтримуються Apache Tika (рис. 2.3)
4. Робочий процес Solrwayback (рис. 2.4)
5. Два приклади пошуку URI в веб-архіві (рис. 2.5)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата
Перевірка на плагіат	доц., к.т.н. Вовк Р.Б.	

7. Дата видачі завдання 04 вересня 2025 р.

Керівник _____
(підпис)

Завдання прийняв до виконання_____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерської роботи	15.09.2025	виконано
2	Дослідження актуальності проблеми відстеження змін у веб-сторінках	29.09.2025	виконано
3	Дослідження методів та алгоритмів сервісів архівування з відстеженням змін в веб-сторінках	15.10.2025	виконано
4	Представлення архітектури та методології пошуку змін у веб-сторінках та архівах	08.11.2025	виконано
5	Огляд функціональності навігаційної панелі веб-архівів та пропозиції щодо її вдосконалення	20.11.2025	виконано
6	Затвердження пояснювальної записки роботи завідувачем кафедри	13.12.2025	виконано

Студент – магістр _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Магістерська робота: 77 с., 19 рис., 2 табл., 39 джерел.

Тема: Моделі та методи сервісів архівування з відстеженням змін в веб-сторінках

Мета магістерської роботи - розроблення теоретичних і практичних основ побудови моделей та методів сервісів веб-архівації, які забезпечують ефективне відстеження, аналіз і візуалізацію змін у веб-сторінках.

Об'єкт дослідження - процеси збирання, зберігання та аналізу версій веб-документів у системах веб-архівації.

Предмет дослідження - моделі, алгоритми та архітектурні рішення сервісів архівування з підтримкою пошуку і візуалізації змін у веб-сторінках.

Результати дослідження

В роботі розроблено архітектуру пошукової системи змін у тексті з використанням технологій Lucene та Solr, що забезпечує ефективну обробку версійних колекцій документів.

Висновок

Реалізовано прототип користувацького інтерфейсу для інтерактивної роботи з архівами, який враховує особливості людського сприйняття змін та когнітивні закономірності пошуку інформації.

ВЕБ-АРХІВУВАННЯ, ВІДСТЕЖЕННЯ ЗМІН, ІНДЕКСАЦІЯ, APACHE LUCENE, ВЕРСІЙНІ ДОКУМЕНТИ, ПОШУКОВІ СИСТЕМИ, ВІЗУАЛІЗАЦІЯ ЗМІН, АНАЛІЗ КОНТЕНТУ

ABSTRACT

Master Thesis: 77 pp., 19 fig., 2 tab., 39 sources.

Topic: Models and methods of archiving services with tracking changes in web pages

The purpose of the master's thesis is to develop theoretical and practical foundations for building models and methods of web archiving services that provide effective tracking, analysis and visualization of changes in web pages.

The object of research is the processes of collecting, storing and analyzing versions of web documents in web archiving systems.

The subject of research is models, algorithms and architectural solutions of archiving services with support for searching and visualizing changes in web pages.

Research results

The work developed the architecture of a text change search system using Lucene and Solr technologies, which ensures effective processing of versioned document collections.

Conclusion

A prototype of a user interface for interactive work with archives was implemented, which takes into account the peculiarities of human perception of changes and cognitive patterns of information search.

WEB ARCHIVING, CHANGE TRACKING, INDEXING, APACHE LUCENE, VERSIONED DOCUMENTS, SEARCH SYSTEMS, CHANGE VISUALIZATION, CONTENT ANALYSIS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 9
ВСТУП 10

РОЗДІЛ 1. ДОСЛІДЖЕННЯ АКТУАЛЬНОСТІ ПРОБЛЕМИ ВІДСТЕЖЕННЯ ЗМІН У ВЕБ-СТОРІНКАХ 13
1.1. Концептуалізація процесів пошуку та візуалізації змін у веб-архівах. 13
1.1.1. Обмеження існуючих пошукових інтерфейсів 13
1.1.2. Пропозиції щодо покращення навігації 14
1.2. Проблема доступу до історичних змін контенту у веб-архівах 14
1.2.1. Обґрунтування проблеми 14
1.2.2. Обмеження поточних пошукових інтерфейсів 16
1.2.3. Опис проблеми дослідження 18
1.3. Людська когніція та електронний пошук інформації 18
1.3.1. Вплив когнітивних процесів на дизайн пошукових систем 19
1.3.2. Підтримка когнітивних функцій 20
1.4. Основи функціонування веб-архітектури та веб-архівів 20
1.4.1. Структура веб-архівів 21
1.4.2. Функціональність відтворення веб-архівів 23
1.4.3. Перегляд змін у веб-архівах 24
Висновки до розділу 26

РОЗДІЛ 2. ДОСЛІДЖЕННЯ МЕТОДІВ ТА АЛГОРИТМІВ СЕРВІСІВ АРХІВУВАННЯ З ВІДСТЕЖЕННЯМ ЗМІН В ВЕБ-СТОРІНКАХ 27
2.1. Архітектура пошуку та особливості пошукових систем веб-архівів 27
2.1.1. Основні компоненти пошукової системи 27
2.1.2. Набір пошукових інструментів Apache 27
2.1.3. Робочий процес індексації 31
2.2. Набори даних для аналізу змін у вебсторінках та запитів 36
2.2.1. Загальні набори даних вебсканувань 36

2.2.2 Вебархів "Кінець терміну" (End of Term Web Archive) 37
2.2.3 Набір даних EDGI 37
2.2.4 Набір даних ORCAS 38
2.2.5. Набір даних журналу запитів архіву (AQL) 39
2.3. Аналіз поведінки користувачів під час роботи з інструментами перегляду змін 41
2.3.1. Методи презентації порівняння даних 41
2.3.2. Стратегії візуалізації змін 43
Висновки до розділу 44

РОЗДІЛ 3. ПРЕДСТАВЛЕННЯ АРХІТЕКТУРИ ТА МЕТОДОЛОГІЇ ПОШУКУ ЗМІН У ВЕБ-СТОРІНКАХ ТА АРХІВАХ 45
3.1. Архітектура пошукової системи змін у тексті 45
3.1.1. Етап отримання документів 46
3.1.2. Solr для версійних колекцій документів 47
3.1.3 Обчислення тимчасових діапазонів дійсності за допомогою Lucene 48
3.1.4. Обчислення змін тексту за допомогою Lucene 50
3.1.5. Запити змін тексту за допомогою Lucene 52
3.1.6. Ранжування результатів пошуку змін тексту 53
3.2. Реалізація користувацького інтерфейсу для взаємодії зі змінами тексту 54
3.2.1. Сторінка запиту та результатів пошукової системи змін тексту 58
3.2.3. Перегляд відмінностей 62
3.2.4. Переглядач анімованих видалень 64
3.3. Огляд функціональності навігаційної панелі веб-архівів та пропозиції щодо її вдосконалення 68
Висновки до розділу 70
ВИСНОВКИ 72
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 74

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

UAX - Unicode Standard Annex - додаток до стандарту Юнікод

SERP - Search Engine Results Page - сторінка результатів пошукової системи

SOLR - Apache Solr - Пошукова платформа

Diff – Difference - Різниця (функція порівняння)

IR - Information Retrieval - Інформаційний пошук

DOM - Document Object Model - Об'єктна модель документа

RFC - Request for Comments - Запит на коментарі

WM - Wayback Machine - Система Internet Archive

ВСТУП

Актуальність теми.

У сучасну епоху стрімкого розвитку інформаційних технологій веб-простір став головним джерелом зберігання, поширення та оновлення знань. Щодня мільйони веб-сторінок змінюються, доповнюються або видаляються, що створює виклик для забезпечення цілісності цифрової інформаційної спадщини. Веб-архіви виконують важливу функцію збереження минулих станів Інтернету, проте більшість наявних систем не забезпечують ефективного механізму відстеження змін контенту, порівняння версій чи візуалізації динаміки сторінок.

Проблема полягає у відсутності інтегрованих рішень, які поєднують глибокий аналіз еволюції веб-контенту з когнітивно орієнтованим інтерфейсом користувача. Зважаючи на експоненційне зростання обсягів цифрових даних, традиційні методи архівування вже не відповідають сучасним вимогам до швидкості обробки, масштабованості та аналітичності. Тому виникає потреба у створенні нових моделей та методів веб-архівації, які дозволяють не лише фіксувати копії веб-сторінок, а й забезпечувати їхній динамічний аналіз і відображення змін у часі.

Дана магістерська робота спрямована на комплексне дослідження теоретичних, алгоритмічних та архітектурних засад побудови сервісів архівування з відстеженням змін у веб-сторінках. Особлива увага приділяється розробленню архітектури пошукової системи змін, створенню моделей індексації та візуалізації відмінностей між версіями документів, а також формуванню користувацьких інтерфейсів, що сприяють зручному та ефективному аналізу еволюції веб-контенту.

Актуальність теми обумовлена необхідністю забезпечення збереження, доступності та достовірності веб-контенту, який є невід’ємною частиною культурної, наукової та інформаційної спадщини людства. В умовах постійного оновлення веб-ресурсів втрачається значна частина даних, які

можуть мати наукову, історичну чи аналітичну цінність. Веб-архіви виступають ключовими інструментами у збереженні цієї інформації, однак сучасні системи часто не мають достатніх можливостей для детального аналізу змін і візуального порівняння версій.

Особливої актуальності набуває створення інтелектуальних сервісів, здатних автоматично виявляти, класифікувати та візуалізувати зміни у веб-контенті, використовуючи сучасні методи індексації, пошуку та когнітивно-орієнтованого дизайну. Такі рішення мають вагоме значення не лише для архівних установ, а й для дослідників, журналістів, аналітиків, які потребують інструментів для відстеження інформаційних змін і перевірки достовірності джерел.

Таким чином, розроблення моделей і методів сервісів архівування з підтримкою механізмів відстеження змін є актуальним завданням сучасної інформаційної науки та програмної інженерії.

Метою магістерської роботи є розроблення теоретичних і практичних основ побудови моделей та методів сервісів веб-архівації, які забезпечують ефективне відстеження, аналіз і візуалізацію змін у веб-сторінках.

Об’єктом дослідження є процеси збирання, зберігання та аналізу версій веб-документів у системах веб-архівації.

Предметом дослідження є моделі, алгоритми та архітектурні рішення сервісів архівування з підтримкою пошуку і візуалізації змін у веб-сторінках.

Для досягнення поставленої мети в роботі необхідно **вирішити такі завдання**:

- Проаналізувати сучасний стан проблеми архівування та відстеження змін у веб-сторінках.

- Дослідити архітектуру пошукових систем веб-архівів і визначити їхні функціональні обмеження.

- Розглянути існуючі алгоритми індексації, пошуку та обчислення відмінностей між версіями документів.

- Розробити модель архітектури пошукової системи змін у веб-сторінках із використанням технологій Apache Solr та Lucene.

- Реалізувати прототип користувацького інтерфейсу для візуалізації змін і проведення інтерактивного пошуку за історичними даними.

Методи дослідження

У процесі дослідження застосовувалися методи системного аналізу, формалізації інформаційних процесів, алгоритмічного моделювання, а також методи проєктування архітектурних рішень програмних систем. Для реалізації моделей пошуку використовувалися технології індексації Apache Lucene і Solr, що забезпечують обробку великих обсягів версійних даних. Також застосовувалися методи когнітивного моделювання під час створення інтерфейсу користувача, орієнтованого на зручну взаємодію з архівними даними.

Наукова новизна отриманих результатів

Запропоновано цілісну архітектуру сервісу веб-архівації з підтримкою аналізу змін у текстовому контенті веб-сторінок та розроблено методику побудови пошукових індексів для версійних документів із визначенням часових діапазонів дійсності сторінок.

Практичне застосування результатів

Розроблені моделі, методи та архітектурні рішення можуть бути використані для вдосконалення існуючих систем веб-архівації, таких як Internet Archive або національні архівні ініціативи. Результати дослідження можуть застосовуватися при створенні інструментів моніторингу змін у державних, освітніх та інформаційних порталах, а також у проєктах з цифрової історії.

Структура магістерської роботи. Робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 77 сторінjк, і містить 19 рисунків, 2 таблицs, список використаних джерел із 39 найменувань.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ АКТУАЛЬНОСТІ ПРОБЛЕМИ ВІДСТЕЖЕННЯ ЗМІН У ВЕБ-СТОРІНКАХ

1.1. Концептуалізація процесів пошуку та візуалізації змін у веб-архівах

Веб-сторінки є динамічними об'єктами, що еволюціонують з часом. Веб-архіви виконують функцію збереження історичних копій цих сторінок. Користувачі веб-архівів, зокрема дослідники та журналісти, часто потребують можливості ідентифікувати та аналізувати зміни, що відбулися на веб-сторінках протягом певного періоду. Однак, наявні інтерфейси пошуку веб-архівів не забезпечують адекватної підтримки для виконання цього завдання.

1.1.1. Обмеження існуючих пошукових інтерфейсів

У веб-архівах із функціональністю повнотекстового пошуку, кілька версій однієї веб-сторінки, які відповідають пошуковому запиту, зазвичай відображаються або ізольовано (без явного переліку змін), або групуються таким чином, що приховує істотні зміни. Така архітектура ускладнює користувачеві завдання відстеження еволюції контенту.

Ми пропонуємо розробку пошукового рушія для тексту змін, що дозволяє користувачам ефективно знаходити моменти та характер змін на веб-сторінках.

В пропонованій магістерській роботі детально описано імплементацію як бек-енду, так і фронт-енду пошукового рушія для виявлення зміненого контенту у веб-сторінках та веб-архівах. Фронт-енд включає інструмент візуалізації, який забезпечує користувачеві можливість переглядати зміни між двома версіями веб-сторінки в контексті, зокрема, у вигляді анімованого відображення.

1.1.2. Пропозиції щодо покращення навігації

Для подальшої підтримки користувачів у перегляді змін, пропонуються модифікації до навігаційного банера відтворення архівних систем, наприклад, Wayback Machine (як ілюстративного прикладу подібної системи).

Оцінка функціональності пошукового рушія була проведена на корпусі веб-сторінок, які демонстрували зміни протягом значного проміжку часу (наприклад, між 2016 та 2020 роками).

Сторінка результатів пошуку тексту змін здатна чітко ідентифікувати моменти, коли певні терміни або фрази були додані чи видалені з веб-сторінок. Крім того, аналіз інвертованого індексу може бути використаний для виявлення важливих або часто видалених термінів у всьому корпусі архіву.

Узгодження отриманого набору даних із реальним набором даних про кліки (пошукові запити користувачів) підтвердило, що користувачі активно шукали ті самі тематичні терміни, які згодом були видалені з веб-сторінок. Це підкреслює актуальність потреби в інструменті пошуку змін.

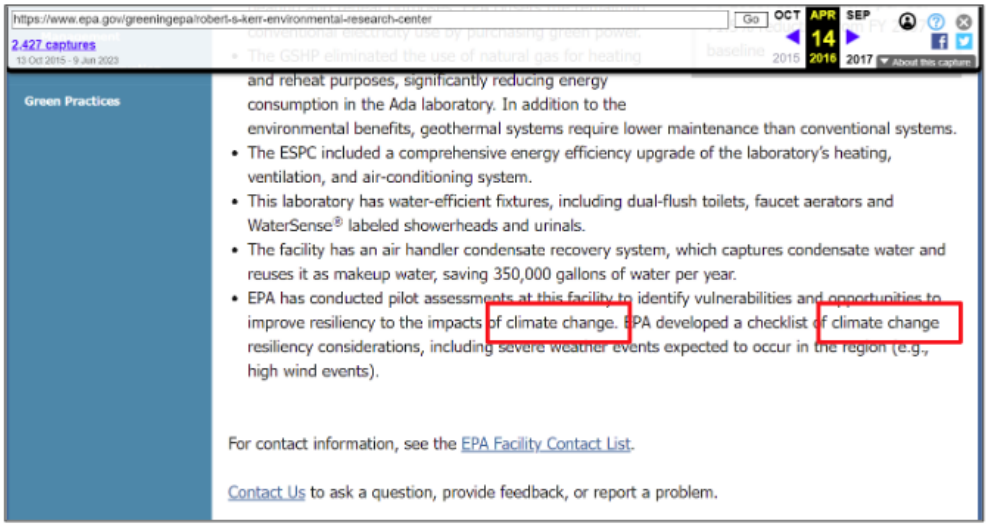
1.2. Проблема доступу до історичних змін контенту у веб-архівах

Відомо, що просте архівування історичних версій веб-сторінок є недостатнім для забезпечення користувачів належними засобами пошуку та ідентифікації змін у контенті, які їх цікавлять. Ця проблема набуває особливої актуальності в контексті збереження громадської інформації та забезпечення прозорості.

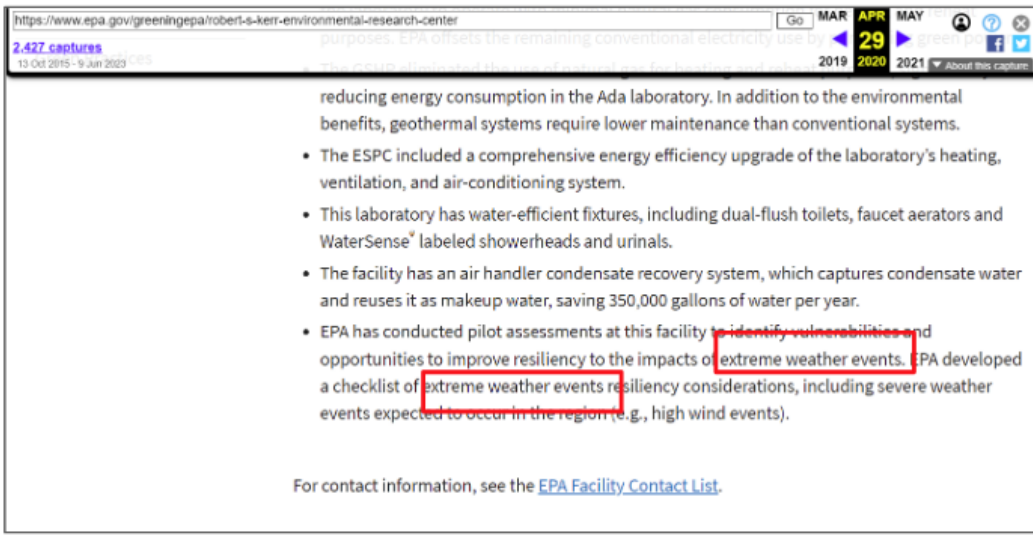
1.2.1. Обґрунтування проблеми

Яскравим прикладом реальних веб-сторінок, що зазнали значних змін, є федеральні екологічні веб-ресурси, зокрема ті, що стосуються періоду між президентськими термінами (наприклад, 2016–2020 роки). Хоча

громадськість покладається на уряд у забезпеченні доступу до неупередженої та достовірної інформації, у зазначений період на веб-сайтах федеральних екологічних агентств було зафіксовано суттєві редагування та вилучення контенту, а також видалення цілих сторінок.



а) Версія сторінки від 2016 року використовує фразу "зміна клімату"



б) Версія сторінки від 2020 року використовує фразу "екстремальні погодні події"

Рис. 1.1. Зміни на урядових веб-сайтах США у період з 2016 по 2020 рік, що демонструють вилучення фрази "зміна клімату"

Наприклад, на сторінці центру EPA Керра (як ілюстрація подібних змін) було виявлено систематичну заміну терміну "зміна клімату" на "екстремальні погодні події". Такі зміни, які можуть бути наочно проілюстровані (як на рис. 1.1), відстежувалися ініціативою з екологічних даних та управління (EDGI) з метою підвищення обізнаності громадськості та зміцнення урядової підзвітності, особливо за відсутності законодавства про цифрове легальне депонування.

1.2.2. Обмеження поточних пошукових інтерфейсів

Незважаючи на доведену необхідність, жоден із поточних інтерфейсів пошуку веб-архівів не підтримує пошук за такими відомими видаленими термінами.

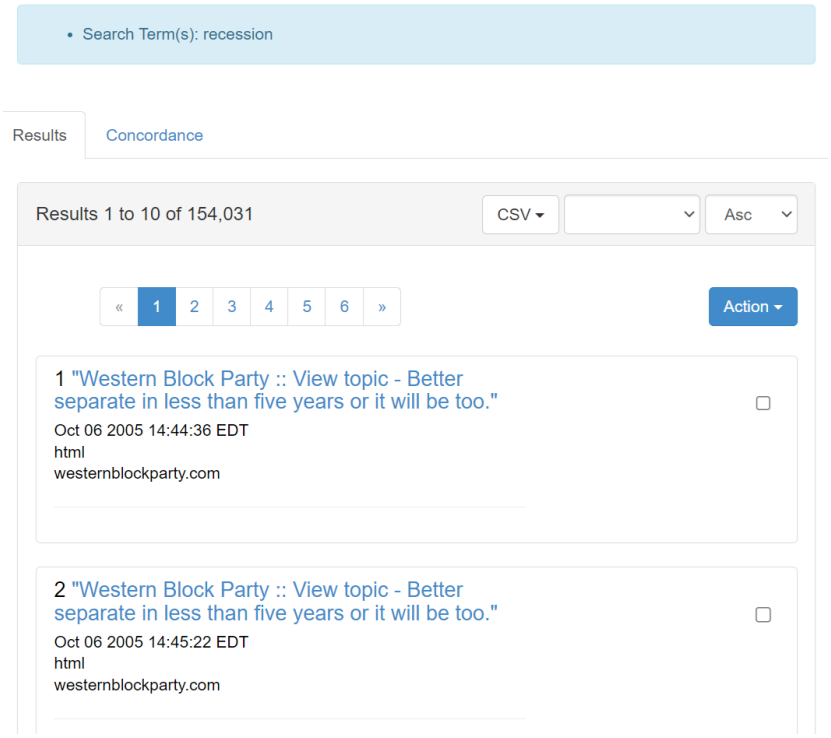
Поточна функціональність пошуку у веб-архівах характеризується значною обмеженістю:

- повна відсутність пошукових можливостей.
- обмеження пошуку лише метаданими.
- повнотекстовий пошук доступний лише для окремих колекцій.

Ключовим недоліком є те, що жоден з інтерфейсів не дозволяє користувачеві безпосередньо формулювати запити щодо змін, що відбулися на веб-сторінках. Проте, наявні емпіричні дані, отримані внаслідок двох формувальних досліджень, підтверджують, що реальні користувачі прагнуть використовувати веб-архіви саме для відстеження змін у часі.

Іншою відкритою проблемою в галузі пошуку інформації у веб-архівах є оптимальний спосіб презентації множинних копій однієї веб-сторінки, які відповідають пошуковому запиту. Існуючі підходи мають свої недоліки:

- включення лише однієї версії: Зменшує візуальне захаращення, але приховує історичні зміни.
- включення всіх версій: Надає доступ до всіх релевантних копій, але значно збільшує безлад (клаттер) у результатах.



а)

б)

Рис. 1.2. Приклади результатів пошуку у веб-архівах із групуванням та без групування для терміну "рецесія"

На рисунку 1.2 а подано результати пошуку без групування, де два верхні результати посилаються на ту саму сторінку, але з копіями, що відрізняються лише на одну хвилину (свідчить про надмірне дублювання).

На рисунку 1.2 б подано пошук у колекції Internet Archive Wayback Machine, який показує лише один результат на сторінку. Основна версія

сторінки, на яку веде посилання, містить пошуковий термін 28 разів, тоді як найперша доступна архівна копія не містить цього терміну.

Як демонструє порівняння згрупованих та незгрупованих результатів (рис. 1.2), незгрупована сторінка надмірно захаращена ідентичними результатами, тоді як згрупована версія не інформує про характер змін сторінки стосовно пошукового терміну. Оптимальним рішенням є групування результатів пошуку за термінами змін, що дозволяє представити кілька версій сторінки як єдиний, але інформативно насичений результат.

1.2.3. Опис проблеми дослідження

Наведемо опис проблеми: користувачі мають нагальну потребу у візуалізації та пошуку історичних змін веб-сторінок за допомогою веб-архівів. Однак поточні інтерфейси не дозволяють шукати терміни, що змінювалися в часі, а механізми представлення результатів неефективні для досягнення цієї мети.

Наведені вище виклики формують наступні дослідні питання:

1. Яким чином можна забезпечити доступність та зрозумілість змін, що відбуваються на веб-сторінках?

2. Як підвищити ефективність навігації користувачів веб-архівів для перегляду змін з часом?

3 Яким чином агреговані зміни у корпусі веб-сторінок можуть бути обчислювально використані для надання переконливих доказів редакційних намірів?

1.3. Людська когніція та електронний пошук інформації

Основний внесок цієї роботи полягає в оптимізації електронного пошуку інформації (ІП) у веб-архівах. Перед деталізацією архітектури пошуку, необхідно встановити, як людська когніція (розуміння) впливає на поведінку пошуку інформації (ІП) та забезпечує її підтримку. Розуміння цих

закономірностей є критично важливим для дизайну інтерфейсу пошуку тексту змін.

1.3.1. Вплив когнітивних процесів на дизайн пошукових систем

Процес пошуку інформації моделюється як навчання, яке розпочинається з визначення невідомого, формулювання запитання, а подальша відповідь генерує нове невідоме.

Дизайн пошукових систем має враховувати значний вплив людської когніції, зокрема:

1. Особиста інформаційна інфраструктура.

На поведінку користувача під час пошуку впливають попередні ментальні моделі систем ІП та способи моделювання знань (включно з навичками висновування, організації та метакогнітивними навичками, такими як планування).

2. Обмеження пам'яті.

Робоча пам'ять людини має обмежену ємність порівняно з довготривалою. Надмірна новизна в інтерфейсі ІП-системи виснажує робочу пам'ять, відволікаючи ресурси від основного завдання — обробки знань. Системи повинні мінімізувати когнітивне навантаження, пов'язане з освоєнням інтерфейсу.

3. Узгодження комунікації.

Існує дисбаланс між неявною та контекстуальною людською комунікацією та потребою комп'ютерів у явних командах. Користувачі прагнуть до відкритих пошукових завдань, які повинні бути перетворені на серію закритих, дискретних команд для комп'ютера.

4. Специфічні потреби користувачів.

Різні потреби в пошуку інформації вимагають різних завдань з різними когнітивними вимогами. Ефективність системи ІП прямо залежить від її здатності підтримувати кожен тип завдання. Ідеальний дизайн є ітеративним

і здатний підтримувати користувачів у вирішенні завдань, які вони ще не артикулювали.

5. Точність і повнота.

Користувачі прагнуть до точності (результати відповідають завданню) та повноти (відсутність пропущених релевантних результатів). У системах ІП оптимізація одного показника часто відбувається за рахунок іншого.

1.3.2. Підтримка когнітивних функцій

Системи ІП можуть використовувати когнітивні особливості для підвищення ефективності:

1. Попередня уважна обробка (Pre-attentive Processing).

Виділення результатів пошуку (наприклад, підсвічування) допомагає користувачам у попередній обробці. Використання нативних команд операційної системи знижує когнітивне навантаження.

2. Дизайн інтерфейсу.

Системи повинні підтримувати зміну гранулярності колекції та надавати засоби для моделювання перегляду документів. Основний дисплей має сприяти швидкій ідентифікації тематики документа.

Теорія інформаційного фуражування пояснює, як люди знаходять інформацію в умовах обмеженого часу. Користувачі свідомо чи підсвідомо оцінюють корисність певної дії (наприклад, перехід за посиланням) порівняно з її впливом на досягнення кінцевої мети. Ця оцінка релевантності здійснюється через інформаційні сліди (метадані). Для ефективного пошуку інформація спочатку має бути кластеризована, а користувачі застосовують загальні стратегії збору інформації, визначаючи, де шукати в першу чергу.

1.4. Основи функціонування веб-архітектури та веб-архівів

Основний внесок цієї роботи полягає у відображенні та пошуку текстових змін на веб-сторінках. Веб-архіви є ключовим сховищем

історичних версій веб-сторінок. Для успішної імплементації бек-енду системи пошуку текстових змін необхідно глибоко розуміти архітектуру та функціональні механізми цих архівів.

Концептуально, "Живий Веб" представляє навігаційну колекцію найновіших версій веб-сторінок, тоді як "Минулий Веб" (Past Web) складається з історичних версій, збережених у веб-архівах.

Ці архіви містять значний потенціал для аналізу динаміки змін контенту з часом.

1.4.1. Структура веб-архівів

Існує різноманіття веб-архівів, включаючи національні, засновані на передплаті, бібліотечні та комплексні (наприклад, Wayback Machine від Internet Archive).

Протокол Memento є стандартизованим HTTP-протоколом узгодження вмісту, який слугує для зв'язку між живим та минулим вебом. Оригінальний ресурс у живому вебі ідентифікується як URI-R (Original Resource).

Для запиту історичної версії (так званого "мементо") користувачі повинні вказати як URI-R, так і бажану дату та час. Memento-сумісні сервери повертають архівовану версію, яка є найближчою до запитаної дати.

Кожна архівована версія має власний прямий URI (URI-M, або URI Memento), також відомий як "захоплення" або "знімок". Структура URI-M залежить від конкретного архіву. Деякі архіви (наприклад, Internet Archive, Archive-It) включають URI-R та дату/час у прозорій формі, тоді як інші (наприклад, Archive Today) використовують більш непрозорі, короткі ідентифікатори.

Нижче наведено лістинг 1.1 який ілюструє HTTP-запит з використанням заголовка Accept-Datetime до TimeGate (точки доступу Memento) Wayback Machine та відповідну HTTP-відповідь, яка перенаправляє на найближче доступне мементо.

Лістинг 1.1. Модифікований HTTP-запит та відповідь коли сервер відповідає протоколу Memento

```
curl -I -v -H "Accept-Datetime: Sun, 23 Mar 2025 00:00:00 GMT"
http://web.archive.org/web/http://epa.gov/acidrain/

HTTP/1.1 302 FOUND
Date: Wed, 15 Oct 2025 09:26:06 GMT
x-archive-redirect-reason: found capture at 20250322000840
location:
http://web.archive.org/web/20250322000840/https://www3.epa.gov/acidrain/
HTTP/1.1 200 OK
Date: Wed, 15 Oct 2025 09:26:06 GMT
memento-datetime: Sat, 22 Mar 2025 00:08:40 GMT
link: <https://www3.epa.gov/acidrain/>; rel="original",
<http://web.archive.org/web/timemap/link/https://www3.epa.gov/acidrain/>;
rel="timemap"; type="application/link-format",
<http://web.archive.org/web/https://www3.epa.gov/acidrain/>; rel="timegate",
<http://web.archive.org/web/19970420085456/http://www.epa.gov:80/acidrain/>;
rel="first memento"; datetime="Sun, 20 Apr 1997 08:54:56 GMT",
<http://web.archive.org/web/20250321235530/https://www3.epa.gov/acidrain>;
rel="prev memento"; datetime="Fri, 21 Mar 2025 23:55:30 GMT",
<http://web.archive.org/web/20250322000840/https://www3.epa.gov/acidrain/>;
rel="memento"; datetime="Sat, 22 Mar 2025 00:08:40 GMT",
<http://web.archive.org/web/20250329221858/http://www.epa.gov/acidrain>;
rel="next memento"; datetime="Sat, 29 Mar 2025 22:18:58 GMT",
<http://web.archive.org/web/20251009051617/https://www.epa.gov/acidrain>;
rel="last memento"; datetime="Wed, 09 Oct 2025 05:16:17 GMT"
```

URI TimeMap (URI-T) ідентифікує перелік усіх архівованих версій певної веб-сторінки на конкретному сервері. TimeMaps можуть надаватися у форматах Link, JSON або HTML.

Окрім протоколу Memento, що надає API для переліку захоплень, Internet Archive пропонує доступ до своїх CDX (Crawl Index) API. Індекси CDX, хоча і не є частиною стандарту Memento, містять додаткову інформацію (наприклад, часові мітки сканування та код стану сторінки), яка відсутня у TimeMap. Важливо відзначити, що час сканування відрізняється від дати редагування сторінки, і архів може не містити всіх версій сторінки, якщо частота сканування нижча за частоту змін.

Для узгодження ідентичних веб-сторінок, доступних за декількома схожими URL-адресами, використовується техніка Sort-friendly URI Reordering Transform (SURT). SURT нормалізує URL-адреси до єдиного "дружнього до сортування" ключа, що дозволяє коректно групувати та

знаходити версії, незважаючи на варіації в протоколах, піддоменах чи портах (рис 1.3).

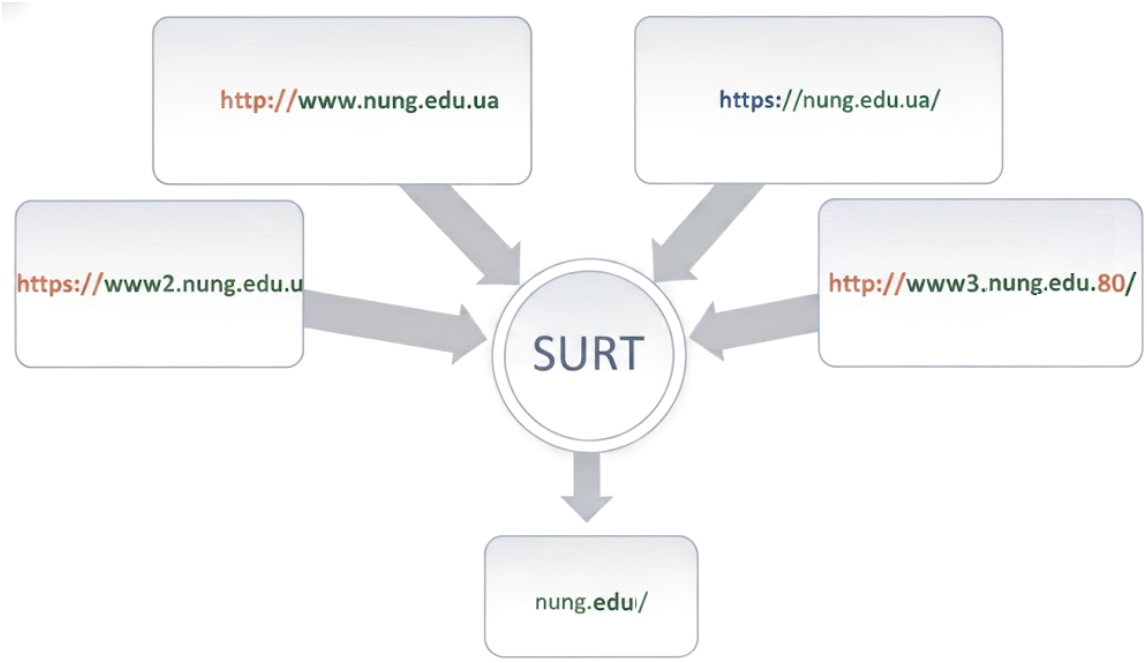


Рис. 1.3. SURT (Sort-friendly URI Reordering Transform) використаний для зіставлення множинних версій URL-адреси

1.4.2. Функціональність відтворення веб-архівів

Веб-архіви зберігаються у форматі файлів Web ARChive (WARC). Файли WARC агрегують записи, що містять повні HTTP-заголовки відповідей, HTML-документи та вбудовані ресурси (зображення, CSS, JavaScript), забезпечуючи можливість відтворення історичного досвіду перегляду. Записи WARC включають необхідні метадані та дату сканування ресурсу.

Провідним інструментом з відкритим кодом для відтворення файлів WARC є PyWB. Для коректної емуляції відтворення, система повинна переписати всі посилання на вбудовані ресурси зі своїх URI-R на відповідні URI-M з датою, найближчою до дати відтворення основної сторінки.

Явище пошкодження мементо (memento damage) виникає, коли не всі ресурси сторінки архівуються, що призводить до неповного відтворення

сторінки. Відсутність ресурсів (наприклад, зображень чи CSS) спотворює зовнішній вигляд сторінки на момент її архівування.

1.4.3. Перегляд змін у веб-архівах

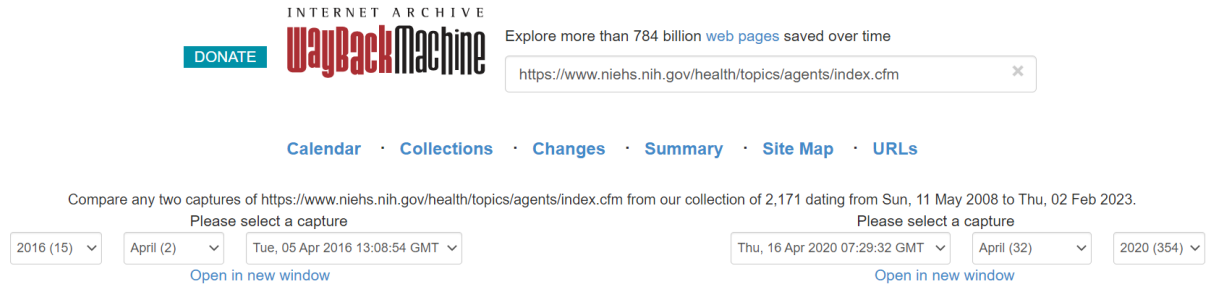
Хоча поточні пошукові системи веб-архівів не інтегрують інформацію про зміни у свої результати, існують окремі інструменти для перегляду змін на відомих сторінках.

Таблиця 1.1.

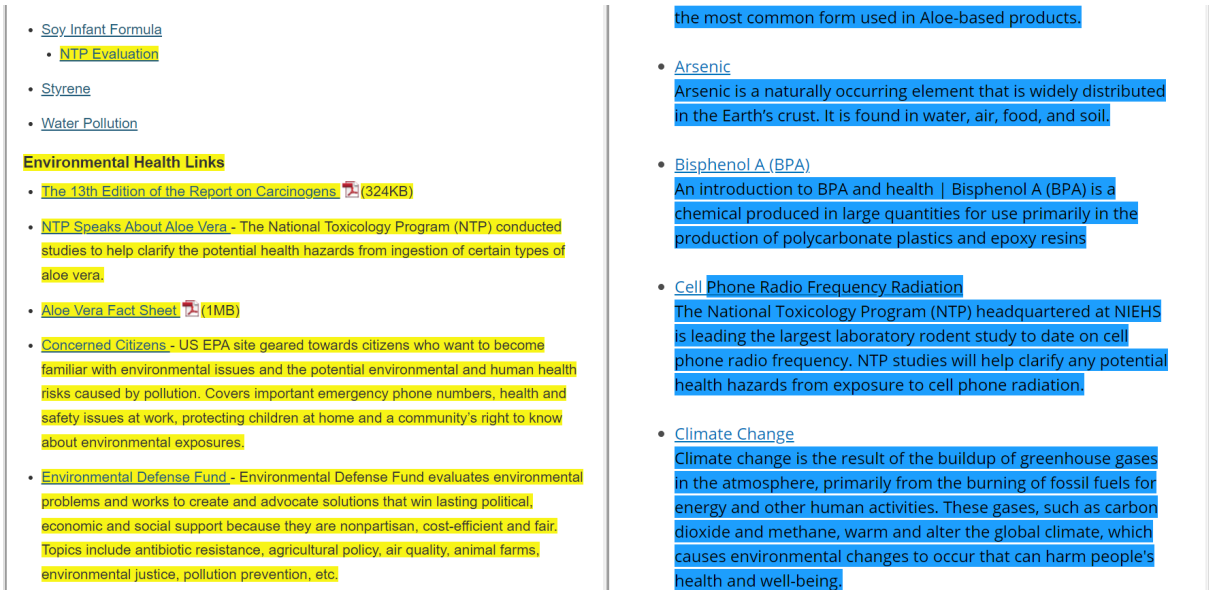
Інструменти перегляду змін

Інструмент	**Функціональність**	**Обмеження**
GLAM Workbench	Виявлення змін в історії версій сторінки; виділення терміну в контексті.	Не індексує контент; пошук лінійний (повільний); працює для одного URL; відсутність контексту попередніх версій.
WikiBlame	Пошук змін на конкретній сторінці Wikipedia.	Не індексує контент; обмежується однією сторінкою за раз.
Інструмент Differences (Wikipedia)	Перегляд змін між версіями в статичному контексті.	Обмежений функціональністю Wikipedia.
Інструмент Changes (Wayback Machine)	Порівняння двох архівних копій (захоплень).	Користувач повинен знати дві дати для порівняння; відсутність інтегрованої функції пошуку змін; порівнює лише версії з одного архіву; відображає лише статичний контекст.

Інструмент Changes Internet Archive (рис. 1.4) побудований на основі набору Web-Monitoring-Diff. На поточний момент він не пов'язаний з жодним інструментом пошуку змін і дозволяє порівнювати лише дві версії одночасно, представлені у статичному форматі поруч.



а)



б)

Рис. 1.4. Інструмент Changes Wayback Machine від Internet Archive показує користувачам різницю між двома архівними копіями

Даний інструмент працює лише для копій у цьому конкретному веб-архіві і відсутній механізм пошуку для знаходження дат та часу змін.

На рисунку 1.4 а інструмент Changes Wayback Machine вимагає, щоб користувач знав дату і час обох версій сторінки для створення порівняння. На рисунку 1.4 б Changes Wayback Machine допомагає користувачам досліджувати додавання (синім кольором) та видалення (жовтим кольором). Термін pollution (забруднення), позначений жовтим кольором ліворуч, був видалений.

Таким чином, існує суттєвий пробіл у функціональності, оскільки жоден з наявних інструментів не забезпечує повнотекстового пошуку змін у

контексті колекції пов'язаних веб-сторінок та не підтримує порівняння версій з множинних архівів.

Висновки до розділу

У першому розділі проаналізовано сучасний стан проблеми відстеження змін у веб-сторінках і визначено її наукову та практичну актуальність. Виявлено, що більшість існуючих веб-архівів не забезпечують зручного доступу до історичних версій контенту та не мають розвинених інструментів візуалізації змін. Проведене дослідження когнітивних аспектів пошуку показало, що ефективність користувацької взаємодії залежить від дизайну інтерфейсу та способів подання змін. Окреслено ключові технологічні принципи побудови веб-архівів і визначено їхні структурні обмеження. У результаті встановлено, що необхідне створення нових моделей і методів архівування, здатних підтримувати аналіз і відображення динаміки веб-контенту.

РОЗДІЛ 2. ДОСЛІДЖЕННЯ МЕТОДІВ ТА АЛГОРИТМІВ СЕРВІСІВ АРХІВУВАННЯ З ВІДСТЕЖЕННЯМ ЗМІН В ВЕБ-СТОРІНКАХ

2.1. Архітектура пошуку та особливості пошукових систем веб-архівів

2.1.1. Основні компоненти пошукової системи

Для забезпечення ефективного запитування документів необхідно, щоб текстова інформація була витягнута та перетворена на певну базу даних. Базою даних, що переважно використовується для швидкого пошуку, є інвертований індекс, який дозволяє здійснювати пошук за термінами, а не послідовно переглядати кожен документ.

Процес створення індексу включає токенізацію (розбиття тексту на терміни) вмісту кожного документа. Ефективна архітектура пошуку передбачає:

- Бек-енд-платформу для виконання запитів до інвертованого індексу.
- Фронт-енд-інтерфейс, що дозволяє користувачам надсилати запити та отримувати результати у визначеному порядку.

2.1.2. Набір пошукових інструментів Apache

Рішення з відкритим кодом від Apache є галузевим стандартом для високопродуктивного пошуку.

Apache Lucene є високопродуктивною пошуковою системою, що реалізує інвертований індекс з оптимізаціями (наприклад, списки пропусків) і формує основу для пошукової платформи.

Основою Lucene є архітектура інвертованого індексу. На відміну від традиційних баз даних, які зіставляють документи з їхнім вмістом, інвертований індекс зіставляє терміни (токени) з документами, які їх містять, а також із відповідними метаданими (наприклад, частотою терміна та його

позицією в документі). Ця структура даних забезпечує надзвичайно швидке вилучення документів за запитом.

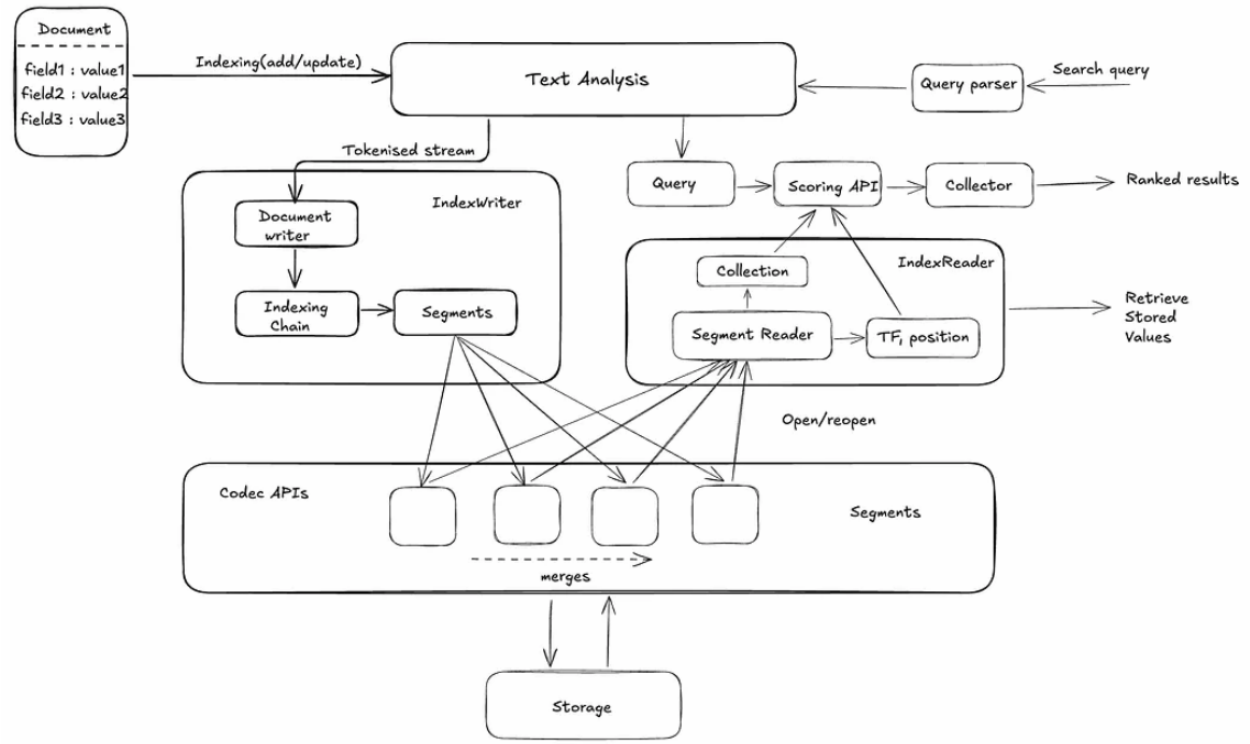


Рис. 2.1. Принцип роботи системи Apache Lucene

Індексація в Lucene включає наступні кроки:

- Токенізація. Вихідний текст документа розбивається на окремі токени (терміни) згідно зі стандартом Unicode Annex #29 (UAX #29).

- Аналіз. До токенів застосовуються різні процеси очищення та нормалізації, як-от перетворення на нижній регістр, видалення стоп-слів (загальновживаних слів) та стемінг (зведення слів до основи).

- Створення постингів. Для кожного унікального терміна створюється запис (постинг), що містить ідентифікатори документів, у яких він зустрічається, частоту його вживання та позиційні дані (що необхідні для пошуку фраз).

Apache Solr — це повнофункціональна пошукова платформа, побудована на базі Lucene. Solr є повноцінним додатком, що працює як окремий пошуковий сервер і надає розширені функції, які виходять за рамки

низькорівневих можливостей Lucene, зокрема кластеризацію, розподілений пошук та зручні інтерфейси.

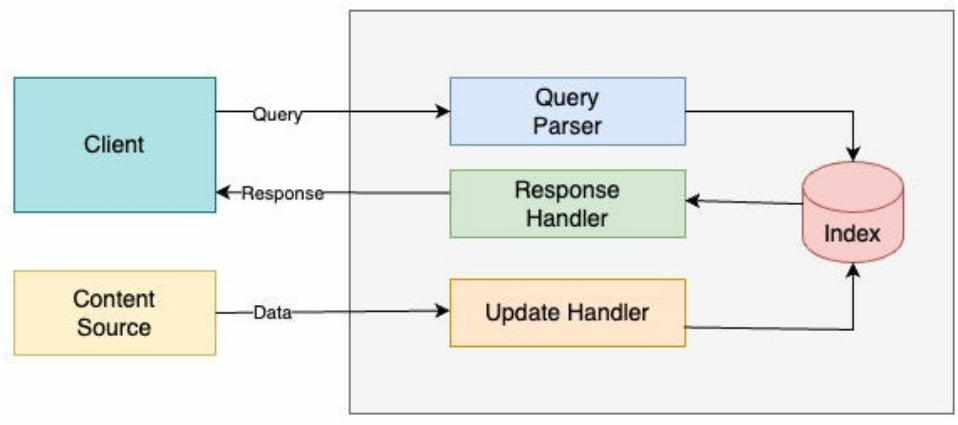


Рис. 2.2. Спрощена архітектура Solr

Solr трансформує бібліотеку Lucene (яка є лише механізмом індексації та пошуку) у повнофункціональний пошуковий сервіс, доступний через стандартні протоколи, насамперед HTTP та формати даних XML/JSON.

Apache Tika використовується для вилучення тексту та метаданих із різних форматів файлів для подальшого індексування Lucene. Основна мета Apache Tika — забезпечити єдиний програмний інтерфейс (API) для роботи з різноманіттям цифрових форматів, усуваючи необхідність для розробника писати окремий парсер для кожного типу файлу.

Tika реалізована як модульний набір бібліотек, що дозволяє легко додавати підтримку нових форматів.

Tika виступає як передобробник (pre-processor) даних для систем інформаційного пошуку:

Сирий Документ → Apache Tika (Парсинг) → Чистий Текст та Мета дані → Apache Lucene/Solr (Індексація)

Без Tika, пошукові системи могли б індексувати лише простий текст або HTML. Tika забезпечує їх вмістом із "глибоких" форматів, таких як PDF або DOCX.

Окрім основної Java-бібліотеки, Tika пропонує інструменти для командного рядка та графічного інтерфейсу, що дозволяє користувачам тестувати вилучення контенту без програмування.

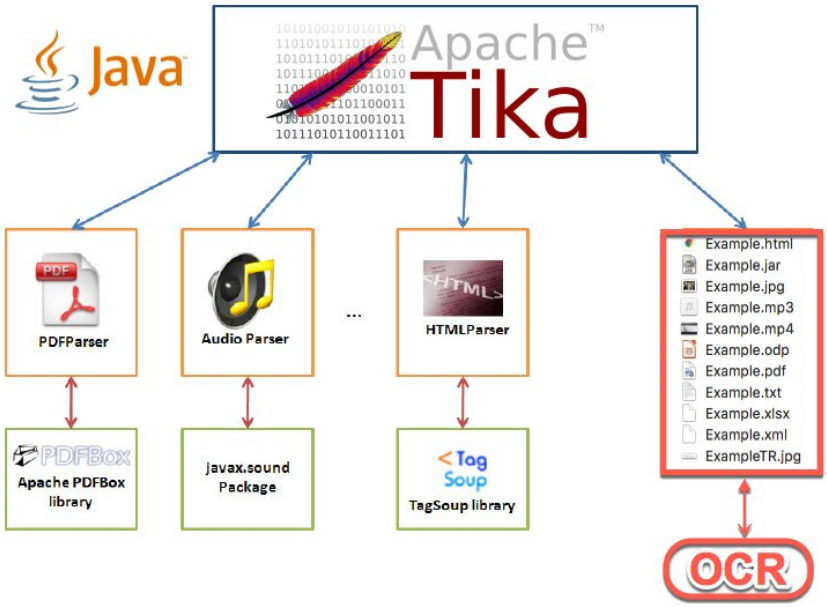


Рис. 2.3. Короткий перелік форматів що підтримуються Apache Tika

Solarium — провідна клієнтська бібліотека PHP для Solr, яка може бути використана для розробки користувацького інтерфейсу, що транслює запити користувачів у формальний синтаксис запитів Lucene. Її основне призначення — надати розробникам PHP-додатків інтуїтивно зрозумілий, високорівневий API, що точно моделює концепції Solr, дозволяючи їм абстрагуватися від низькорівневої HTTP-комунікації та складної побудови параметрів запитів.

Solarium прагне точно відображати внутрішні механізми Solr. Наприклад, замість того, щоб мати окремі запити для додавання, видалення та фіксації, Solarium об'єднує їх в один об'єкт Update Query, що відображає роботу Update Handler в Solr, оптимізуючи мережевий трафік.

В цілому, Solarium є проміжним шаром між PHP-додатком і пошуковим сервером Apache Solr, що значно підвищує продуктивність розробки, знижує ймовірність помилок і дозволяє ефективно використовувати складні функції Solr.

2.1.3. Робочий процес індексації

Під час індексації текстовий вміст документів піддається токенізації на терміни. Токенізація в Lucene відповідає стандарту Unicode Annex #29. Крім того, для підтримки пошуку фраз, в індексі зберігаються позиції термінів у документі. Токенізація також забезпечує функціональність виділення (highlighting) — підсвічування пошукових термінів у текстових фрагментах, що відображаються у результатах пошуку.

Solrwayback є прикладом набору інструментів для пошуку у веб-архівах, що включає Lucene, Solr, Apache Tika та індексатор UK Web Archive Discovery.

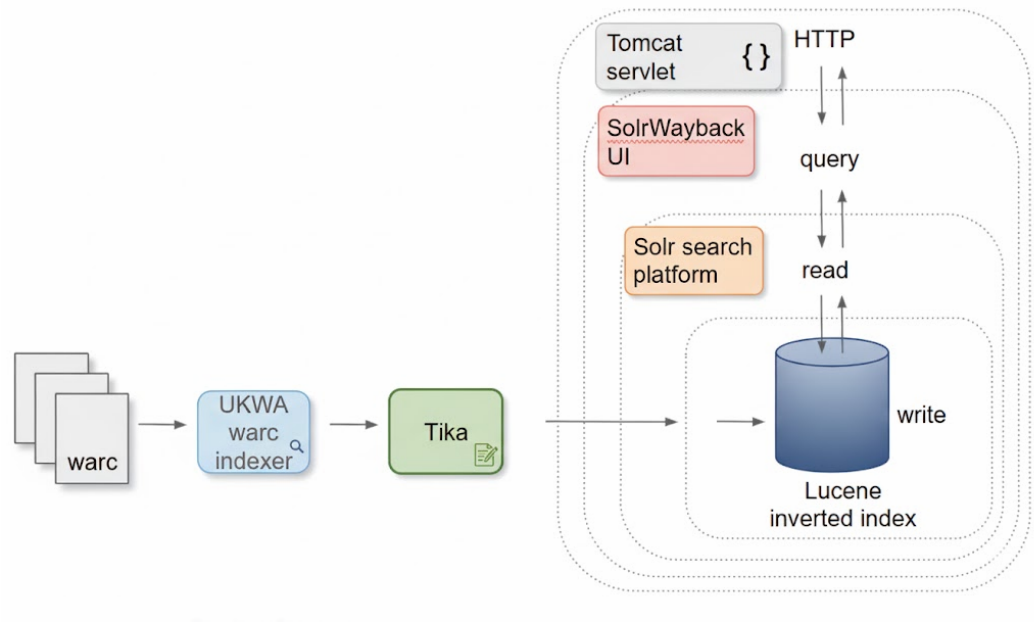


Рис. 2.4. Робочий процес Solrwayback

Робочий процес індексації (рис. 2.4) передбачає:

- Індексація WARC-файлів. Оскільки Apache Tika не має вбудованої підтримки WARC, використовується індексатор UK Web Archive Discovery, який витягує текст із записів WARC і передає його для подальшої індексації. Він також може включати опцію видалення шаблонів (наприклад, файлів JavaScript) для оптимізації індексу.

- Виконання запитів. Користувачі взаємодіють через інтерфейс, де запити інтерпретуються Solr, який звертається до індексу Lucene, і повертає результати.

Оскільки веб-архіви є тимчасовими за своєю природою, бек-енд повинен підтримувати тимчасові запити. Lucene та Solr підтримують індексацію діапазонів дат документів та виконання запитів за цими діапазонами з різною гранулярністю.

Користувачі веб-архівів потребують можливості знаходити та переглядати зміни на веб-сторінках. Проте, наявні інтерфейси не підтримують це завдання. У системах з повнотекстовим пошуком:

- Кілька версій сторінки, що відповідають запиту, показуються окремо (створюючи безлад) без переліку змін.

- Версії групуються таким чином, що приховують зміни.

Internet Archive, дотримуючись принципу "зберегти все", архівував понад 900 мільярдів веб-сторінок. Проте, більшість цих копій не індексуються повним текстом, змушуючи користувачів знати URL для пошуку. Такий вибір, хоча і максимізує обсяг матеріалу, знижує доступність інформації.

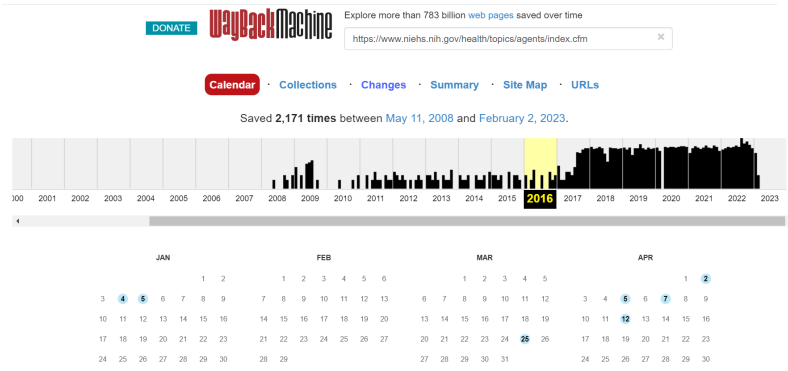
2.1.4. Існуючі інтерфейси пошуку у веб-архівах

Повнотекстовий пошук визначено як високо затребувану функцію. Однак багато архівів надають лише пошук за URI (рис. 2.5), вимагаючи від користувача знання точної адреси.

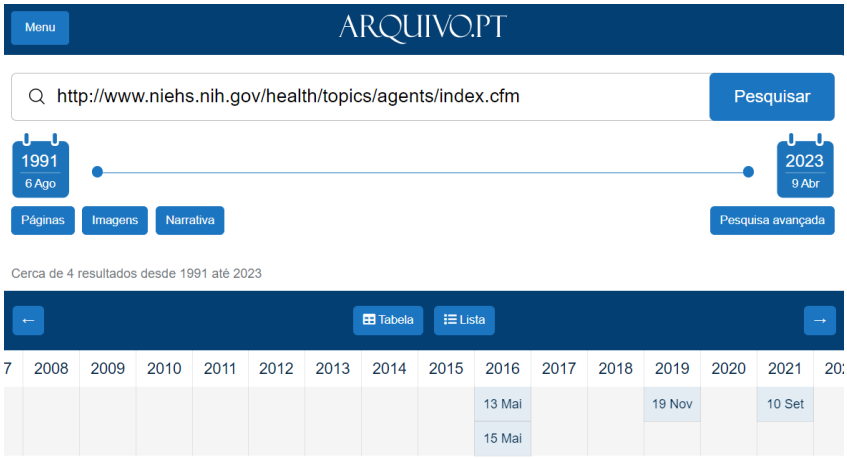
Деякі інтерфейси пошуку URI (наприклад, Archive Today, рис. 2.6) надають мініатюри, які можуть допомогти виявити основні візуальні зміни, але не надають інформації про зміни тексту.

Веб-архіви, такі як португальський вебархів (Arquivo.pt), пебархів Великої Британії та деякі колекції Archive-It, підтримують повнотекстовий пошук. Їхні форми запитів (рис. 2.7) дозволяють фільтрувати за полями, як-от

домен або дата. Критичним недоліком є те, що жоден з інтерфейсів не має функціональності для запиту видаленого терміну.



а) Пошук URI у Wayback Machine (Internet Archive)



б) Пошук URI у Portuguese Web Archive

Рис. 2.5. Два приклади пошуку URI в веб-архіві

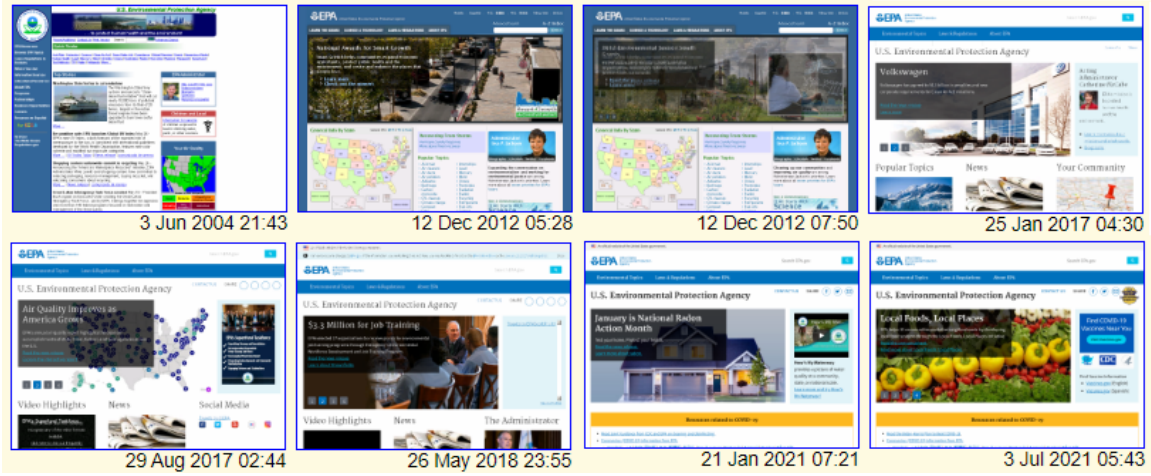


Рис. 2.6. Пошук URI на Archive Today із мініатюрами, що демонструють основні зміни на сторінці з часом

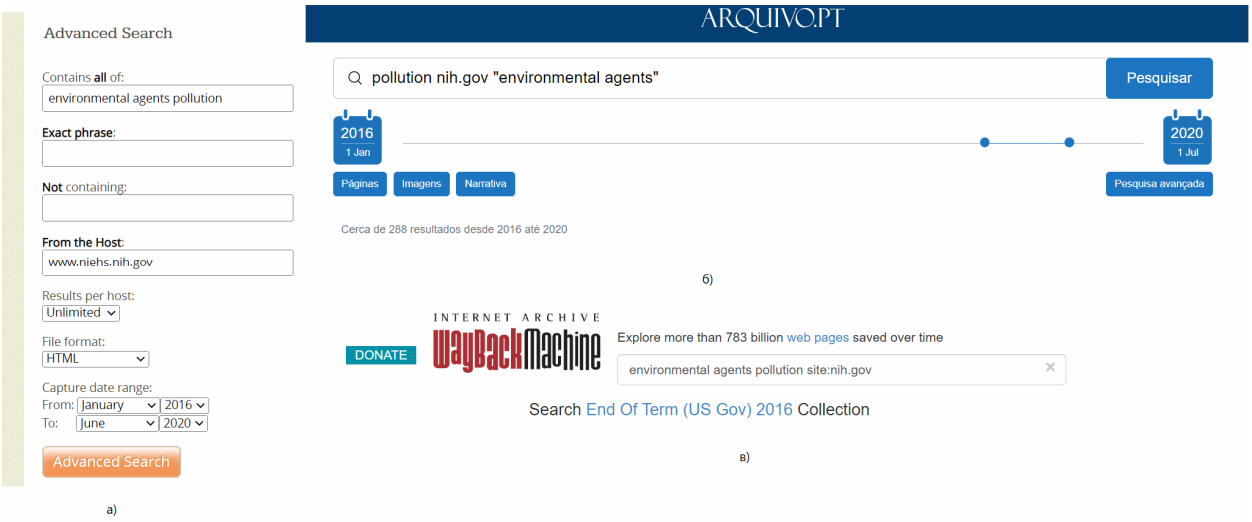


Рис. 2.7. Форми запитів для повнотекстового пошуку по архівах

Деякі веб-архіви підтримують повнотекстовий пошук із різноманітними можливостями фільтрації по невеликих колекціях або по всьому веб-архіву. Жоден із наведених пошукових інтерфейсів не дозволяє користувачам шукати терміни, які були видалені з вебсторінок.

На рис. 2.7 а показано загальне поле пошуку для Archive-It, на рис 2.7 б - поле пошуку по всьому Веб-Архіву, що дозволяє повнотекстовий пошук та фільтрацію за датою.

На рис 2.7 в показано поле пошуку у Wayback Machine (Internet Archive) для колекції "End of Term Archive 2016", що дозволяє повнотекстовий пошук та фільтрацію за сайтом, але лише по одній колекції закінчення терміну одночасно.

Проблема відображення кількох версій однієї сторінки залишається невирішеною (рис. 2.8):

- Archive-It показує кожну відповідну версію окремо, що призводить до надмірного безладу (клаттеру).

- Arquivo.pt дозволяє обмежувати максимальну кількість версій, що зменшує безлад, але може призвести до пропуску релевантних версій.

- Пошук колекції Wayback Machine показує лише одну версію сторінки, зменшуючи безлад, але приховуючи інші релевантні версії та зміни між ними.

Отже, жоден з існуючих інтерфейсів повнотекстового пошуку у веб-архівах не вирішує проблему ефективного представлення інформації про зміни. Версії або групуються без явної деталізації змін, або включаються окремо, що не дозволяє користувачеві шукати чи переглядати зміни в контексті часу.

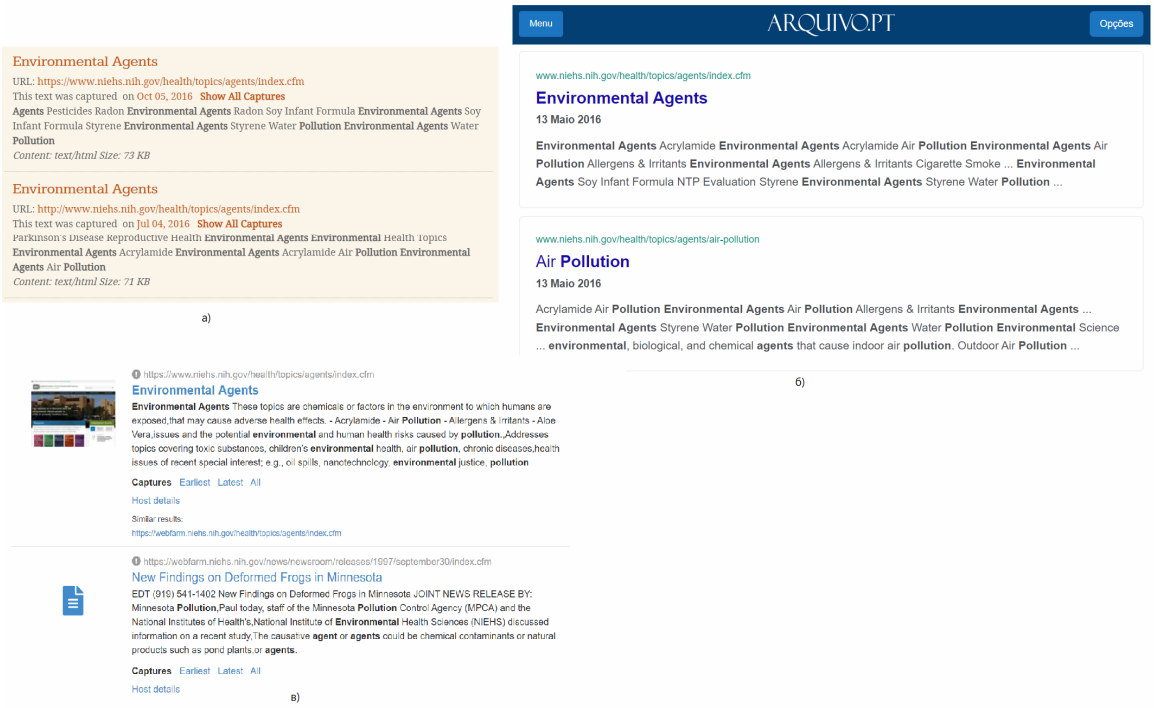


Рис. 2.8. Три пошукові інтерфейсів веб-архівів які не групують версії, щоб показати зміни на сторінці з часом

а) Сторінка результатів пошуку (SERP) колекції Archive-It для запиту з рисунка 2.7 а, що відображає заголовок, URI, дату, текстовий фрагмент (snippet), метадані, посилання на відтворення (replay link) та посилання на додаткові збереження (additional captures link). Дві версії сторінки показані індивідуально.

б) SERP Arquivo.pt для запиту з рисунка 2.7 б, що відображає URI, заголовок, дату, текстовий фрагмент і посилання на відтворення.

в) SERP колекції Wayback Machine для запиту з рисунка 2.7 в, що відображає URI, заголовок, текстовий фрагмент, знімок екрана (screenshot), посилання на відтворення та посилання на додаткові збереження.

2.2. Набори даних для аналізу змін у вебсторінках та запитів

Представлене дослідження, зосереджене на розробці бекенду та фронтенду системи пошуку текстових змін, вимагає валідації реалізації на основі репрезентативного набору даних вебсторінок з фіксованими змінами. Більшість існуючих колекцій архівованих вебсторінок містять лише одну версію сторінки, що унеможливлює проведення часового аналізу змін. Набори даних, що включають множинні версії, зазвичай формувалися з метою моніторингу. У цьому підрозділі представлено огляд наборів даних, потенційно придатних для оцінки ефективності системи пошуку текстових змін.

2.2.1. Загальні набори даних вебсканувань

Академічні дослідження часто послуговуються великими наборами даних вебсканувань, такими як ClueWeb [12] та Common Crawl [18]. Метою цих колекцій є збір знімків великої кількості унікальних URI. Колекції ClueWeb охоплюють 2009, 2012 та 2022 роки, кожна з яких містить від одного до двох мільярдів вебсторінок. Common Crawl, що функціонує з 2008 року, наразі збирається на щомісячній основі. Остання версія Common Crawl (липень 2024 року) містить 2,5 мільярда вебсторінок.

Великі колекції, зокрема ClueWeb та CommonCrawl, здійснюють сканування в різні моменти часу, базуючись на початковому наборі, і не завжди гарантують архівування одних і тих самих сторінок. Переваги цього підходу полягають у індексації нових сторінок, виявлених через нові посилання, та уникненні вже видалених сторінок. Однак, оскільки ці набори даних не були спеціально призначені для збору множинних версій одного URI, відсутня гарантія регулярного сканування будь-якої конкретної сторінки, що ускладнює аналіз змін у часі. Крім того, для релевантності аналізу необхідно, щоб зафіксовані зміни на вебсторінках були значущими.

2.2.2 Вебархів "Кінець терміну" (End of Term Web Archive)

Архівування федеральних вебсайтів, особливо в період завершення терміну повноважень президента, є критично важливим завданням, яке виконується кількома установами. Вебархів "Кінець Терміну" створюється у партнерстві п'яти організацій, включаючи Internet Archive та Бібліотеку Конгресу. Цей вебархів містить функцію повнотекстового пошуку [19], проте кожна колекція "кінця терміну" містить лише один знімок кожного веб-URI. В роботі [15] порівнювали колекції 2008 та 2012 років для виявлення змін у датах сканування та вебадресах, але окремі терміни не підлягали аналізу.

2.2.3 Набір даних EDGI

В роботі [110] відстежували зміни на 30 вебсайтах федеральних екологічних агентств США у період з 2016 по 2020 рік. Вони порівнювали зміни 56 попередньо відібраних екологічних термінів та фраз на 40 000 вебсторінках, використовуючи архівні ресурси Internet Archive. Їхній набір даних включає файл counted_urls.csv, що містить URI сторінок та відповідні мементо 2016 і 2020 років в Internet Archive, зразок якого наведено у Лістингу 2.1. Інший файл, obama_count.csv, містить підрахунок 56 термінів та фраз у кожному мементо 2016 року, як показано у Лістингу 2.2. Файл trump_count.csv містить підрахунок термінів за 2020 рік.

Лістинг 2.1. Файл counted_urls.csv, що відображає спарені мементо, з позначкою NA для відсутнього захоплення.

	url - o	final captured url - t
0	https://www3.epa.gov/enviro/facts/multisystem.html	http://web.archive.org/web/20160612091334id
1	https://www3.epa.gov/enviro/facts/multisystem.html	http://web.archive.org/web/20200101042321id
2	https://www3.epa.gov/enviro/facts/multisystem.html	NA
3	https://www.osha.gov/pls/imis/inspectionNr.html	http://web.archive.org/web/20160322140439
4	https://www.osha.gov/pls/imis/inspectionNr.html	NA

Лістинг 2.2 демонструє структуру CSV-файлу який показує частоти вживання термінів, а значення 999 використовується для позначення

випадків, коли сторінка повернула не 200 HTTP-статус (тобто помилку або недоступність)

Лістинг 2.2. Файл obama_count.csv, що відображає підрахунок термінів у 2016 році, з позначкою 999 для статусу HTTP, відмінного від 200.

```
adaptation,agency mission,air quality,anthropogenic,benefits,brownfield
17,0,1,0,5,0
999,999,999,999,999,999
```

Для ідентифікації того, чи був термін доданий або вилучений у період 2016–2020 років, необхідно завантажити обидва файли підрахунку термінів та порівняти відповідні рядки, зіставляючи значення. Для асоціювання підрахунків термінів з конкретною сторінкою, відповідний рядок у файлі URL має бути співставлений.

У своєму аналізі даних встановили, що приблизно 20% вебсайту EPA було вилучено в період між 2016 та 2020 роками, що обмежувало публічний доступ до екологічної інформації. Вони виявили диференційовані зміни в термінології залежно від типу агентства та глибини сторінки. Зокрема, було встановлено, що деякі ключові терміни, як-от "зміна клімату", були видалені з більшості федеральних екологічних вебсайтів протягом цього періоду.

Оскільки цей набір даних містить зафіксовані (відомі) зміни, він є особливо придатним для оцінки системи пошуку текстових змін. 56 термінів та фраз, визначених EDGI, можуть бути використані для валідації точності алгоритмів, застосованих для розрахунку текстових змін. Крім того, можливе виявлення додаткових термінів та фраз, вилучених на вебсайтах, поза тими, що відстежувалися EDGI, для оцінки повноти індексу текстових змін.

2.2.4 Набір даних ORCAS

Відкритий ресурс для аналізу кліків у пошуку (ORCAS) — це колекція даних, що містить запити Microsoft Bing та відповідні кліки за період 2017–

2020 років. Набір даних використовує k-анонімність, гарантуючи, що кожен запит у колекції був здійснений значною кількістю різних користувачів. Це забезпечує захист конфіденційності користувачів та підтверджує, що запити є популярними. ORCAS є частиною колекції наборів даних Microsoft Machine Reading Comprehension (MS MARCO), включно з набором даних релевантності пасажів. ORCAS буде використаний для встановлення зв'язку між термінами запитів та вилученими термінами для сторінок під час оцінки індексу пошуку текстових змін.

2.2.5. Набір даних журналу запитів архіву (AQL)

Журнал запитів архіву (Archive Query Log - AQL) [20] — це набір даних, створений на основі архівованих сторінок результатів пошуку (SERP), що дозволяє дослідникам вивчати еволюцію запитів та SERP у часі. Набір даних надає доступ до 356 мільйонів запитів, однак лише сім відсотків відповідних SERP включено до колекції. Це обмежує аналіз лише відомими запитами. Наприклад, оскільки всі запити доступні в наборі даних, дослідники можуть отримати відповіді на питання, що стосуються виключно запитів. Вони також можуть завантажити відповідні SERP для аналізу результатів цих запитів. Проте, якщо досліднику необхідно визначити запити для конкретної сторінки результатів, це неможливо зробити за допомогою поточного набору даних. Крім того, не гарантується наявність пов'язаного кліку для сторінки.

Отже, для побудови успішної пошукової системи необхідно визначити ментальні моделі користувачів у вебархівах, завдання, які вони виконують за допомогою цих архівів, та обмеження пам'яті, пов'язані з інтерпретацією змін. Далі розглянуто, як URI та вміст сторінок змінюються з часом, та функціонування кодів стану HTTP. Ці концепції будуть інтегровані в методологію виявлення та візуалізації змін у вебсторінках.

Отже в попередніх пунктах пояснено функціонування протоколу Memento, включаючи URI-M та TimeMaps. Розглянуто, як архівовані коди

стану HTTP можуть бути отримані з файлів CDX. Представлено використання SURT для канонізації URL-адрес. Файли WARC представлені як контейнери для архівованих вебсторінок. PyWB введено як інструмент для відтворення архівованих вебсторінок. Введено концепцію пошкодження мементо, що впливає на можливість перегляду минулої версії вебсторінки, та розглянуто інструмент Changes від Wayback Machine. Протокол Memento буде використаний для інтеграції архівованих вебсторінок у пошукову систему. Файли CDX будуть використані для ефективного визначення архівованих кодів стану з метою індексації. SURT буде застосований для узгодження множинних наборів даних. Необхідно ідентифікувати інструмент для індексації файлів WARC. PyWB буде використаний для створення анімованого інструменту різниць. Буде продемонстровано, що деякі користувачі переглядають декілька версій вебсторінок через пошкодження мементо однієї з версій. Технологія, що забезпечує роботу інструменту Changes, буде використана для створення анімованого інструменту різниць.

У підрозділі про архітектуру пошуку представлено Lucene та Solr як встановлені пошукові системи. Пояснено, що індексація вимагає токенізації. SolrWayback представлено як встановлений інтерфейс пошуку вебархівів. Було показано, що не всі вебархіви підтримують повнотекстовий пошук, і що існуючі сторінки результатів пошуку не мають ефективного рішення для групування версій вебсторінок. Lucene, Solr та SolrWayback будуть використані для побудови нашої пошукової системи. Токенізація буде інтегрована на сторінку результатів пошуку для виділення та для анімованого інструменту різниць. Буде створено пошукову систему, що підтримує запити на зміни та групування результатів у зрозумілий спосіб.

Також представлено набір даних EDGI, який містить відомі зміни, та набір даних ORCAS, який містить пари запит/клік. Набір даних EDGI буде використаний для оцінки системи пошуку текстових змін. Набори даних EDGI та ORCAS будуть узгоджені для демонстрації кореляції між змінами термінів та запитами для цих вебсторінок.

2.3. Аналіз поведінки користувачів під час роботи з інструментами перегляду змін

Успішна реалізація ключових компонентів даної роботи — сторінки результатів пошуку, що інтегрує демонстрацію відмінностей, а також двох спеціалізованих інструментів для перегляду змін — вимагає глибокого розуміння поведінки користувачів під час взаємодії з інструментами візуалізації змін. У цьому підрозділі підсумовано емпіричні дослідження, які диференціюють дизайнерські рішення для виявлення (discovery) та перегляду (inspection) змін у різних цифрових середовищах.

2.3.1. Методи презентації порівняння даних

Існують три основні парадигми представлення результатів обчислення відмінностей (diffs) користувачеві:

1. Рядковий формат (Inline View). Демонструються лише безпосередні відмінності у послідовному рядковому форматі (рис. 2.9). Цей підхід є ефективним, коли ступінь схожості між двома версіями високий, і користувач прагне бачити виключно зміни.

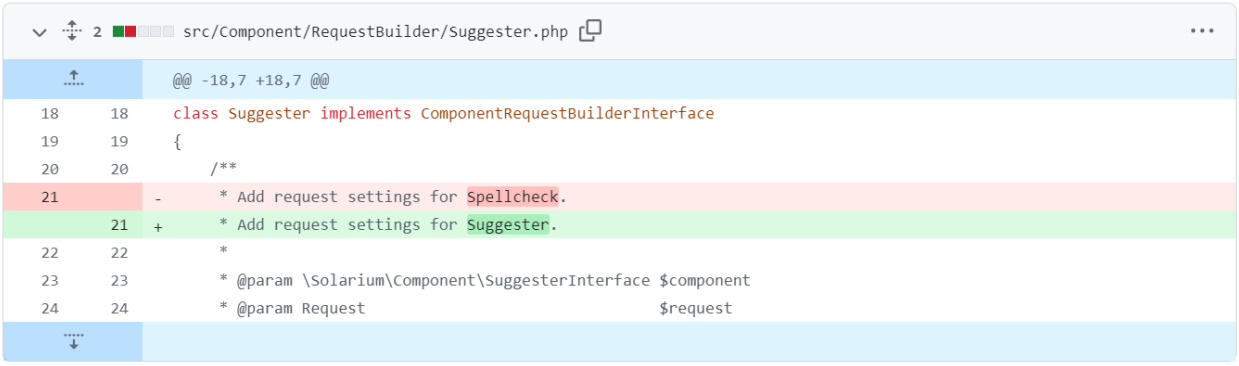


Рис. 2.9. Рядковий перегляд різниць коду Solarium на GitHub. Цей перегляд показує користувачеві кожну зміну на рядках, розташованих послідовно

2. Комбінований перегляд (Unified View). Видалення позначаються одним стилем, а подальше вставлення — іншим, часто розташовуючи їх

близько до місця зміни (рис. 2.10). Хоча цей перегляд є більш компактним, надмірна кількість змін може ускладнити його читабельність.

Рис. 2.10. Комбінований перегляд різниць. Цей перегляд включає всі рядки разом, але кожна індивідуальна зміна показана в контексті рядка.

3. Порівняння пліч-о-пліч (Side-by-Side Juxtaposition). Представлення версій поруч (рис. 2.11). Це дозволяє користувачеві одночасно бачити як змінений, так і незмінний контекст. Недоліком є підвищений час обробки через збільшений обсяг візуальної інформації.

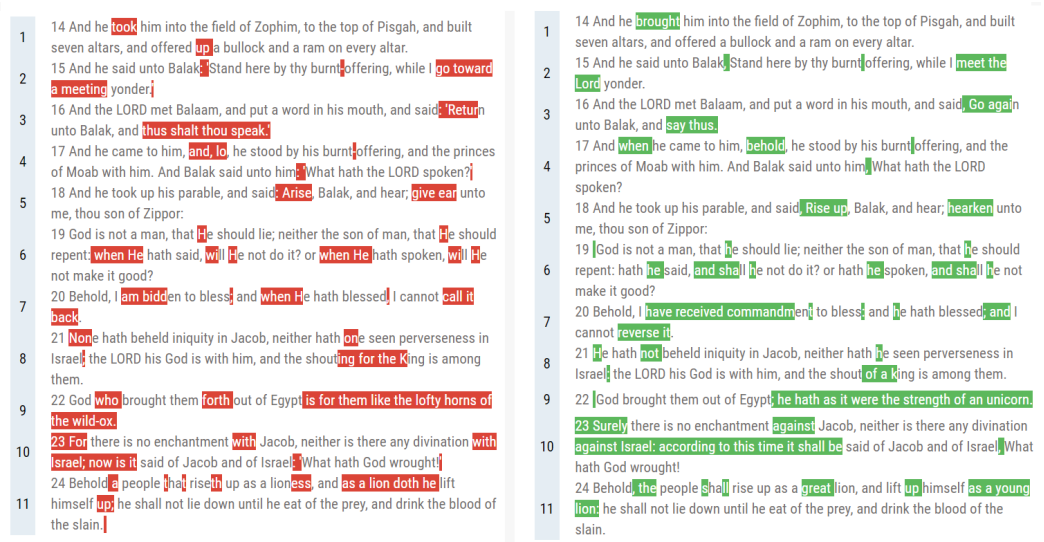


Рис. 2.11. Перегляд відмінностей у форматі пліч-о-пліч

Цей формат (рис. 2.11) відображає одну версію тексту з позначеними видаленнями (deletions) та іншу версію тексту з позначеними додаваннями (additions).

2.3.2. Стратегії візуалізації змін

Визначають три методи візуалізації відмінностей у даних: юкстапозиція (розташування малих множин поруч), накладання (overlay) та явне виведення зміни (explicit change). Для текстових даних явне виведення зміни, яке передбачає осмислене віднімання слів, не застосовується. Отже, статичні візуалізації є або юкстапозицією, або накладанням. Рядковий та комбінований формати є прикладами накладання, тоді як перегляд пліч-о-пліч є прикладом юкстапозиції.

1. Виявлення зміни та візуальна пам'ять.

Пояснює, що виявлення зміни охоплює встановлення факту зміни, а також ідентифікацію, що саме змінилося і де. Ретельний дизайн має вирішальне значення для оптимізації здатності користувачів помічати зміни. Виявлення змін обробляється у візуальній короткостроковій пам'яті. Стверджується, що попередня обробка (preattentive processing) є основною дизайнерською технікою, що використовується для швидкого виявлення змін, зокрема в ситуаціях оцінювання.

2. Розрізнення динамічної та завершеної зміни.

Дослідники можуть вимірювати виявлення змін, реєструючи час реакції при різних інтервалах між стимулами. Користувачі здатні розпізнати факт того, що зміна відбулася, за короткий проміжок часу. Проте, для розпізнавання відмінності між початковим і наступним станом, користувачам потрібен довший інтервал. Це розрізнення часто називають динамічною зміною (актом зміни в процесі) проти завершеної зміни (визначенням того, що зміна відбулася).

3. Стратегії анімації для представлення змін.

Також визначають кілька стратегій анімації, що можуть бути використані для представлення змін:

- Залежні від часу (Gap-dependent, Saccade-dependent, Blink-dependent): Використовують часові проміжки для розділення версій.

- Залежні від руху (Shift-dependent, Cut-dependent): Зсувають сцену або представляють іншу її перспективу.

Залежні від відволікання (Spot-dependent, Closure-dependent): Виділяють зміни шляхом навмисного введення відволікаючих факторів або блокування елемента, що змінюється.

- Поступова зміна (Gradual Change): Створює плавний перехід між станами протягом кількох секунд.

- Повторення (Repetition): Може використовуватися разом із будь-якою з перелічених стратегій, коли анімація повторюється до моменту розпізнавання зміни користувачем.

Альтернативою статичній візуалізації комбінованого перегляду є анімація кожної відмінності окремо. Перевага цієї модифікації полягає в тому, що анімація може допомогти користувачеві засвоїти контекст змін, включаючи їхнє місцеположення, зміст та обсяг.

Висновки до розділу

Другий розділ присвячено аналізу методів і алгоритмів, що лежать в основі сервісів архівування з відстеженням змін у веб-сторінках. Розглянуто архітектуру пошукових систем і компоненти, що забезпечують ефективне індексування та обробку запитів на основі технологій Apache Lucene і Solr. Проаналізовано наявні набори даних для дослідження змін у веб-контенті, серед яких End of Term Web Archive, EDGI, ORCAS та AQL. Встановлено, що комбінація даних запитів і текстових версій сторінок дає змогу точніше виявляти зміни й покращує релевантність результатів пошуку.

РОЗДІЛ 3. ПРЕДСТАВЛЕННЯ АРХІТЕКТУРИ ТА МЕТОДОЛОГІЇ ПОШУКУ ЗМІН У ВЕБ-СТОРІНКАХ ТА АРХІВАХ

У цьому розділі представлено методологічну базу та архітектуру, розроблену для вирішення перших двох запитань магістерського дослідження які показано в першому розділі.

Спочатку ми розглядаємо запитання 1: як зробити зміни на вебсторінках помітними та зрозумілими? Ми деталізуємо необхідні конструкції для бекенду такої системи та описуємо реалізацію її фронтенду, який включає інструмент анімованого відтворення змін. Далі розглянемо друге питання: як можна підвищити ефективність навігації користувачів вебархівів для перегляду змін з часом? Ми опишемо додаткову функціональність навігаційних кнопок, що забезпечують пропуск ідентичних версій, а також представляємо нову сторінку призначення для архівованих перенаправлень.

3.1. Архітектура пошукової системи змін у тексті

Архітектура пошукової системи для змін тексту є трирівневою, як схематично зображено на рис. 3.1:

- Рівень 1 охоплює отримання документів (Document Acquisition).
- Рівень 2 включає документи та індекси (Documents and Indices).
- Рівень 3 складається з користувацького інтерфейсу (User Interface).

Ця трирівнева архітектура являє собою типову багаторівневу (multi-tier) модель для пошукової системи:

Рівень отримання (Level 1: Acquisition) відповідає за збір, обробку та підготовку вихідних даних (наприклад, WARC-файлів або мементо) для індексації.

Рівень обробки та зберігання (Level 2: Documents and Indices) включає самі дані та механізми їхньої індексації та аналізу (наприклад, Lucene/Solr) для обчислення та зберігання інформації про зміни.

Рівень представлення (Level 3: User Interface) відповідає за взаємодію з користувачем, відображення результатів пошуку змін та надання інструментів для їх візуалізації.

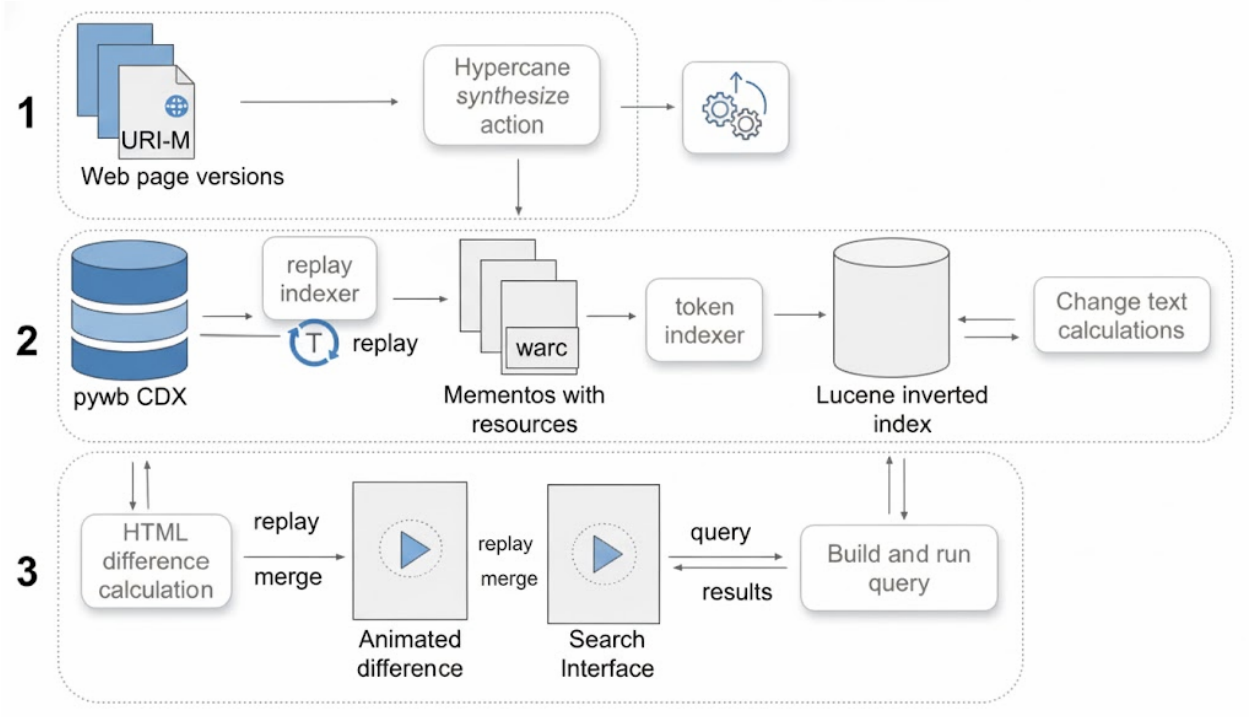


Рис. 3.1. Архітектура пошукової системи для змін у тексті

3.1.1. Етап отримання документів

Документи для даної пошукової системи мають бути у форматі WARC (Web Archive). Хоча користувачі можуть відтворювати публічні архівні записи, вони зазвичай не мають прямого доступу до вихідних WARC-файлів. Індексування цих публічних записів є неможливим без інструменту, здатного конвертувати URI-M (Memento URI) у WARC-файл.

Hypercane — це інструмент для взаємодії з колекціями веб-архівів, одна з функцій якого полягає у синтезі WARC-файлів на основі наданого URI-M.

3.1.2. Solr для версійних колекцій документів

Будь-яка колекція документів, включаючи WARC, набуває корисності лише за умови можливості її пошуку та доступу. WARC-файли стають доступними для пошуку після їх індексації. Ми індексували WARC-файли в Lucene. Платформа Solr, побудована на базі Lucene, слугує надійною основою для пошукової платформи WARC. Оскільки Lucene раніше не використовувався для роботи з концепцією тимчасового діапазону дійсності веб-архіву, ми модифікували XML-конфігурацію Solr для належного зберігання обчислень змін тексту в індексі Lucene.

Для підтримки тимчасових діапазонів дійсності для кожної версії документа ми використали тип поля Solr для діапазону дат. Поле validity_range та його тип dateRange були додані до XML-схеми Solr, як показано у лістингу 3.1.

Лістинг 3.1. Типи полів XML-схеми Solr

```
<field name="validity_range" type="dateRange" />
<fieldType name="dateRange" class="solr.DateRangeField" />
```

Для обчислення зміни тексту між версіями були створені три спеціалізовані поля: deleted_term (видалення), added_term (додавання) та semi_del_term (напіввидалення). Кожне з цих полів представляє набір термінів; отже, прямі фразові запити до цих полів не мають сенсу. Варто зазначити, що, хоча прямі фразові запити вимкнені, можна створити запит Lucene, що підтримує фразовий пошук за змінами термінів, комбінуючи ці термінові поля з полем вмісту, яке підтримує фразові запити.Ми додали ці три поля до XML-схеми Solr (лістинг 3.2).

Лістинг 3.2. Поля XML-схеми Solr

```
<field name="deleted_term" type="text_general" indexed="true" termOffsets="false" s
<field name="added_term" type="text_general" indexed="true" termOffsets="false" sto
<field name="semi_del_term" type="text_general" indexed="true" termOffsets="false"
```

3.1.3 Обчислення тимчасових діапазонів дійсності за допомогою Lucene

Обчислення тимчасових діапазонів дійсності було реалізовано на Java. Після завершення індексації всієї колекції, можна обчислити діапазони дійсності для документів із непорожніми заголовками та вмістом.

Для ідентифікації дійсних документів та впорядкування їхніх версій необхідна ітерація по всьому індексу Lucene. Нормалізований URL кожного документа використовувався як ключ хеш-мапи. Тимчасове впорядкування версій для кожного документа досягалося за допомогою вставки в дерево (Tree Insertion), що показано в лістингу 3.3.

Лістинг 3.3. TemporalDocument.java: Упорядкування документів за датою

```
public class TemporalDocument implements
    Comparable<TemporalDocument> {

    private long wayback_date;

    public int compareTo(TemporalDocument that) {

        if (this.wayback_date > that.wayback_date) return 1;

        else if (this.wayback_date < that.wayback_date) return -1;

        else return this.docnum - that.docnum;
    }
}
```

Після впорядкування всіх версій виконується прохід по хеш-мапі для зв'язування кожної версії з її наступником. Хоча дерево ефективне для впорядкування, для обчислення діапазонів дійсності потрібен зв'язаний список.

Другий прохід по хеш-мапі обчислює діапазони дійсності, використовуючи Java Calendar та часові мітки, а також порядковий номер версії документа (лістинг 3.4).

Лістинг 3.4. TemporalDocument.java. Обчислення діапазонів дійсності

```
public String getDateRange() {
    String nextDateStr;

    if (next == null) {

        Calendar c = Calendar.getInstance();
        c.setTime(this.getWaybackDate());
        c.add(Calendar.DATE, 30);

        nextDateStr = getSolrDateString( c.getTime() );
    }

    else {
        //inclusive
        nextDateStr = next.getWaybackDateStr();
    }
    // ... [code truncated]
}
```

Результатом цього процесу є діапазони дійсності у форматі JSON. Користувачі системи можуть імпортувати цей JSON в індекс через панель Solr. Для кращої інтеграції з можливостями публікації панелі Solr було реалізовано автоматичне сегментування вихідного JSON на частини по тисячі записів.

Продуктивність. Для тестового набору даних, що включав 100 000 WARC-файлів (близько 10 000 дійсних документів), алгоритм обчислення діапазонів дійсності зайняв 1,98 секунди. Для 200 000 WARC-файлів (близько 20 000 дійсних документів) час виконання склав 3,95 секунди на машині з Windows.

Очищення Даних. Після початкового індексування індексатором UKWA WARC спостерігалися пошкоджені символи Unicode та HTML-сутності. Символи, що мали бути розділовими знаками, були перетворені на їхні ASCII-еквіваленти перед обчисленням діапазонів дійсності. Символи Unicode виправлялися за допомогою Java, а HTML-сутності замінювалися вбудованими функціями PHP HTML. Потім текст повторно публікувався в індекс Lucene. Крім того, індексатор WARC UKWA не видаляє стандартні

порти при нормалізації URL. Ми видалили стандартні порти за допомогою вбудованої бібліотеки java.net.URI в Java перед повторною публікацією та обчисленням діапазонів дійсності.

3.1.4. Обчислення змін тексту за допомогою Lucene

Обчислення тексту змін кожного документа відносно його наступної версії відбувається одночасно з обчисленням тимчасових версій.

1. Початкове заповнення. Кожен документ заповнюється його набором термінів та їхньою кількістю шляхом токенізації поля вмісту за допомогою стандартного аналізатора Lucene.

2. Обчислення додавань/видалень. Для обчислення додавань та видалень використовується операція різниці множин (Set Difference Operation) (лістинг 3.5). Цей Java-метод обчислює множини видалених (deletions) і доданих (additions) термінів, використовуючи операції різниці множин (HashSet.removeAll()).

Лістинг 3.5. TemporalDocument.java. Обчислення видалених та доданих термінів

```
public String getDeletedTerms() {

    HashSet<String> value = next.terms;
    HashSet<String> valueP = this.terms;

    //deletions
    HashSet<String> deletedValues = new HashSet<String>(valueP);
    deletedValues.removeAll(value);

    //additions
    HashSet<String> atermset = new HashSet<String>(value);
    atermset.removeAll(valueP);
```

На практиці видалені терміни призначаються версії, що передує видаленню терміна (рис. 3.2). Термін "yellow" було видалено з цього документа. Термін "yellow" призначається полю для видалених термінів (deleted terms field) для версії, яка безпосередньо передує видаленню.

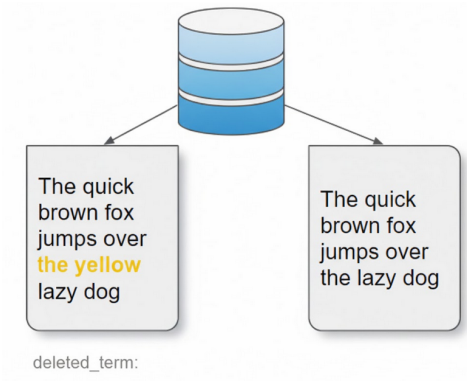


Рис. 3.2. Призначення поля для видалених термінів

Це ілюструє ключовий принцип індексації змін у тексті:

а) Ідентифікація Зміни. Система виявляє, що певний термін (наприклад, "yellow") присутній у версії V_n і відсутній у версії V_{n+1}.

б) Призначення (Прив'язка). Для забезпечення пошуку змін, факт видалення прив'язується до останньої версії, де цей термін був присутній (V_n). Таким чином, запит на видалений термін поверне версію, яка є найбільш релевантною для користувача (версію, з якої термін вилучений).

3. Обчислення напіввидалень. Використовується кількість термінів для обчислення напіввидалень (semi-deletions) (лістинг 3.6). Усі терміни вважаються додаваннями у першій версії документа.

Лістинг 3.6. TemporalDocument.java. Обчислення напіввидалених термінів

```
public String getSemiDeletedTerms() {

    HashMap<String, Integer> map = next.termCounts;
    HashMap<String, Integer> mapP = this.termCounts;

    HashSet<String> deletedValues = new HashSet<String>();

    for (Map.Entry<String, Integer> entry : mapP.entrySet()) {

        String key = entry.getKey();
        int value = entry.getValue();

        int valueN = (map.containsKey(key)) ? map.get(key) : 0;

        if (value > valueN && valueN > 0) {
            deletedValues.add(key);
        }
    }
```

4. Ранжування на основі кількісних показників. Для реалізації ранжування, де сторінка з більшою кількістю екземплярів видаленого терміну ранжується вище, ніж сторінка з меншою кількістю, ми опублікували кількість термінів в індекс. Це було досягнуто шляхом публікації потоку термінів у поле deleted_terms, де кожен термін повторювався стільки разів, скільки він був видалений.

3.1.5. Запити змін тексту за допомогою Lucene

Бекенд підтримує прямі та непрямі запити для пошуку змін.

Таблиця 3.1.

Опис запитів

Тип Запиту	**Запит Lucene**	**Опис**
Видалений термін	deleted_term:TERM	Повертає версію сторінки, що передує повному видаленню терміна.
Видалена фраза	text:"PHRASE TERMS" deleted_term:PHRASE deleted_term:TERMS	Шукає текст, що містить фразу, та обидва терміни, що були повністю видалені (неявний булевий оператор "I").
Напіввидалений термін	semi_del_term:TERM	Повертає сторінки, де деякі, але не всі екземпляри терміна були видалені.
Напіввидалена фраза	semi_del_term:(PHRASE OR TERMS) text:"PHRASE TERMS"	Повертає надмножину можливих відповідностей. Вимагає, щоб фраза була в попередній версії та принаймні один із термінів класифікувався як напіввидалений.

Запит на напіввидалену фразу забезпечує надмножину, яка включає всі можливі відповідності. Для того, щоб бути напіввидаленою фразою, фраза повинна з'являтися $m>1$ раз у попередній версії та $0<n<m$ разів у наступній версії. Поточний запит гарантує наявність фрази в попередній версії та

напіввидалення принаймні одного терміна, але не гарантує, що сама фраза зберігалася принаймні один раз у наступній версії.

3.1.6. Ранжування результатів пошуку змін тексту

Система використовує стандартний бал Lucene для ранжування, за винятком випадків, коли весь вміст на сторінці був видалений.

Розглянемо процес виключення несправжніх видалень. Ручний перегляд результатів показав, що сторінки, на яких було видалено весь вміст, часто є "м'якими 404" (soft 404) або помилками індексування динамічно завантаженого вмісту. Ці випадки не представляють справжніх значущих видалень. Виявлено, що помилки динамічно завантаженого вмісту (де мементо пошкоджено, оскільки динамічний вміст не архівується) траплялися значно частіше, ніж м'які 404 (рис. 3.3).

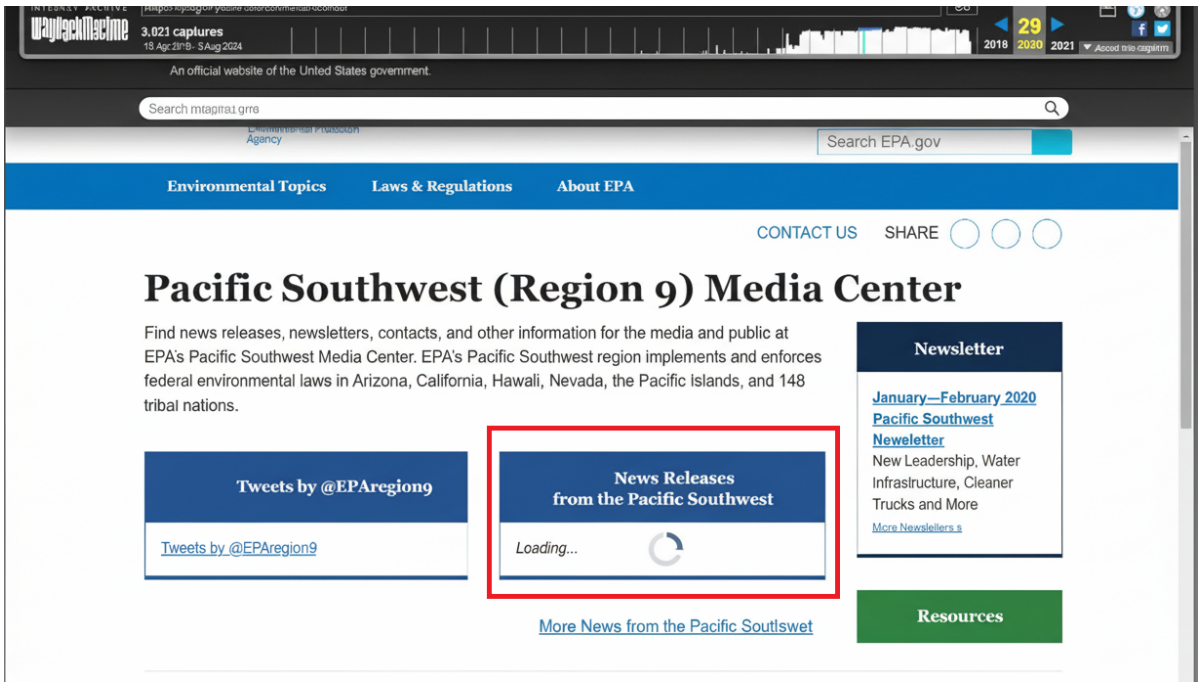


Рис. 3.3. Динамічно завантажений контент як хибнопозитивне видалення в індексі змін тексту

Представлена сторінка на рис. 3.3 перейшла від статично завантаженого контенту у 2016 році до динамічно завантаженого контенту до

2020 року. Обсяг сторінки, ймовірно, зменшився більш ніж удвічі, хоча сам контент не був видалений.

Цей приклад ілюструє поширену проблему при аналізі веб-архівів:

Хибнопозитивне видалення (False Positive Deletion) - алгоритм індексації помилково класифікує значне зменшення розміру сторінки як масове видалення тексту.

Причина: Зміна в методі доставки контенту (перехід зі статичного HTML на динамічне завантаження через JavaScript/AJAX).

Наслідок: Якщо веб-архіватор (краулер) не зміг належним чином захопити або відтворити динамічно завантажений контент у новій версії (2020), індекс фіксує лише статично доступну частину, що робить нову версію значно меншою і створює ілюзію видалення контенту, хоча насправді він просто не був архівований.

Оскільки несправжні видалення не відповідають справжнім змінам, було прийнято рішення ранжувати ці елементи останнімиВідповідно, ми автоматично виявляємо м'які 404, коли стара версія вебсторінки принаймні в десять разів більша за нову версію. Цей розрахунок було інтегровано в Java-бекенд (лістинг 3.7) для взаємодії з PHP-фронтендом.

Лістинг 3.7. Обчислення Soft-404

```
s404log = Math.log10(this.content.length() * 1.0 /
    next.content.length());
```

Цей вираз обчислює десятковий логарифм відношення довжини контенту поточної версії (this) до довжини контенту наступної версії (next).

3.2. Реалізація користувацького інтерфейсу для взаємодії зі змінами тексту

Користувацька взаємодія з індексом термінів зміни тексту реалізована через спеціалізований пошуковий інтерфейс, створений із застосуванням

Solarium — PHP-інтерфейсу для Solr. Цей інтерфейс забезпечує можливість формування запитів щодо видаленого терміна, видаленої фрази, доданого терміна або доданої фрази. Сторінка формування запиту представлена на рисунку 3.4.

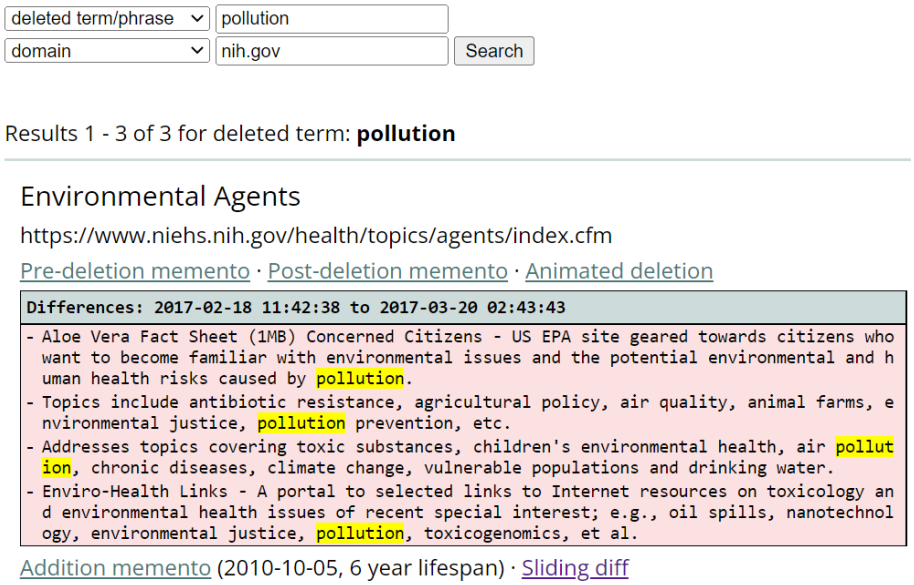


Рис. 3.4. Випадаючі меню на сторінці результатів пошукової системи для змін у тексті

Цей елемент інтерфейсу (випадаюче меню) використовується для фільтрації або уточнення пошуку, а також для індикації ключових часових точок, де відбулися відповідні зміни:

- Додавання терміна - позначає першу версію, де запитуваний термін з'явився.

- Видалення терміна - позначає версію, що передує моменту, коли термін зник.

Таке чітке індикування допомагає користувачеві швидко локалізувати часові рамки значущих змін на вебсторінці.

На сторінці результатів пошукової системи (рис. 3.5) для змін у тексті версії сторінки які відповідають додаванню та видаленню запитуваного терміна "pollution" (забруднення), чітко індиковані.

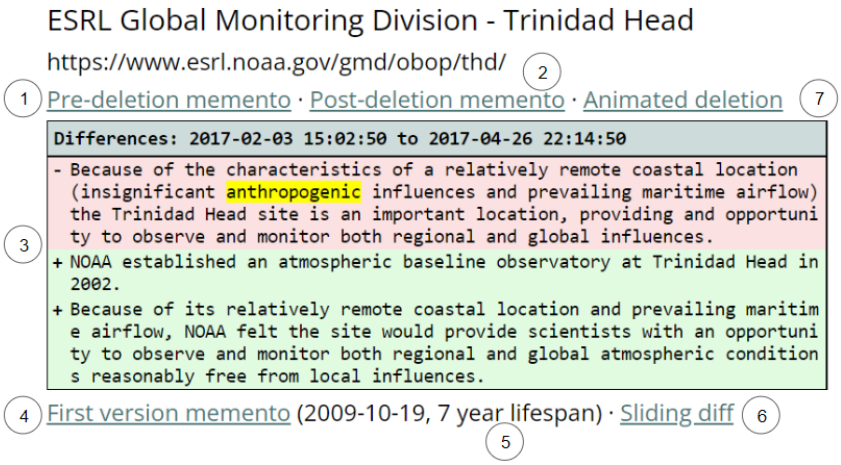


Рис. 3.5. Інтерфейс пошуку змін у тексті

Цей рисунок демонструє ключові компоненти результату пошуку змін у тексті:

- 1, 2, та 4 - індивідуальні посилання для відтворення відповідних мементо сторінки.

- 3 - відображення різниці (diff) між версіями, що передують і слідують за видаленням.

- 5 - обчислення терміну існування контенту (content lifespan).

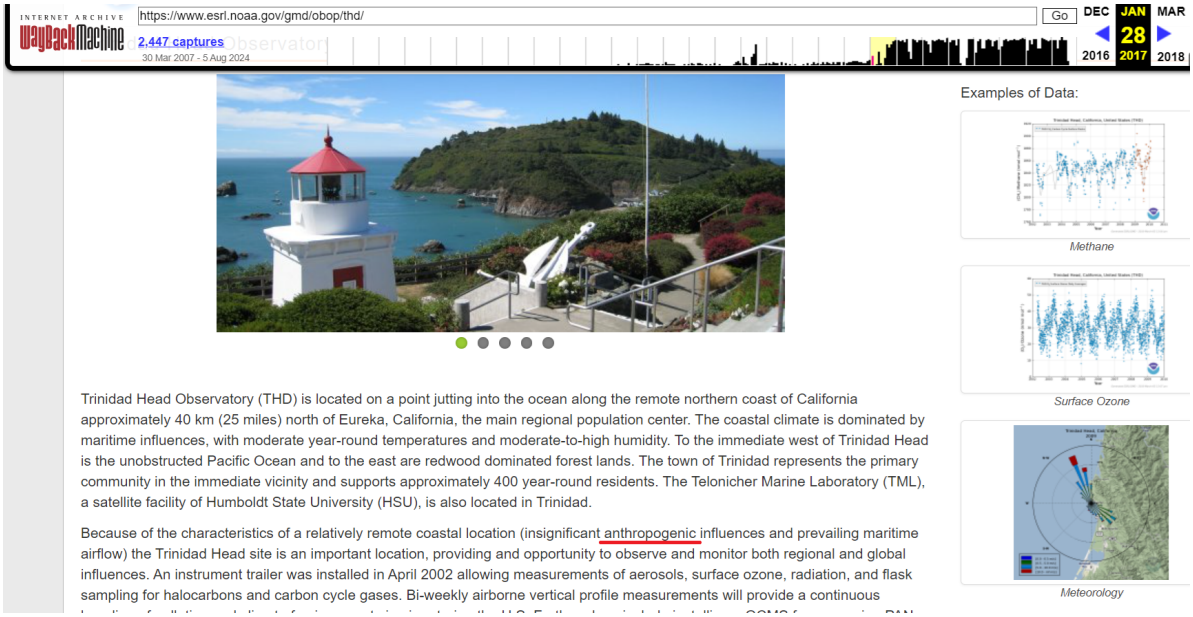
- 6 - посилання на переглядач ковзних відмінностей (sliding diff viewer) для всіх проіндексованих версій сторінки.

- 7 - Посилання на анімацію видалення (deletion animation).

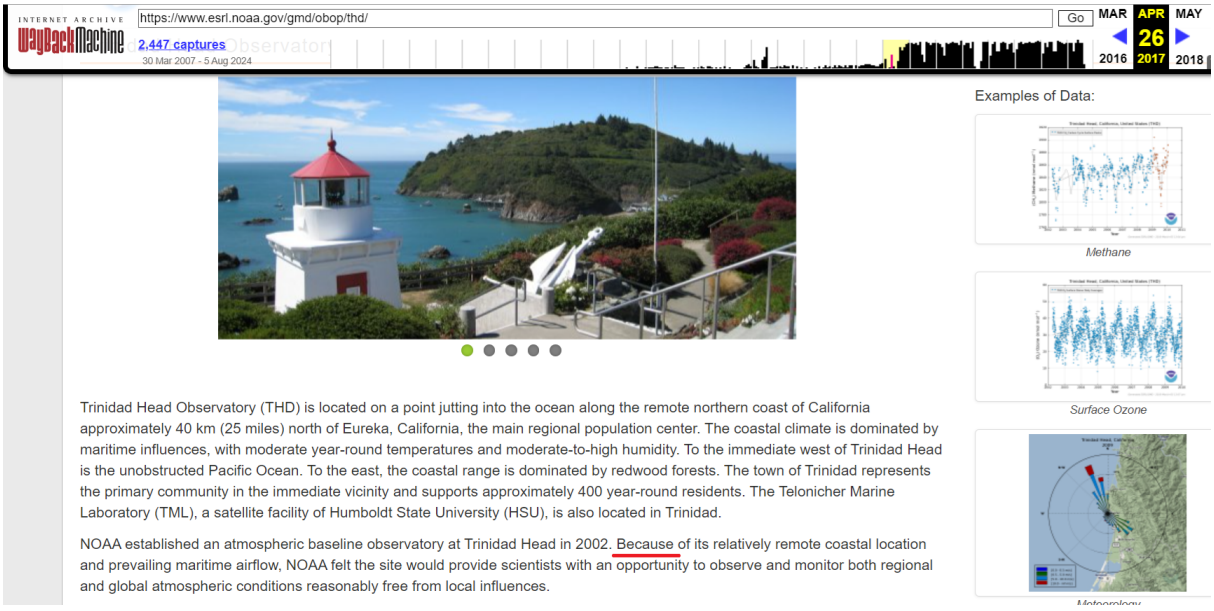
Цей інтерфейс агрегує часові дані, надаючи користувачеві комплексний огляд значущої зміни:

Посилання (1, 2, 4) забезпечують прямий доступ до архівних знімків для контекстуального перегляду. Diff (3) слугує основним індикатором того, що саме було змінено. Термін існування (5) дає змогу оцінити, як довго змінений контент був присутній на сторінці. Ковзний Diff (6) дозволяє детально проаналізувати всі проміжні зміни між двома основними точками. Анімація (7) пропонує динамічний та інтуїтивно зрозумілий спосіб візуалізації процесу видалення.

Результати пошуку відображаються на сторінці результатів пошукової системи, проілюстрованій на рис. 3.5. Інтерфейс SERP містить посилання на мементо до та після видалення, включаючи відповідні дати і час, а також надає фрагмент ковзної різниці, анімовані елементи видалення та відмінності, що демонструють зміни (рис. 3.6).



а)



б)

Рис. 3.6. Приклад видалення терміна "anthropogenic" зі сторінки

На рис. 3.6 а термін "anthropogenic" присутній у версії сторінки за січень 2017 року. Звертається увага, що інша архівна копія цієї сторінки в веб-архіві залишається недоступною станом на зараз. На рис. 3.6 б термін відсутній у версії сторінки, датованій 26 квітня 2017 року, що відповідає результату пошуку, показаному на рисунку 3.5.

Цей рисунок слугує візуальною демонстрацією результату пошуку змін у тексті:

- версія до видалення а) підтверджує існування запитуваного терміна (січень 2017).

- версія після видалення б) підтверджує відсутність терміна (квітень 2017).

Контраст між цими двома версіями верифікує факт видалення та визначає часовий діапазон, в межах якого відбулася зміна.

3.2.1. Сторінка запиту та результатів пошукової системи змін тексту

Розглянемо сторінку запиту. Користувачі можуть ініціювати пошук видалення окремого терміна за полем. Інтерфейс, реалізований за допомогою PHP Solarium, надає випадаюче меню (рис. 3.4) для визначення наміру пошуку у полі видалень, а також текстове поле для введення видалених термінів.

Для уможливлення підрахунку входжень фраз із довільною кількістю термінів, функціональність, яка безпосередньо не підтримується Lucene, ми застосували реалізацію UAX на PHP.

Система відображає у результатах пошуку кожну сторінку, де виявлено принаймні одне входження заданої фрази, але менше, ніж кількість входжень у попередній версії документа. Код, що відповідає за підрахунок фраз, представлений у лістингу 3.7, а код для фільтрації отриманих результатів наведено у лістингу 3.8.

Лістинг 3.7. temporal_document.php. Підрахунок термінів

```
public function content_joined_with_spaces() {
    return $this->text_joined_with_spaces($this->content);
}

//implementation of UAX29
public function text_joined_with_spaces($phrase) {
    //...
}

public function count_phrase_freq($phrase) {
    $text_joined_with_spaces = $this->content_joined_with_spaces();
    return substr_count($text_joined_with_spaces, $phrase);
}

// ...

public function compare_phrase_freq($doc2, $phrase) {
    return $this->count_phrase_freq($phrase) -
        $doc2->count_phrase_freq($phrase);
}
```

Цей PHP-код демонструє функції для підрахунку частоти фраз, включаючи використання імплементації UAX29 для токенізації. Функція compare_phrase_freq обчислює різницю частот фрази між двома документами.

Лістинг 3.8. semi_del_results.php: Фільтрація результатів запиту для фраз

```
if (!$is_deleted_phrase || ($is_deleted_phrase &&
    $document->compare_phrase_freq($document2, $deleted_term) > 0)) {
}
```

Цей PHP-фрагмент демонструє логіку фільтрації результатів пошуку для фраз із напіввидаленням. Умова if перевіряє, чи не є фраза видаленою (!$is_deleted_phrase), АБО, якщо вона була видалена, чи різниця частот фрази між поточною версією та наступною версією більша за нуль (> 0):

Комбінація обраного поля та терміна користувача ("TERM") трансформується у відповідний запит Lucene. Додатково користувачі мають

можливість фільтрувати результати за доменом верхнього рівня (наприклад, TLD.COM), що додає до запиту сегмент домен:TLD.COM (рис. 3.7).

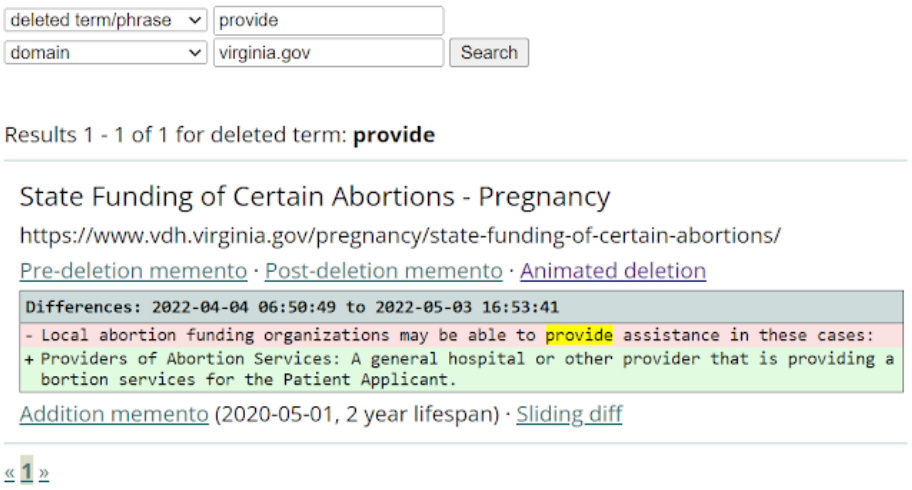


Рис. 3.7. Можливість фільтрації результатів за доменом верхнього рівня

Ця функція забезпечує пошукову фасету, дозволяючи користувачам обмежувати результати змін у тексті лише тими вебсторінками, які належать до певного домену верхнього рівня. Це підвищує точність і релевантність пошуку, фокусуючи його на конкретній архівній колекції або організації.

Розглянемо сторінку результатів (SERP). У відображенні результатів, забезпеченому реалізацією Solarium, множинні версії однієї сторінки агрегуються для формування єдиного значущого результату пошуку. Версія, що передує видаленню, наступна версія, а також версія, яка містить додавання терміна, відображаються як єдиний запис, супроводжуючись посиланнями для відтворення цих версій через PyWB.

Система обчислює термін існування контенту (content existence term) — різницю між часовими мітками версії з додаванням терміна та версії після його видалення, і відображає цю інформацію. Фрагмент тексту, представлений у SERP, є "diff" між індексованим текстом версій до та після видалення, який фільтрується для відображення лише рядків, що містять пошуковий термін, із застосуванням виділення цього терміна.

3.2.2. Підтримка складних запитів

Розглянемо запит видаленої фрази. Оскільки запит на видалену фразу повертає версію сторінки, що безпосередньо передує видаленню, наступна версія з видаленням повинна бути включена до результату. Це досягається за допомогою діапазону дійсності (validity_range), який використовується для знаходження наступної версії (лістинг 3.9). Імплементація діапазонів дійсності з включеними границями створює перекриття в одну секунду між послідовними версіями в індексі.

Лістинг 3.9. deletion_results.php. Пошук версії після видалення

```
$query2 = $client->createSelect();
$query2->setQuery('url_norm:'.$document->url_norm);
$query2->createFilterQuery('crawltime')->setQuery(
    'validity_range:['.$document->get_next_wayback_date().' TO '
    .$document->get_next_wayback_date().']');
$query2->setDocumentClass('TemporalDoc');

// ... [code truncated]

$query2->addSort('id', $query::SORT_DESC);
$resultset2 = $client->select($query2);
$document2 = $resultset2->getIterator()->current();
```

Цей PHP-код використовує Solarium для пошуку наступної версії документа (після видалення), використовуючи поле validity_range.

Розглянемо етап виділення фрази. Для коректного відображення виділеної фрази необхідне додаткове кодування. Хоча diff обчислюється за допомогою бібліотеки PHP diff, Solarium не може виділити його, оскільки diff не є полем Lucene. Проблема ускладнюється тим, що пошукові фрази можуть бути розділені токенами у результуючому тексті (наприклад, "цілорічний" у запиті може бути "ціло-річний" у документі). Для обходу цієї проблеми кожен пошуковий термін виділяється індивідуально (лістинг 3.10).

Цей PHP-код спочатку обчислює різницю (diff) між двома документами. Потім він використовує регулярний вираз (preg_replace) для

виділення знайденого видаленого терміна ($deleted_term) у результаті різниці ($diff_out) за допомогою HTML-тегів із жовтим фоном (#FFFF00).

Лістинг 3.10. deletion_results.php. Виділення пошукових термінів у різниці

```
$diff_out = $document->diff($document2, $deleted_term);

$diff_out = preg_replace('/\b(' . $deleted_term . ')\b/i',
'<span style="background-color: #FFFF00">$1</span>', $diff_out);
```

Для видаленої фрази термін існування контенту обчислюється шляхом запиту останньої версії сторінки, де будь-який із термінів позначено як додавання. Дата цього результату використовується як нижня межа для першого входження фрази.

Після виконання запиту на напіввидалену фразу, система перевіряє кожен потенційний збіг, щоб підтвердити, що це дійсно приклад напіввидаленої фрази. Для токенізації текстового поля та підрахунку входжень фраз з довільною кількістю термінів використовується реалізація UAX #29 на PHP, оскільки ця функціональність недоступна безпосередньо через Lucene. Система відображає як результат кожну сторінку, яка має принаймні одне, але менше входжень фрази, ніж у попередній версії.

3.2.3. Перегляд відмінностей

Кожен результат SERP є агрегацією версій, що, може приховувати динаміку змін. Для розгрупування версій впроваджено інструмент ковзної різниці, який дозволяє користувачеві порівнювати кожну версію з наступною.

Цей інструмент доступний через посилання в кожному результаті і використовує показ різниці у форматі "пліч-о-пліч" (рис. 3.8). Користувачі можуть навігувати між наборами відмінностей за допомогою повзунка, що є ефективно для порівняння змін у вихідному коді з часом.

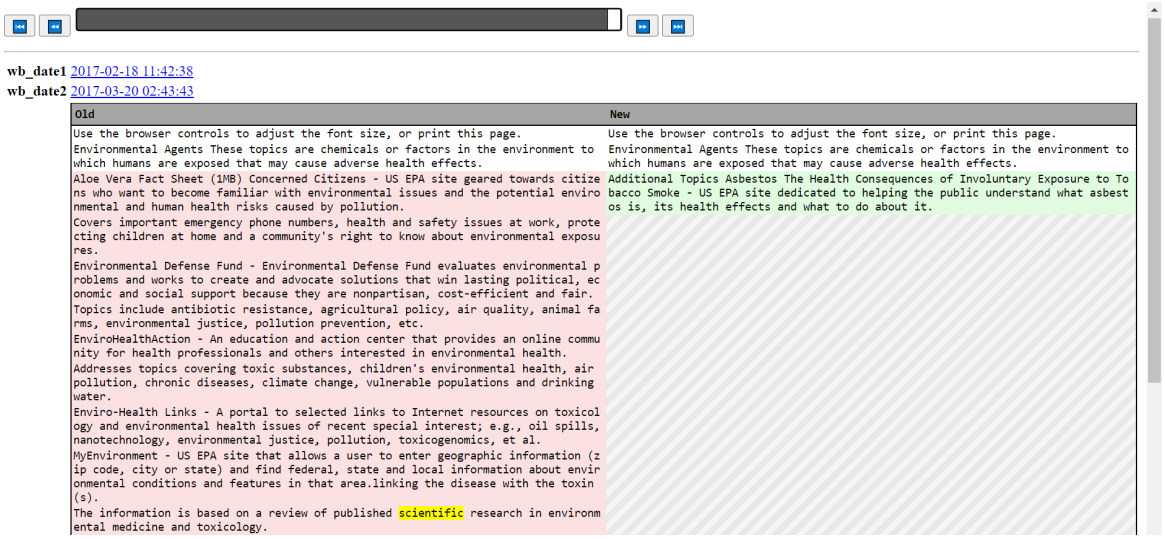


Рис. 3.8. Переглядач відмінностей у версіях

Переглядач ковзних відмінностей (рис. 3.8) показує, що термін 'scientific' було видалено зі сторінки у 2017 році, разом із контекстом цього видалення. Цей переглядач (sliding difference viewer) є інструментом візуалізації, який дозволяє користувачеві чітко визначити, який саме термін був видалений і показати оточення тексту, звідки було вилучено термін, що є критично важливим для розуміння значущості цієї редакції в архіві.

Інструмент реалізовано з використанням Solarium, бібліотеки PHP diff та додаткового JavaScript. Він використовує індексований текстовий вміст із Lucene. Дата додавання та видалення передається через параметри URL (лістинг 3.11).

Лістинг 3.11. deletion_results.php. Посилання із вхідними даними для повзунка різниці

```
<a href="diff-slider.php?page=' . urlencode($document->url_norm) .
'&wbdate1=' . $slid_diff_page->wayback_date . '&wbdate2=' .
$document2->wayback_date . '&dterm=' . $deleted_term . '">
Sliding diff</a>
```

Цей PHP-код генерує посилання на переглядач ковзної різниці (diff-slider.php), передаючи параметри URL, що включають нормалізований URL сторінки, дату першого та другого мементо, а також видалений термін.

За допомогою кнопок швидкого перемотування користувач може пропускати ідентичні мементо, оскільки JavaScript зберігає звичайний текст кожної сторінки для швидкого порівняння послідовних версій (лістинг 3.12).

Лістинг 3.12. slider.js. Згортання в JavaScript

```
function navigateDiff(nav_type) {
    //...
    else if (nav_type == NAV_COAL_FORW) {

        var idx = rangeInputDom.value;
        idx = idx + 1;

        while (idx < wb_arr.length - 1 &&
            diff_arr[idx - 1].diff == 'Page versions are identical') {
            idx = idx + 1;
        }

        rangeInputDom.value = idx;
    }
}
```

Ця JavaScript-функція демонструє логіку "згортання" (coalescing) ідентичних версій у переглядачі різниць, дозволяючи користувачеві швидко перемотувати вперед, пропускаючи сторінки з ідентичним вмістом.

3.2.4. Переглядач анімованих видалень

Результати пошуку також містять посилання на одночасне відтворення версій до та після видалення (рис. 3.9). Цей інструмент подвійного відтворення демонструє анімацію різниці в контексті.

Цей рисунок 3.9 ілюструє роботу переглядача анімованих видалень (Animated Deletions Viewer) — інструменту, який використовується для динамічної візуалізації змін у веб-архівах. Його ключові функції:

- Фокус на зміні - чітко показує факт видалення конкретної фрази ("endangered species").

- Динамічне представлення - надає анімоване відтворення версій до та після зміни, що допомагає користувачеві сприймати зміни в контексті та в ілюзії реального часу.

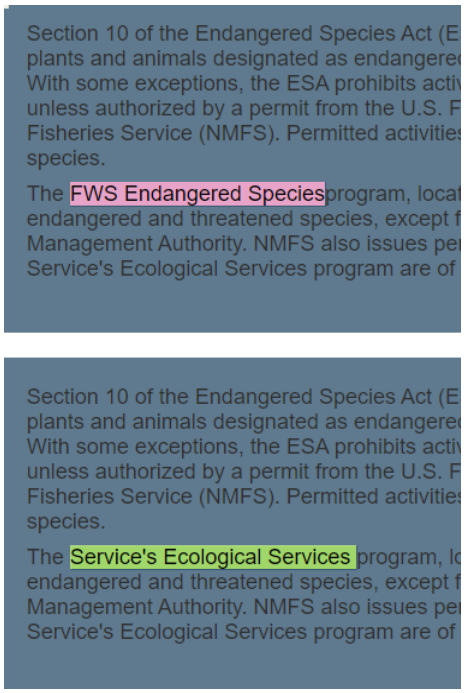


Рис. 3.9. Анімація демонструє видалення фрази "endangered species" зі сторінки

Для посилення перцептивного сприйняття змін використовуються відтінки та кольори виділеного тексту. На відміну від статичних інструментів, анімація послідовно переходить до кожної зміни, створюючи ілюзію того, що зміни відбуваються в реальному часі. Алгоритм перцептивного сприйняття деталізовано в лістингу 3.12.

Лістинг 3.12. web_diff.py. Анімація змін по одній за раз

```
function printLetterByLetter(index, speed){

var anchor = '<a class="wm-diff-anchor" id="wm-diff-del' +
index + '"></a>';

window.location = window.location.origin + window.location.pathname +
window.location.search + '#wm-diff-del' + index

var interval = setInterval(function(){
//...

document.getElementById(destination).innerHTML = anchor +
destText.substring(0, destText.length - j);

//...

sleepFor(400);
}
```

Ця функція JavaScript printLetterByLetter створює якір (anchor) та оновлює місцеположення вікна (window.location) для переходу до нього. Вона використовує setInterval для анімації видалення, поступово обрізаючи текст (destText) і вставляючи якір, з паузою в 400 мілісекунд між кроками (sleepFor(400)).

Для створення анімації використано бібліотеку Python HTML difference від EDGI. Код було розширено для генерації HTML та JavaScript для анімації об'єднаних сторінок.

Бібліотека EDGI обчислює всі зміни. Для забезпечення релевантності анімації, видалення та додавання, не пов'язані з терміном запиту, були видалені перед виконанням анімації (лістинги 3.13 та 3.14).

Лістинг 3.13. web_diff.py. Видалення видалень, які не відповідають запитуваному терміну або фразі

```
if ' ' not in dterm:
    #//...
else:
    #phrase search
    #//UAX29 implementation
    if " " + dterm + " " not in text_joined_with_spaces:
        deletion.decompose()
    else:
        deletion['id'] = 'wm-diff-del-wrapper' + str(i)
        deletion.a['id'] = 'wm-diff-del' + str(i)
        deletion.a.string = ''
        i = i + 1
```

Цей фрагмент коду Python (лістинг 3.13) обробляє видалення, особливо у випадку пошуку фрази, використовуючи імплементацію UAX29 для перевірки наявності фрази. Якщо фраза не знайдена, видалення deletion.decompose().

Цей фрагмент коду Python (лістинг 3.14) виконує два основні завдання:

- Видаляє <del>-теги та обгортає <ins>-теги, що не відповідають запитуваному терміну/фразі.

- Перенумеровує якірні ID (wm-diff-del) для коректної роботи анімації, забезпечуючи послідовність після видалення нерелевантних змін.

Лістинг 3.14. web_diff.py. Видалення додавань, які не відповідають запитуваному терміну або фразі

```
comparison['combined'] = re.sub(r'<del class="wm-diff"
id="wm-diff-del[0-9]+".*?</del><ins(.*?)>(.+?)</ins>',
r'<del class="wm-diff" id="wm-diff-del-wrapper\g<1>\g<2></del>
<INS ID="wm-diff-ins-wrapper\g<1>"\g<3></INS>', comparison['combined'])

comparison['combined'] = comparison['combined'].replace(
r'<ins class="wm-diff">', '')


comparison['combined'] = comparison['combined'].replace('</ins>', '')

diffids = re.findall(r'wm-diff-[\w\-a-z]+[0-9]+', comparison['combined'])

lastid = 0

for i in range(len(diffids)):

    id = int(re.sub(r'[^0-9]', '', diffids[i]))
    notid = diffids[i].replace(str(id), '')

    if id < lastid:
        comparison['combined'] = comparison['combined'].replace(
            diffids[i], notid + str(lastid))
    elif id > lastid:
        lastid = id
```

Для коректної роботи анімації знадобилися додаткові модифікації: упорядкування якірних посилань (anchor links), а також заміна відносних посилань на зображення на повні адреси, оскільки відтворення відбувається поза середовищем.

Отже, сторінка результатів пошукової системи змін тексту демонструє розширену функціональність у порівнянні з іншими тимчасовими пошуковими системами (наприклад, SolrWayback) та стратегіями моніторингу веб-сайтів (наприклад, EDGI).

Система групує лише ті версії сторінки, які містять значущі зміни пошукового терміна, на відміну від SolrWayback, який може відображати

ідентичні версії та не включати версію, що йде безпосередньо після видалення.

Фрагмент різниці дозволяє вивчати зміни в контексті, а додаткові інструменти ковзної різниці та анімації забезпечують додаткову деталізацію та динаміку змін. Вивчення змін у контексті неможливе при використанні лише стратегії моніторингу веб-сайтів.

3.3. Огляд функціональності навігаційної панелі веб-архівів та пропозиції щодо її вдосконалення

Наше дослідження виявило, що навігаційна панель у системі відтворення Wayback Machine (Internet Archive) є функціональною, але не реалізує свій максимальний потенціал корисності для користувачів. На основі аналізу журналів використання та виявлених потреб користувачів ми пропонуємо низку змін до функціональності цієї панелі. Пропоновані вдосконалення включають: оптимізацію посилань стрілок навігації на значущі мементо зі змінами, інтеграцію піктограми для інструменту змін (diff tool), механізми запобігання помилкам, зокрема попередження про перенаправлення, а також нову цільову сторінку для перенаправлень, що забезпечує вищу автономію користувача.

Користувачі, які прагнуть проаналізувати часові зміни веб-сторінок, зазвичай є досвідченими користувачами веб-архівів. Вони традиційно використовують TimeMap, часову шкалу або навігаційні кнопки панелі відтворення для ручного визначення змін.

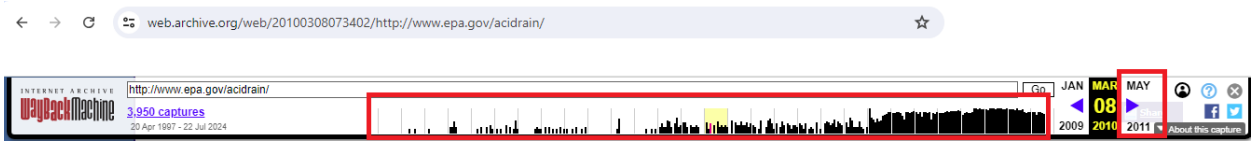


Рис. 3.10. Поточні кнопки навігаційної панелі Wayback Machine

Поточна навігаційна панель Wayback Machine, проілюстрована на рис. 3.10, має дві ключові зони: часову шкалу (лівий прямокутник) та навігаційні кнопки (правий прямокутник). Фактично, правий блок містить три посилання на найближчі збережені знімки за днем, місяцем та роком.

Основна проблема полягає в тому, що навігаційні кнопки посилаються на версії, які можуть бути ідентичними або пропускати версії зі змінами, що не відповідає потребам користувачів, які прагнуть переглядати послідовні зміни. Аналіз журналів підтверджує, що користувачі бажають бачити наступну версію веб-сторінки, де відбулися зміни. Кнопки мають посилатися на попередню та наступну версії сторінки зі змінами, а не на попередній та наступний знімки, і не повинні пропускати жодної значущої версії.

Крім того, інструмент змін (diff tool) доступний лише через TimeMap, а не безпосередньо з системи відтворення. Пряме посилання на панелі відтворення до інструменту змін є необхідним для того, щоб користувачі могли бачити виділені зміни (у стилі diff) без необхідності ручного порівняння двох мементо.

Рис. 3.11. Пропоновані зміни в навігаційній панелі

На рисунку 3.11 представлено прототип панелі, що інтегрує запропоновану функціональність:

1. Навігація за змінами: кнопки тепер посилаються на попередню та наступну версії сторінки, що містять фактичні зміни.

2. Інтеграція інструменту змін: додано піктограму (верхній ряд, ліворуч) для швидкого доступу до інструменту змін, який відображає різницю між поточною та попередньою версіями сторінки.

3. Попередня версія: остання версія в попередньому діапазоні дійсності.

4. Наступна версія: перша версія в наступному діапазоні дійсності.

У подальшому прототипі заплановано додати випадаюче меню під датою мементо, яке міститиме список усіх мементо в поточному діапазоні дійсності.

Реалізація цієї сторінки вимагатиме додаткової інфраструктури.

1. Визначення подібності перенаправлення.

Визначення, чи має текст перенаправлення подібність до поточної сторінки, може бути виконане за допомогою існуючих алгоритмів текстової подібності.

2. Знаходження кращої відповідності.

Якщо перенаправлення нижче порогу подібності, необхідно знайти кращу відповідність за повним текстом, що наразі веб-архівами не підтримується.

3. Недоступні сторінки.

Якщо сторінка недоступна, завданням є рекомендація сторінки зі схожим вмістом.

Отже, було розроблено бекенд та фронтенд пошукової системи зміни тексту, а також інструменти візуалізації: анімацію змін та інструмент ковзної різниці (sliding diff tool). Також запропоновано прототип навігаційної панелі, яка орієнтується на зміни між версіями сторінки, а не лише на часові інтервали.

Висновки до розділу

У третьому розділі розроблено архітектуру та методологію системи пошуку й аналізу змін у веб-сторінках. Запропонована модель базується на використанні Apache Solr для обробки версійних колекцій документів та механізмів Lucene для визначення часових діапазонів і текстових

відмінностей. Реалізовано алгоритми пошуку, ранжування та візуалізації змін, які забезпечують більш точне виявлення та представлення еволюції контенту. Створено концепцію користувацького інтерфейсу, що підтримує інтерактивну роботу з архівними версіями сторінок, включно з переглядом різниць і анімованими видаленнями. Розроблені рішення доводять ефективність запропонованого підходу та можуть бути інтегровані у сучасні системи веб-архівації для підвищення зручності та аналітичних можливостей користувачів.

ВИСНОВКИ

У результаті проведеного магістерського дослідження, спрямованого на формування теоретичних і практичних засад створення моделей та методів сервісів архівування з відстеженням змін у веб-сторінках, було досягнуто низку вагомих наукових і прикладних результатів. Робота комплексно охоплює питання аналізу існуючих систем веб-архівації, обґрунтування когнітивних аспектів взаємодії користувача з веб-архівами, дослідження алгоритмів та архітектур індексації, а також розроблення підходів до побудови ефективної пошукової системи змін у веб-сторінках.

Перший розділ роботи було присвячено дослідженню теоретичних і методологічних аспектів проблеми виявлення та відстеження змін у веб-контенті. У процесі аналізу встановлено, що зростання обсягів динамічного веб-контенту, а також швидкість його оновлення створюють суттєві виклики для збереження інформаційної спадщини та забезпечення достовірності цифрових джерел.

Було показано, що існуючі пошукові інтерфейси веб-архівів мають значні обмеження щодо гнучкості пошуку, інтерактивної навігації та візуалізації змін. Досліджено когнітивні аспекти взаємодії користувачів із системами архівування: доведено, що ефективність сприйняття історичних змін суттєво залежить від когнітивного навантаження, архітектури інтерфейсу та способу представлення результатів пошуку.

Другий розділ роботи зосереджено на аналізі архітектурних рішень, алгоритмів індексації та наборів даних, що використовуються у процесі побудови систем відстеження змін у веб-контенті. Проведено порівняльний аналіз інструментів Apache Lucene, Solr та інших пошукових технологій, які формують ядро сучасних систем веб-архівації. Визначено, що саме використання інвертованих індексів і гібридних структур пошуку забезпечує високу продуктивність обробки історичних запитів.

Особливу увагу приділено дослідженню поведінки користувачів під час роботи з інструментами візуалізації змін. Було встановлено, що ефективні методи презентації відмінностей базуються на адаптивній візуалізації, що враховує контекст змін і тип контенту. Отже, у межах другого розділу було визначено ключові алгоритмічні компоненти системи архівування, здатної забезпечити аналітичне відстеження та гнучке представлення еволюції веб-ресурсів.

Третій розділ присвячено розробленню архітектури та методології системи відстеження змін у веб-сторінках. На основі аналітичних результатів попередніх розділів було спроєктовано архітектуру пошукової системи, що забезпечує обробку версійних колекцій документів із використанням платформи Apache Solr. Реалізовано механізми визначення часових діапазонів дійсності документів, обчислення відмінностей між версіями текстів за допомогою індексів Lucene, а також алгоритми ранжування результатів пошуку змін.

Розроблено концепцію користувацького інтерфейсу, який підтримує інтелектуальну взаємодію з архівними даними. Інтерфейс включає сторінку запиту з розширеними параметрами фільтрації, модуль порівняння версій, а також переглядач анімованих видалень, що дозволяє користувачу візуально спостерігати еволюцію контенту. Додатково представлено рекомендації щодо вдосконалення навігаційної панелі веб-архівів — шляхом інтеграції модулів часової візуалізації та контекстного аналізу.

Таким чином, третій розділ забезпечив практичну реалізацію теоретичних положень дослідження та підтвердив ефективність запропонованих методів виявлення змін, ранжування та візуалізації результатів пошуку у веб-архівах.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Apache Lucene internals. Okay, first things first. | by Vikas Sangle | Medium. - https://sanglevikas25.medium.com/apache-lucene-internals-8035dfae89ed
2. Apache Solr Fundamentals. Apache Solr is an information retrieval… | by Manish Sharma | Medium - https://medium.com/@mansha99/apache-solr-fundamentals-43e360962cc8
3. Frew, L., Nelson, M. L., Weigle, M. C. (2023). Making Changes in Webpages Discoverable: A Change-Text Search Interface for Web Archives. Proceedings of the ACM/IEEE Joint Conference on Digital Libraries.
4. Costa, M., Gomes, D., Silva, M. J., et al. (2016). The evolution of web archiving. Sobre / Arquivo.pt.
5. Davis, C. (2014). Archiving the Web: A Case Study from the University of Victoria Libraries. Journal of the Code4Lib
6. Cruz, D., Silva, M. J., Costa, M. (2012). Adapting search user interfaces to web archives. (Paper).
7. Weigle, M. C., et al. (n.d.). Visualizing Digital Collections of Web Archives. Columbia University Report.
8. Ainsworth, S. G., Nelson, M. L. (2013). Evaluating Sliding and Sticky Target Policies by Measuring Temporal Drift in Acyclic Walks Through a Web Archive. arXiv preprint.
9. ‘‘Web-Archiving’’ (DPC Technology Watch Report). Digital Preservation Coalition.
10. Van de Sompel, H., Nelson, M. L., et al. (2013). Memento: Time travel for the Web. International Journal of Digital Libraries.
11. Berberich, K., Weikum, G., et al. (2007). Temporal validity and coalescing in versioned document collections.

12. Sherratt, C., Jackson, M. (2022). GLAM Workbench change-detection tools for web archives.
13. Almeida, A. et al. Web Archival Data Analysis: A Survey of Methods and Applications. ACM Computing Surveys, vol. 50, no. 4, 2017.
14. Mann, T. G. & Mauldin, K. The Role of Headers in Archival Search. Proceedings of the 8th International Conference on Web Archiving (ICWA), 2013.
15. Nelson, M. L. et al. The Impact of Memento on Web Archiving and Preservation. International Journal on Digital Libraries, 2017.
16. Burrows, L. & Nelson, M. L. The Coalescing Proxy: Optimizing Web Archive Access. Proceedings of the 12th International Conference on Web Archiving (ICWA), 2017.
17. Nelson, M. L. et al. Measuring the Completeness of Web Archives. D-Lib Magazine, vol. 21, no. 3/4, 2015.
18. Schoch, T. Using Log Analysis to Discover User Needs and Usability Issues. Journal of Usability Studies, vol. 10, no. 3, pp. 91–105, 2015.
19. Wickett, L., & Van de Sompel, H. Memento: Time Travel for the Web. International Journal on Digital Libraries, vol. 12, no. 2-3, pp. 135–151, 2012.
20. Gomes, D., Miranda, C., & Ferreira, J. L. Web Archiving: A Survey. Journal of Universal Computer Science, vol. 17, no. 14, pp. 1957–1981, 2011.
21. Baeza-Yates, R., & Ribeiro-Neto, B. Modern Information Retrieval: The Concepts and Technology behind Search. Addison-Wesley, 2011.
22. Zhu, J. et al. Identifying New URL Patterns for Restructured Websites in Web Archives. Proceedings of the ACM Web Science Conference (WebSci), 2018.
23. Van de Sompel, H., & Nelson, M. L. The Memento Protocol. RFC 7089, The Internet Engineering Task Force (IETF), 2013.

24. Nelson, M. L., Van de Sompel, H., & Maimone, R. The Soft-404 Problem in Web Archiving. Proceedings of the Joint Conference on Digital Libraries (JCDL), 22-26 June 2009.
25. Denoue, L., et al. (2018). SlideDiff: Visualizing evolving documents with animations. Proceedings of the IEEE Conference on Information Visualization.
26. Chevalier, F., et al. (2010). Diffamation: Animated difference visualization for text revisions. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.
27. Internet Archive. (n.d.). Heritrix: Web crawler for web archiving. Retrieved from (https://github.com/internetarchive/heritrix3)
28. Internet Archive. (n.d.). Wayback Machine API documentation. Retrieved from (https://archive.org/help/wayback_api.php)
29. Pennock, M. (2013). Web archiving: Managing and preserving digital content. JISC Report.
30. Owens, T. (2014). What do you mean by archive? Genres of usage for digital preservers. The Signal Blog, Library of Congress.
31. Dougherty, M., et al. (2010). Citations and link rot: Decay of scholarly references on the web. D-Lib Magazine, 16(1–2).
32. Leetaru, K. (2012). Archiving the web: Challenges and opportunities for social science researchers. SSRN Electronic Journal.
33. DataCite Metadata Working Group. (2021). DataCite Metadata Schema Documentation for DOI. Retrieved from (https://schema.datacite.org)
34. Arnold, T., Ayers, N., Madron, J., Nelson, R., & Tilton, L. (2020). Visualizing a large spatiotemporal collection of historic photography with a generous interface. arXiv preprint. Retrieved from (https://arxiv.org/abs/2009.02242)
35. Klasen, V., et al. (2023). How we see time: The evolution and current state of temporal visualizations. Visual Communication, 22(1), 34–52. (https://doi.org/10.1080/23729333.2022.2156316)

36. Almeida, P. S. G. de, et al. (2022). Turn-based temporal media web visualization and interaction design. NOVA University Lisbon. Retrieved from (https://run.unl.pt/bitstream/10362/176242/1/Almeida_2022.pdf)
37. Wang, E., & Cook, D. (2021). Interactive visualization to explore structured temporal data. The R Journal, 13(1), 287–301.
38. Chen, J., et al. (2024). SalienTime: User-driven selection of salient time steps for large-scale geospatial data visualization. arXiv preprint. Retrieved from (https://arxiv.org/abs/2403.03449)
39. “Time Series Information Visualization – A Review of Techniques.” (2025). arXiv preprint. Retrieved from (https://arxiv.org/html/2507.14920v1)