МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет радіоелектроніки
Факультет Комп'ютерних наук
(повна назва)
Кафедра Програмної інженерії
(повна назва)

# КВАЛІФІКАЦІЙНА РОБОТА
# Пояснювальна записка

рівень вищої освіти другий (магістерський)

## Дослідження методів автоматичного оцінювання результатів інформаційного пошуку
**(тема)**

Виконав:
Студент 2 курсу, групи ІПЗм-19-2
Мустафаєв Є.О.
(прізвище, ініціали)

Спеціальність 121 Інженерія програмного забезпечення
(код і повна назва спеціальності)
Тип програми освітньо-наукова

Керівник проф. Шостак І.В.
(посада, прізвище)

Допускається до захисту
Зав. кафедри _____ З.В. Дудар
(підпис) (прізвище, ініціали)

2021

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 121 Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:
Зав.кафедри _____
(підпис)
« ____ » _____ 2021 р.

**ЗАВДАННЯ**
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента Мустафаєва Євгенія Олеговича
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів автоматичного оцінювання результатів інформаційного пошуку

затверджена наказом університету від 26.03.2021 р. № 386Ст

2. Термін подання роботи до екзаменаційної комісії 10 05 2021р.

3. Вихідні дані до роботи *проаналізувати існуючі алгоритми, що використовуються для вимог підтримки прийняття рішень, мови розробки програмного забезпечення*

4. Перелік питань, що потрібно опрацювати в роботі *мета роботи, аналіз проблемної галузі і постановка задачі, опис запропонованих варіантів оптимізації, використовувані методи та алгоритми, опис розробленої програмної системи, опис застосованих програмних рішень, аналіз можливих застосувань*

5. Перелік графічного матеріалу із зазначенням креслеників, схем, слайдів, ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)

*Мета завдання, обґрунтування доцільності розробки, постановка задачі, базові моделі, методи й алгоритми, структурно-логічна схема взаємодії даних,*

*інтерфейс програмної системи, результати дослідної експлуатації програмної системи, висновки*

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

| Найменування розділу | Консультант (посада, прізвище, ім'я, по батькові) | Позначка консультанта про виконання розділу | |
|---|---|---|---|
| | | підпис | дата |
| спецчастина | проф. Шостак І.В. | | |
| | | | |

## КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Терміни виконання етапів роботи | Примітка |
|---|---|---|---|
| 1. | Аналіз предметної галузі | 25 березня 2021 р. | виконано |
| 2. | Огляд існуючих методів | 31 березня 2021 р. | виконано |
| 3. | Розробка алгоритмів, проектування та розробка ПЗ | 15 квітня 2021 р. | виконано |
| 4. | Підготовка пояснювальної записки | 20 квітня 2021 р. | виконано |
| 5. | Спецчастина | 28 квітня 2021 р. | виконано |
| 6. | Підготовка презентації та доповіді | 03 травня 2021 р. | виконано |
| 7. | Попередній захист | 05 травня 2021 р. | виконано |
| 8. | Нормоконтроль, рецензування | 07 травня 2021 р. | виконано |
| 9. | Занесення роботи в електронний архів | 08 травня 2021 р. | виконано |
| 10. | Допуск до захисту в зав. кафедри | 10 травня 2021 р. | виконано |

Дата видачі завдання ____ _____ 2021р.

Студент _____
(підпис)

Керівник роботи _____ проф. Шостак І.В
(підпис) (посада, прізвище, ініціали)

# РЕФЕРАТ /ABSTRACT

Пояснювальна записка до кваліфікаційної роботи магістра: 96 с, 37 рис., 6 дод., 35 джерел

АЛГЕБРА СКІНЧЕННИХ ПРЕДИКАТІВ, АВТОМАТИЗОВАНЕ ПРИЙНЯТТЯ РІШЕНЬ, ІНТЕЛЕКТУАЛЬНІЙ АНАЛІЗ, ОБРОБКА ТЕКСТІВ, ПРЕДИКАТНА МОДЕЛЬ ЗНАНЬ, СЕМАНТИЧНИЙ АНАЛІЗ.

Об'єкт дослідження – моделі та методи інтелектуального аналізу текстів в вигляді питань-відповідей.

Метою роботи є аналіз та розробка моделей і комплексів програм для аналізу, оцінки й поліпшення якості доступу до даних соціальних питально-відповідних сервісах (СПВС) і універсальних пошукових системах

Метод дослідження – математичне моделювання та алгебра скінченних предикатів, методи семантичного аналізу.

В результаті роботи було розроблено предикатну модель процесних знань при виявленні структури питань-відповідей, а також реалізовано програмну систему, що реалізує метод вирішення кванторних лінійних рівнянь, що застосований для вирішення задачі логічного результату в системі питань-відповідей.

ALGEBRA OF FINITE PREDICATES, AUTOMATED DECISION MAKING, INTELLECTUAL ANALYSIS, TEXT PROCESSING, PREDICATE MODEL OF PROCESS KNOWLEDGE, SEMANTICAL ANALYSIS.

The object of research – models and methods of intellectual analysis of texts in the form of questions and answers.

The purpose of the work is to analyze and develop models and sets of programs for analysis, evaluation and improvement of quality of access to data of social question-relevant services (SPVS) and universal search engines

Research method - mathematical modeling and algebra of finite predicates, methods of semantic analysis.

As a result, a predicate model of process knowledge in identifying the structure of questions and answers was developed, as well as a software system that implements the method of solving quantifier linear equations, which is used to solve the problem of logical result in the system of questions and answers.

Я, Мустафаєв Євгеній Олегович, студент гр. ІПЗм-19-2, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів автоматичного оцінювання результатів інформаційного пошуку», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElAr KhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

# ЗМІСТ

# ВСТУП

Інтернет став важливим джерелом інформації про здоров'я для багатьох людей. У цей час у мережі доступний величезний обсяг медичної інформації. Згідно з дослідженнями, проведеним центром Pewresearch в 2018 році, 59% дорослих інтернет-користувачів у США шукали інформацію про стан здоров'я в мережі [1].

Проте, висновки інтернет-користувача про стан свого здоров'я, зроблені на основі веб-даних, можуть не відповідати реальності через наявність великої кількості невірної інформації у відкритому доступі або невміння користувача коректно інтерпретувати отримані знання. Медична інформація, отримана в мережі, може послужити сигналом до самодіагностики, самолікування або відвідування лікаря без достатніх на те підстав, і, як наслідок, завдати шкоди здоров'ю користувача. У зв'язку із цим актуальними є завдання оцінки якості медичної інформації в мережі, а також розвитку методів пошуку інформації про здоров'я користувачів і коректної її інтерпретації.

Існує багато способів доступу до медичної інформації в інтернеті: універсальні пошукові системи (Google, Bing тощо), спеціалізовані й професійні пошукові системи (Pubmed, Cochrane, Google Scholar), медичні портали. Дослідження стосується питань якості інформації в соціальних питально-відповідних сервісах (СПВС) і універсальних пошукових системах.

Метою роботи є аналіз та розробка моделей і комплексів програм для аналізу, оцінки й поліпшення якості доступу до даних соціальних питально-відповідних сервісах (СПВС) і універсальних пошукових системах і веб-сторінок про здоров'я людини. Поставлена мета досягалася рішенням наступних завдань:

– розробити метод наближеної оцінки якості даних питально-відповідного сервісу;

– реалізувати відповідний комплекс програм, провести ручну оцінку із залученням медичних фахівців;

– дослідити проблему якості даних СПВС через оцінювання користувачів-авторів.

– розробити метод оцінки компетентності користувачів медичних розділів СПВС;

– розробити метод персоналізації пошуку по колекції веб-сторінок, присвячених питанням здоров'я людини.

Використовувалися методи інформаційного пошуку, тематичне моделювання, аналіз текстових даних методами алгебро-логічних функцій [2]. Для поліпшення якості автоматичної обробки текстів розроблений модуль виправлення помилок і друкарських помилок. Крім того, для перевірки, інтерпретації й доповнення результатів автоматичних методів застосовані методи експертної оцінки.

Представлена методика експертної оцінки якості медичних розділів СПВС лікарями, розроблений комплекс програм для її практичної реалізації.

У якості однієї зі складових методу запропонована модель тематичного фокуса користувача. Крім того, розроблений алгорим персоналізації пошуку медичної інформації розширенням запиту даними медичної карти пацієнта.

Результати можуть бути використані для підвищення якості й зручності використання питально-відповідного сервісу, для підвищення якості автоматичного питально-відповідного пошуку. Запропонований алгоритм оцінки компетентності користувача СПВС у медичних темах може бути використаний для обчислення рейтингу користувача або маршрутизації нового питання конкретному користувачеві-фахівцеві з теми питання. Модуль виправлення помилок і друкарських помилок є адаптивним, тобто може бути застосований до текстової колекції будь-якого виду після напівавтоматичного навчання.

# 1 АНАЛІЗ СТАНУ РОЗВ'ЯЗАННЯ ПРОБЛЕМИ ТА ОБҐРУНТУВАННЯ ЦІЛЕЙ ДОСЛІДЖЕННЯ

## 1.1 Аналіз методів інформаційного пошуку

При аналізі текстових даних (наприклад, веб-сторінок), дослідники активно користуються поняттями теорії інформаційного пошуку. концепції, що приводяться нижче, докладно описані в джерелі [3].

Інформаційний пошук (Information Retrieval, IR) – процес пошуку у великій колекції довільного неструктурованого матеріалу (зазвичай документа), що задовольняє інформаційну потребу користувача. Під неструктурованими даними зазвичай розуміють дані, що не мають ясної, семантично очевидної структури, легко реалізованої програмно.

Тред (англ. thread) в інтернет-форумах, блогах, списках розсилання, конференціях – послідовність відповідей, сповіщень, тобто «ветка обговорення».

Сповіщення подаються в вигляді зв’язаної послідовності («ветки»), якщо їх поєднує загальна тема, або загальний ідентифікатор ветві.

Основним завданням інформаційного пошуку є розробка систем, що виконують пошук по довільному запиту. Мета такої системи – знайти документи, які є найбільш релевантними стосовно довільної інформаційної потреби користувача, повідомлюваній системі за допомогою однократних, зазвичай текстових, запитів [4].

Інформаційна потреба (Information Need) – це тема, про яку користувач у конкретний момент часу прагне довідатися більше за допомогою пошукової системи. Її слід відрізняти від запиту – текстового виразу інформаційної потреби, повідомлюваного системі.

Документ називається релевантним, якщо, з погляду користувача, він містить коштовну інформацію, що задовольняє його інформаційну потребу.

Зокрема, технології систем пошуку в мережі Інтернет багато в чому засновані на наведених вище концепціях.

У цей час інтернет використовують як середовище для спілкування, розваг, пошуку й одержання інформації. Обсяг даних у веб згодом росте експоненційно. Існує дослідницький напрямок Big Data Analysis, у рамках якого розробляються методи, у яких витяг корисного (тобто нетривіального, раніше невідомого) знання проводиться в основному за рахунок великого обсягу даних.

Універсальною точкою входу в інтернет-сервіси зазвичай є пошукова система загального призначення, наприклад, Google або Bing. Для одержання спеціалізованої інформації можуть використовуватися пошукові системи, спроектовані спеціально для знаходження інформації конкретного виду. Прикладами можуть служити Tineye – веб-сервіс пошуку по зображеннях, пошукова система для наукових публікацій Google Scholar, каталоги медичної інформації Pubmed, Cochrane. Крім того, існують різні системи, де інформація як створюється, так і споживається рядовими користувачами інтернету.

Контентом, що генерується користувачами (User Generated Content, UGC), називається будь-яка текстова або візуальна інформація довільного формату, створювана у вебі не експертами, а рядовими користувачами, зазвичай усередині соціальних сервісів.

Сервіси, що дозволяють кожному (зареєстрованому чи ні) користувачеві вільно створювати й споживати UGC, а, що також припускають спілкування, наприклад, соціальні мережі, блогі або форуми, прийнято називати соціальними. Такі сервіси зазвичай складаються з множини сторінок, кожна з яких присвячена певному питанню або вузькій темі. Формат форуму припускає обговорення довільної теми довільним числом користувачів у вигляді послідовних коментарів. Єдина логічно зв'язана бесідою послідовність коментарів форуму називається тредом. Блог – це підвид форуму, у якім рівноправне спілкування між користувачами трансформується у відносини виду «один автор – багато читачів».

Соціальний питально-відповідний сервіс (СПВС, Community Question Answering, CQA) – це сервіс, що пропонує в якості треду використовувати

формат відповідей на одне питання, що поставлено у першому повідомленні. При цьому тред, у якому відповідь на запитання знайдена і верифікована, за правилами СПВС закривається. Таким чином, СПВС відрізняється від форуму тим, що в треді не передбачається подальше спілкування після одержання правильної відповіді на запитання. Зазвичай, творці СПВС організують усередині сервісу різні рейтингові системи для стимулювання користувацької активності, наприклад, введення рейтингу або рівня користувача, голосування за кращу відповідь, гейміфікація відповідей на запитання. Нижче приводиться кілька прикладів найбільш великих і відомих СПВС:

– англомовна система, присвячена тематиці мов і технологій програмування Stackoverflow;

– більш широкий спектр технічних тем Stackexchange –;

– Yahoo! Answers – найбільший міжнародний сервіс, що підтримує множина мов (включаючи російський) і будь-яку тематику. Крім того існують великі аналоги Yahoo! Answers на корейському, китайському й російському мовах: Naver, Baidu Zhidao відповідно. У даній роботі для експериментів використовувалися питання, відповіді, а також знеособлені дані користувачів сервісу Yahoo! Answers.

## 1.2 Аналіз оцінки якості методів інформаційного пошуку

Розробка будь-яких нових методів має на увазі оцінку якості їх роботи. В області IR тестування нових підходів найчастіше проводиться шляхом їхнього порівняння з існуючими методами. Для цього науковим співтовариством розроблені стандартні заходи оцінки якості методів інформаційного пошуку. Нижче приводяться визначення тих заходів, що використовувалися для тестування методів, представлених у даній роботі.

Оцінка якості методу в інформаційному пошуку й аналізі текстових даних найчастіше припускає порівняння результатів роботи цього методу з існуючими аналогами на тестовій колекції документів, яку можна описати формально як трійку (D,Q,R), де D – множина документів, на яких тестується метод, Q – множина інформаційних потреб, R, у свою чергу, – множина трійок (q,d,r), у яких документу d, виданому по запиту, що виражає інформаційну потребу d, ставиться у відповідність оцінка релевантності q. У літературі тестові колекції часто називають золотим стандартом (gold standard, ground truth).

Асессмент (assessment) – це процес створення множин оцінок релевантності R для тестової колекції (D, Q, R). Асессмент тестової колекції зазвичай проводиться вручну спеціально запрошеними для цієї мети людьми – асесорами. Залежно від цілей системи, що тестується, інформаційного пошуку в якості асесорів можуть виступати як експерти в певній області, так і рядові користувачі інтернету. У кожному разі оцінка відповідності конкретного документа тестової інформаційної потреби носить суб'єктивний характер. Тому, для одержання надійної й об'єктивної загальної оцінки системи необхідно мати досить велике число оцінених інформаційних потреб.

Якщо в маленьких тестових колекціях можна оцінити всі можливі пари «тестова інформаційна потреба – документ», то для більших колекцій така стратегія буде занадто витратною. Тому в другому випадку застосовують так званий метод загального казана (pooling), при якому оцінюється релевантність р. З суміші з *n* перших документів, що повертаються декількома (тими, що тестуються, або просто найбільш популярними на момент оцінки) пошуковими системами [5].

Для виміру рівня суб'єктивності оцінок деяке число пар оцінюється декількома асессорами. Потім по множині еквівалентних оцінок обчислюється рівень згоди асессорів. У літературі найпоширенішим показником згоди називається зазвичай каппа-статистика Коена [5].

Каппа-статистика Коена (Kohen's Kappa statistics) – захід погодженості категоріальних оцінок, яка робить виправлення на випадковий збіг оцінок і виражається формулою:

$$К = \frac{P(A) - P(E)}{1 - P(E)}, \quad (1.1)$$

де P(A) – частка збігів;

P(E) – очікувана частка випадкових збігів.

У різних джерелах називаються різні границі прийнятності каппа-статистики, але в загальному випадку вважається, що точні граничні значення залежать від призначення даних. У даній роботі з аналогії з [6] погодженість уважається високою при К > 0,75 і неприйнятною при К < 0,4.

В області інформаційного пошуку результат роботи методу або системи часто є або може бути представлений у вигляді множини документів, видаваних по запиту. Така множина документів називається видачею. Оцінка продуктивності методу на тестових даних проводиться об'єднанням оцінок релевантності в один числовий показник, який легко зрівняти з показниками продуктивності інших методів або систем. Базовими метриками якості IR вважаються точність і повнота. Щоб дати їм визначення, розглянуто основне завдання IR – пошук релевантних документів у колекції по запиту – що наведено в таблиці 1.1 сполучених ознак:

Таблиця 1 – Класифікація документів після виконання запиту

| | Релевантні | Нерелевантні |
|---|---|---|
| Знайдені | Істинно позитивні ($t_p$) | Хибно позитивні ($f_p$) |
| Незнайде | Хибно негативні ($f_n$) | Істинно негативні ($t_n$) |

Точність P – це частка релевантних документів серед усіх знайдених:

$$P = \frac{tp}{tp + fp}. \quad (1.2)$$

Повнота R – це частка знайдених релевантних документів серед усіх релевантних (формула 1.3).

$$R = \frac{tp}{tp + fn} \quad . \tag{1.3}$$

Використання тільки однієї з оцінок точності або повноти не цілком характеризує метод, що тестується, через несиметричність даних: у колекціях досить великого розміру переважна більшість документів по запиту є нерелевантними. Наприклад, система, що видає вся множина документів колекції по будь-якому запиту, забезпечує абсолютну повноту пошуку (R = 1), при цьому буде непридатною для використання. Тому найчастіше для одержання адекватної оцінки точність і повноту комбінують за допомогою зваженого середнього гармонійного – $F_\alpha$:

$$F_\alpha = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} \quad . \tag{1.4}$$

Додатковою перевагою такої оцінки є те, що залежно від цілей системи, що підлягає тестуванню, внесок точності й повноти в загальну оцінку можна варіювати за допомогою параметра ос.

У дослідженні точність, повнота й $F_\alpha$ спосіб використовувалися при розробці методу оцінки компетентності. Якість методів оцінювалася за допомогою інших показників, які, завдяки гарним властивостям, що дискримінують, і стійкості, є де-факто стандартом оцінювання в сучасних дослідженнях – макроусереднення середньої точності й нормованої дисконтованої сукупної вигоди.

Нехай Q – множина інформаційних потреб тестової колекції (D, Q, R), {$d_1$, $d_2$,..., $d_m$} – множина усіх документів, релевантних інформаційної потреби $q_j \in Q$.

Нехай $R_{jk}$ – впорядкована множина документів, видаваних пошуковою системою по запиту $q_j$, до *k*-го релевантного документа.

Середня точність AP – це середнє арифметичне точності P множини $R_{jk}$ для всіх натуральних, що менше або рівних *ij*.

Середня точність AP – це середнє арифметичне точності P множини $R_{jk}$ для всіх натуральних, що менше або рівних *ij*.

Макроусереднена середня точність MAP – це AP, що усереднена по множині інформаційних потреб Q:

$$MAP(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j}P(R_{jk}) \quad (1.5)$$

Усі показники якості, описані вище, визначені в припущенні, що показник релевантності є індикаторною функцією, значення якої дорівнює 1, якщо документ релевантний і 0 інакше. Однак часто розробка нових методів вимагає оцінювати ступінь релевантності документа по деякій шкалі (наприклад, від 1 до 5). Щоб врахувати шкалу релевантності в оцінці, сучасні дослідження часто використовують нормовану дисконтовану сукупну вигоду.

Нехай rel(j) – функція релевантності *j*-го документа у видачі.

Дисконтованою сукупною вигодою на рівні до (Discounted Cumulative Gain at до, $DCG@k$) називається сума зважених значень функції релевантності пошукової видачі аж до k-го документа зі штрафом за низьке положення у видачі релевантних документів:

$$DCG@k = \sum_{j=1}^{k}\frac{2^{rel(j)}-1}{\log(j+1)} \quad (1.6)$$

Нормована дисконтована сукупна вигода на рівні k $NDCG@k$ – це $DCG@k$, нормалізована по значенню DCG ідеально ранжируваної видачі nq, і усереднена по всіх інформаційних потребах Q:

$$NDCG@k = \frac{1}{|Q|}\sum_{q \in Q} n_q DCG@k \quad . \qquad (1.7)$$

До завдань інформаційного пошуку відносять і такі, як класифікація й кластеризація, зокрема, тематична кластеризація документів. У її основі лежить математичний апарат теорії побудови імовірнісних моделей. Далі приводиться формальний опис імовірнісного тематичного моделювання колекції документів.

## 1.3 Опис тематичного моделювання

Тематичні моделі – це сімейство імовірнісних генеративних моделей, використовуваних для визначення тематики документів на основі їх умісту. У загальному випадку під темою розуміється імовірнісний розподіл над «словами» документа. Інший випадок – тематичне моделювання зображень, де під «словами» розуміються невеликі фрагменти, що зображують різні візуальні елементи, які зустрічаються на зображеннях. Темами тут можуть бути, наприклад, смужки, особи людей або текстура дерева. У тематичних моделях зазвичай передбачається, що кожний документ колекції містить у собі суміш різних тем, представлених з певною ймовірністю. Однією з перших тематичних моделей вважається модель імовірнісного латентно-семантичного аналізу (Probabilistic Latent Semantic Analysis, PLSA), запропонована Т. Хоффманом в 1999 році [13].

Модель тематичного фокуса користувача, яка вимагає для своєї реалізації знання про тематичний розподіл кожного документа. Для витягу тематичних розподілів документів використовувалася імовірнісна модель породження

колекції текстів – тематична модель латентного розміщення Діріхле (Latent Dirichlet Allocation, LDA) [14], заснована, у свою чергу, на PLSA.

Формально будь-яка імовірнісна модель визначається в такий спосіб. Нехай W – множина усіх термінів колекції документів D. Передбачається, що існує скінченна множина тем T и вживання кожного терміну w ∈ W у кожному документі d ∈ D пов'язане з деякою темою t ∈ T, заздалегідь невідомої. У контексті тематичного моделювання колекція документів розглядається як множина трійок (d,w,t), обраних випадково й незалежно з дискретного розподілу p (d,w,t), заданого на кінцевій множині D x W x T. Тоді можна визначити імовірнісну тематичну модель породження даних як

$$P(w|d) = \sum_{t \in T} P(t|d)P(w|t) \tag{1.8}$$

де терміни w ∈ W і документи d ∈ D є спостережуваними змінними;

t ∈ T – латентна змінна.

Побудувати тематичну модель колекції документів D – означає знайти розподіли p(w\t) для всіх тем t ∈ T и розподіл p(t\d) для всіх документів d ∈ D.

У літературі розподіли p(w|t) і p(t|d) зазвичай представлені у вигляді матриць:

$$\theta = (\theta_{td})T \times D; \theta_{td} = p(t|d). \tag{1.9}$$

Тематична модель латентного розміщення Діріхле LDA – це тематична модель при додатковому припущенні, що вектори документів Θ: d = (Θ) ∈ M і вектори тем породжуються розподілами Діріхле [13] з параметрами:

$$Dir(\theta_d; \alpha) = \frac{\Gamma(\alpha_0)}{\prod_t \Gamma(\alpha_t)} \prod_t \theta_{td}^{\alpha_t - 1}, \alpha_t > 0, \alpha_0 = \sum_t \alpha_t, \theta_{td} > 0, \sum_t \theta_{td} = 1, \tag{1.10}$$

$$Dir(\varphi_t; \beta) = \frac{\Gamma(\beta_0)}{\prod_\omega \Gamma(\beta_\omega)} \prod_\omega \varphi_{\omega t}^{\beta_\omega - 1}, \beta_\omega > 0, \beta_0 = \sum_\omega \beta_\omega, \varphi_{\omega t} > 0, \sum_\omega \varphi_{\omega t} = 1. \tag{1.11}$$

Джерело [8] дає докладне введення в дану предметну область, описує різні моделі і їх властивості.

Тематичне моделювання використовується при рішенні широкого спектра завдань:

– відстеження тематичних трендів у соціальні медіа й наукових публікаціях [9];

– кластеризація, класифікація, анотування документів і зображень [10];

– розробка рекомендаційних систем [11].

Для поліпшення якості пошуку й доступу до текстових даних необхідно в першу чергу вміти оцінювати якість цих даних. Експерименти проводилися на даних, наданих сервісом для дослідницьких цілей. Крім того, як приклад використання розроблених алгоритмів, потрібно поліпшення якості пошуку по веб-сторінках медичної тематики. Для тестування запропонованого підходу використовувалася тестова колекція веб-сторінок медичної тематики, опублікована в рамках кампанії по оцінці методів інформаційного пошуку CLEF eHealth'14 з метою розвитку цієї області досліджень.

У цей час дані СПВС медичної тематики є предметом активного інтересу дослідників, тому що поряд з питаннями й відповідями надають для досліджень інформацію (рейтинг користувача, дата створення, оцінка питання/відповіді тощо), яка дозволяє досягти більш глибокого розуміння даних, відновити користувацький і ситуативний контекст питання й відповіді. Наприклад, рейтинг користувача застосовувався як цільової функції для добору параметрів одного із запропонованих методів.

У силу обладнання СПВС, відповіді на запитання часто персоналізовані, тобто адресовані конкретно авторові питання. Відновлений контекст користувача дозволяє витягати його персональні характеристики, які можуть бути корисними в різних додатках, наприклад, у завданнях кластеризації, класифікації або колаборативної фільтрації.

СПВС – це соціальний питально-відповідний сервіс із можливістю вільної реєстрації. Сайт має дворівневу систему тематичних розділів: близько 30 розділів

верхнього рівня, які містять сумарно близько 200 підрозділів. Користувачеві, що задає питання, пропонується вибрати підходящий розділ зі списку, що випадає. Наприклад, рис. 1.1 демонструє сторінку сервісу, на якій задано питання «Чім лікувати застуду?» у тематичному підрозділі Хвороби, ліки й дано 7 відповідей.

Набір наданих даних містить близько 11 мільйонів питань і відповідних відповідей (у середньому 4,85 відповіді на запитання).

Дослідження було сфокусоване на медичних розділах сервісу:

– хвороби, ліки;
– лікарі, клініки, страхування;
– дитяче здоров'я;
– відповідає лікар.

За 2019 рік 225 427 унікальних користувачів медичних розділів задали 227 828 питань і відповіли на них. У табл. 1.2 приводяться деякі статистичні показники медичних розділів СПВС у порівнянні з даними сервісу в цілому.

Таблиця 1.2 – Порівняння статистичних показників усіх даних СПВС і його медичних розділів

| Показник | Усі | Медичні |
|---|---|---|
| Число питань | 11 170 398,00 | 227 828,00 |
| Середній розмір питання (у словах) | 9,90 | 10,00 |
| Середнє число відповідей на запитання | 4,85 | 4,13 |
| Середній розмір відповіді (у словах) | 17,15 | 22,99 |
| Середній час одержання «кращої» відповіді (у хвилинах) | 217,53 | 121,61 |
| Число користувачів | 2 690 358,00 | 225 427,00 |
| Число, що відповідали користувачів | 1 022 090,00 | 127 602,00 |
| Середнє число відповідей користувача | 248,27 | 10,94 |
| Середній рейтинг користувача (у відсотках) | | |

Зокрема, можна відзначити, що користувач медичних розділів СПВС дає в середньому значно більше відповідей (248 проти 53), причому середня довжина

відповіді також більше, чим у сервісі в цілому (23 слова проти 15). Крім того, відповідь, яка за результатами голосування співтовариства СПВС стає кращим, у медичних розділах дається швидше (122 хвилини проти 218). Дані показники можуть говорити про те, що якість медичних розділів СПВС вище, ніж якість даних сервісу в цілому. Це є однією із причин, по яких експерименти з даними СПВС фокусуються надалі тільки на медичних розділах:

– дані СПВС пройшли наступні етапи попередньої обробки:усі словоформи наведені до нижнього регістру;
– виправлені орфографічні помилки й друкарського помилки;
– відкинуто 100 слів, що найбільше часто зустрічаються;
– відкинуті слова із частотою 1;
– відкинуті небуквені символи (наприклад, розділові знаки);
– проведено стеммінг слів.

На початковому етапі дослідження був доступний менший набір даних – 128 370 питань із відповідями (у середньому 5 відповідей на запитання) у медичних розділах СПВС. Усі слова даного набору було лематизовані. Набір даних, описаний у табл. 2, має значно більший обсяг і використовувався при розробці моделей і методів, описаних у главі 3. При експериментах з даним набором використовувався стеммінг як більш продуктивний у порівнянні з лематизацією метод нормалізації слів.

У завданні персоналізації пошуку використовувалася англомовна тестова колекція, опублікована в рамках кампанії по оцінці пошуку медичної інформації CLEF eHealth в 2019 році [12]. Колекція містить близько 1 млн веб-сторінок про здоров'я людини й 50 тестових топіків, де топік – це формальний опис деякої інформаційної потреби користувача. Має структуроване представлення (часто у форматі XML). Зазвичай містить у собі запит користувача, докладний опис запиту, опис гіпотетичної ситуації, у якій запит міг бути заданий. Усі топіки складено на основі медичних карт пацієнтів, при цьому, медичні карти також додаються. Крім того, для кожного з 50 топіків надаються оцінки релевантності

частини веб-сторінок у корпусі (близько 1000 оцінок на одну інформаційну потребу), отримані методом ручної експертної оцінки.

Колекція документів зібрана методом обходу веб-сторінок і покриває широкий спектр медичних тем. Серед документів присутні як професійні статті, призначені для медичних фахівців, так і сторінки, орієнтовані на широку публіку. При створенні множин тестових інформаційних потреб використовувалися знеособлені медичні карти реальних пацієнтів з тестової колекції MIMIC II, описаної докладно в [22]. Медична карта пацієнта – це частково структурований звіт про його історії хвороби, який містить у тому числі інформацію про стан його здоров'я на той момент, коли карта була видана пацієнтові на руки.

Топік у тестовій колекції CLEF eHealth 2014 являє собою структуру даних мовою XML, що полягає з декількох полів: запит (title), розширений опис запиту (desc), опис очікуваних результатів (narr), опис гіпотетичної ситуації, у якій міг би бути заданий подібний запит (scenario), додаткова інформація про пацієнта, витягнута з його медичної карти (profile).

Приклад медичної карти пацієнта

**Admission Date: [**2014-03-28**]**
**Discharge Date: [**2014-04-08**]**
**Date of Birth: [**1970-09-21**]**
**Sex: F**
**Service: CARDIOTHORACIC**
**Allergies:**
**Patient recorded as having No Known Allergies to Drugs**

**Attending: [**Attending Info 565**]**
**Chief Complaint: Chest pain**
**Major Surgical or Invasive Procedure:**
**Coronary artery bypass graft 4.**
**History of Present Illness: 83 year-old woman, patient of**
**Dr. [**First Name (Stitle) 5804**] [**Name (Stitle) 2275**], with increased SOB with activity, left shoulder blade/back pain at rest, + MIBI, referred for cardiac cath. This pleasant 83 year-old patient notes becoming SOB when walking up hills or inclines about one year ago. This SOB has progressively worsened and she is now SOB when walking [**01-19**] city block (flat surface). [...]**
**Past Medical History:**
**arthritis; carpal tunnel; shingles right arm 2000; needs right knee replacement; left knee replacement in [**2010**];**

```
thyroidectomy 1978; cholecystectomy [**1981**]; hysterectomy
2001; h/o LGIB 2000-2001 after taking baby ASA; 81 Q0D [...]
```

Структура, показана в наступному прикладі лістингу, задовольняє формату TREC – прийнятому в науковому співтоваристві IR стандарту побудови тестових колекцій TREC (Text Retrieval Evaluation Conference) [13] – найбільш відомої кампанії по створенню тестових колекцій і оцінці методів інформаційного пошуку. Кампанії TREC організуються щорічно, починаючи з 1992 року, американським національним інститутом по стандартах і технологіям U.S. NISThttps://www.nist.gov/

## 1.4 Методи попередньої обробки текстових запитів

Приклад інформаційного запиту тестової колекції CLEF eHealth 2014 у форматі TREC

```
<query>
<title>
thrombocytopenia treatment corticosteroids length </title>
<desc>
How long should be the corticosteroids treatment to cure
thrombocytopenia? </desc> <narr>
Documents should contain information about treatments of
thrombocytopenia, and especially corticosteroids. It should
describe the treatment, its duration and how the disease is cured using
it. <scenario>
The patient has a short-term disease, or has been
hospitalised after an accident (little or no knowledge of the
disorder, short-term treatment) </scenario> <profile >
Professional female
</profile> </narr> </query>
```

У завданнях інформаційного пошуку і аналізу даних часто немає необхідності враховувати зазвичай рідкі або часті слова. Найчастішими словами в текстах природною мовою зазвичай є службові слова (частки, приводи). Крім того, у методах статистичні підходи, що використовують, до аналізу текстів, необхідно поєднувати ті самі слова в різних формах або відмінках. Тому, при проведенні експериментів з текстовими даними, проводять їхню попередню обробку.

Лематизація – це процес приведення кожного слова в документі до його нормальної форми [14].

Таблиця 1.3 – Нормальні форми деяких частин української мови

| Частина мови | Нормальна форма |
|---|---|
| Ім'я іменник | називний відмінок, однина |
| Ім'я прикметник | називний відмінок, чоловічий рід, однина |
| Дієслово | інфінітив |
| Дієприкметник | відповідне дієслово в інфінітиві |

У роботі лематизація проводилася за допомогою інструмента Mystem [15] – морфологічного аналізатора слів. У випадках, коли аналізована текстова колекція містить настільки багато документів, що проведення лематизації вимагає значних обчислювальних ресурсів, її часто заміняють стемінгом [16].

Стемінгом слова називається процес відсікання його змінюваної частини. Стемінг колекції документів – це відсікання змінюваних частин слів у всіх документах колекції. Стемінг, будучи більш простою (і тому продуктивною) технологією в порівнянні з лематизацією, має більш низьку якість нормалізації слів для мов з багатою морфологією, зокрема, для української мови. В експериментах з даними великого обсягу, що припускають попередню обробку за допомогою стемінгу, використовувався алгоритм Snowball [16] – адаптація алгоритму Портеру [17] для великої кількості європейських мов. Англомовні дані,

оброблялися за допомогою реалізації алгоритму Портеру, вбудованої в пошукову систему Terrier [18].

## 1.5 Методи оцінки якості контенту

Важливим етапом у розробці будь-якого методу оцінки якості контенту є визначення й фіксація поняття якості. Різні джерела описують даний термін по-різному. Іноді, якість відповіді сприймають як щось зовнішнє стосовно авторів питання й відповіді – «об'єктивне» знання. Напроти, виходять із того, як сприймається якість автором питання, тобто, наскільки відповідь суб'єктивно задовольняє його інформаційну потребу.

Ще одним аспектом якості питань і відповідей уважається мета питання. Підрозділяють питання на наступні типи:

—пошук інформації;

—спілкування;

—розвага (гумор).

Дана робота сфальцьована насамперед на оцінці якості даних медичної тематики, тому в дослідженні розглядаються питання, метою яких є пошук інформації.

В області аналізу якості СПВС загальної тематики описано кілька методів, що автоматично оцінюють якість даних. Деякі джерела пропонують оцінювати якість контенту «на лету», відразу по вступу питання або відповіді; інші роботи досліджують архівні дані, які містять поряд з контентом позначки-інформацію: користувацькі рейтинги, коментарі, статистику переглядів тощо. Моделі, описані в літературі, широко використовують методи машинного навчання з більшим набором ознак. Більшість робіт згадує наступні групи ознак:

– текстові ознаки, що відбивають грамотність мови, друкарські помилки, візуальне оформлення контенту, читаність тощо;

– користувацькі ознаки, такі як рейтинг, активність, досягнення, рівень експертизи в темі питання, взаємодія з іншими користувачами тощо;

– різні статистики, наприклад, кількість переглядів і кликів.

Ознаки, перераховані вище, не є залежними від тематики, однак у літературі відзначається, що дані різних тематик відрізняються в питаннях поведінки користувачів, використовуваних тезаурусах, і т.ін. Із цієї причини усе більше досліджень фокусується на конкретній темі, наприклад, [17] аналізували дані сервісу Stackoverflow (питання й відповіді на тему програмування й інформаційних технологій) з урахуванням специфіки предметної області й тематичних словників.

Тема аналізу якості медичного контенту СПВС мало представлена в сучасних дослідженнях. Описують лінгвістичні, темпоральні й користувацькі (когнітивні) аспекти якості СПВС, мотивацію користувачів, що відповідають, на 270 питаннях, семплійованих з медичних розділів сервісу Yahoo! Answers. Попередні експерименти по ручній оцінці якості 10 питань медичної тематики описані в [17]. Трьом групам асессорів – користувачам, що задавали питання у СПВС, співробітниками медичної бібліотеки й медсестрам – пропонувалося відповісти на запитання анкети, що зачіпає різні аспекти якості даних. Надалі автори збільшили вибірку до 400 питань, що стосуються всіх видів захворювань і симптомів. Це дослідження можна вважати досить масштабним, якщо врахувати той факт, що якість даних оцінювалася вручну.

Запропонований напівавтоматичний метод оцінки якості питань і відповідей Yahoo! Answers про вірус грипу H1N1 за допомогою тематичного моделювання. Автори описали найбільш важливі теми питань, типи ресурсів, на які посилалися користувачі, і медичні концепції, згадані в даних. У роботі використовувалися схожі методи, наприклад, опитування користувачів і медичних фахівців, тематичне моделювання, словники медичних концепцій. Упор, однак, робився на можливість автоматичної обробки більших масивів даних. Для цього проведений

огляд літератури на суміжну тему – аналіз повідомлень соціального сервісу Twitter, які мають набагато більш лояльну політику доступу до даних.

Дані Twitter показують великий потенціал рішення завдань аналізу медичних даних. Роботи [18] досліджують симптоматичне лікування, використання медичних препаратів, поведінкові фактори ризику захворювань, географічну локалізацію спалахів захворюваності тощо.

Описано також перші спроби побудувати інтелектуальні системи на основі знань, що витягають із даних СПВС. Наприклад, пропонується експериментальна діалогова система медичної тематики на даних Yahoo! Answers, однак не торкаються в роботі питання якості даних.

## 1.6 Огляд методів алгебраїчного представлення логічних структур

У різний час були введені різноманітні алгебраїчні структури, пов'язані з різницею порядку звичайно зв'язують циліндричні алгебри Тарського й поліадичні алгебри Халмоша [9]. Алгебра Халмоша позначається шляхом додавання операцій до кванторній алгебрі, яка у свою чергу є булевою алгеброю з певними додатковими кванторними операціями, заданими на цій алгебрі. Узяття квантора можна відобразити у вигляді деякої операції, певної у відповідній алгебрі. Далі розглядаються різні способи визначення таких операцій, однак спочатку вивчитимемо геометричне значення кванторів як операцій. Нехай змінливі *x* и *в* задані на множині *U*. Припустимо, що *B* - підмножину множини *U*, що є декартовим добутком множини *U* на себе. Множина *У* можна інтерпретувати як бінарний предикат $B(x, y)$, певний на $U^2$ і дорівнює одиниці на всіх парах $(x, y) \in B$. Множинами *UX* і *UY* позначино відповідні проекції множини $U^2$. По визначенню квантора існування вираз $\exists\ x\ B(x, y)$ означає, що на множині *U* задається унарний предикат зі змінливої в, що визначує деяка підмножина в множині *UY*, що полягає з елементів $B_Y \in UY$, для яких існують $B_X \in UX$ такі, що

($B_X$, $B_Y$) ∈ B. Таким чином, на множині *Uv* застосуванням квантора існування до бінарного предиката *B*(*x*, *y)* задається унарний предикат по змінливій в, що визначує множину елементів $B_X$. Геометрично ∃ *x B*(*x*, *y)* є проекція множини *B* на *UY*.

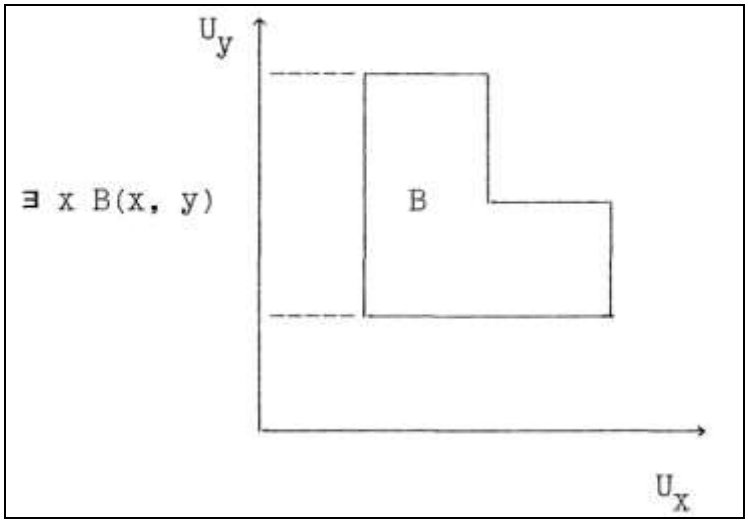


Рисунок 1.1 – Графічне представлення квантора існування

Аналогічне застосування квантора існування по змінливій *y* до предиката *B*(*x*, *y)* визначує проекцію множини *B* на *UY*. Квантор загальності позначається двоїстим видом. Геометрично ∀ *x B*(*x*, *y)* є проекція найбільшого циліндра, що лежить в *У* на множині *UY*.

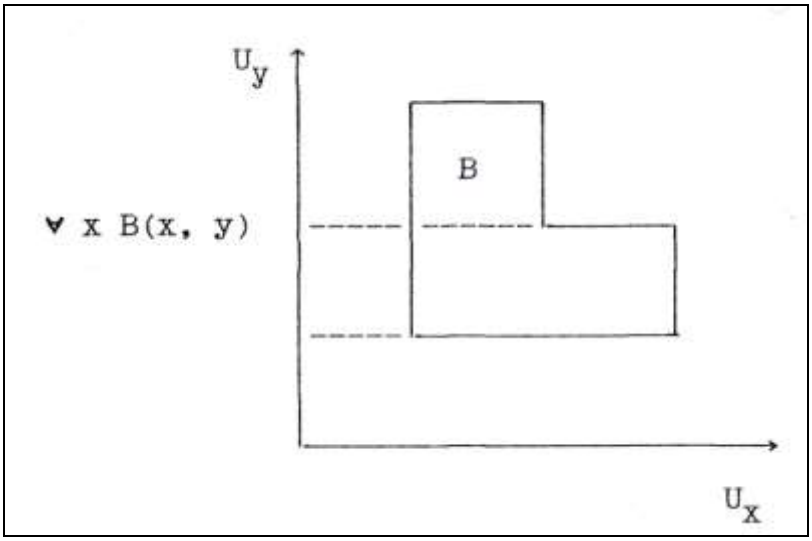


Рисунок 1.2 – Графічне представлення квантора спільності

У випадку багатомісних відношень застосування квантора існування по $k$ змінливими ($k < n$) $n$ - місцевого предиката $P(x_1,..., x_n)$, заданого на множині $Un = U_1 \times ... \times un$, можна інтерпретувати як проекцію деякої підмножини множині $Un$ на множині $U^{n-k} = \underset{j=1}{\overset{n-k}{\times}} U_j^n$. Таким чином, на множині $Un$-$k$ застосуванням квантора існування по $k$ змінливим предиката $P(x_1,..., x_n)$ позначається $n - k$ місцевий предикат. Дане враження еквівалентне наступному рівнянню:

$$\exists x_{i_1},...,x_{i_k}\ P(x_1,...x_n) = Q(x_{i_{n-k}},...,x_n).$$

При $k = n - 1$ дана рівність є частинам случаємо лінійного щодо операції диз'юнкції оператора

$$\begin{aligned} &\exists x_1,...,x_{n-1}\ P(x_1,...x_n) \wedge R_1(x_1) \wedge ... \wedge R_{n-1}(x_{n-1}) = Q(x_n) \\ &при\ R_1(x_1) = 1,..., R_{n-1}(x_{n-1}) = 1. \end{aligned} \qquad (1.1)$$

Таким чином, квантор існування можна облічити як лінійний щодо операції диз'юнкції оператор з додатковою умовою (1.1). Відповідно квантор загальності позначається двоїстим видом як лінійний щодо операції кон'юнкції логічний оператор з додатковою умовою.

У роботі [10] здійснюється аксіоматичне визначення кванторів. Нехай $H$ – булева алгебра. Квантором існування цієї алгебри називається довільне відображення $\exists : H \to H$, що задовольняє наступним умовам:

$$0=0.$$

$$a < \exists\, a.$$

$$\exists\, (a_1 \wedge \exists\, a_2) = \exists\, a_1 \wedge \exists\, a_2, \quad a, a_1, a_2 \in H.$$

Квантор загальності позначається двоїстим видом і пов'язаний із квантором існування за допомогою операції заперечення, $\forall a = \overline{\exists \overline{a}}$.

Однак різниця предикатів першого порядку властиві сутні обмеження, обумовлені тим, що в закономірність предикатів першого порядку, як це описане в роботах [17], задаються «теоретично непорушні рамки опису, які безумовно обкреслені й строго обмежені». Вимоги, пропоновані до сучасних інформаційних систем, роблять необхідним поширенням описових можливостей, використаних ними логічних мов. Можна розширити синтаксичні правила мови логіки предикатів першого порядку так, щоб вони дозволяли можливість використання змінливих предикатних символів. У наслідку такого розширення синтаксису мови здобуто систему, називану логікою предикатів другого порядку. Побудована система може включати в якості аргументів предикатів не тільки терми, але й пропозиції предикатів першого порядку. Очевидно, що така система різниць має набагато більші описові можливості, чому логіка предикатів першого порядку.

З різниці предикатів другого порядку можна зв'язати алгебру предикатів і предикатних операцій, описану в роботах [2, 4]. Таким чином, залежно від того, на якої різниці засновувати бази даних або бази знань, розглянуті раніше, необхідно зіходити з вигідних алгебр запитів і відповідей. У кожному разі базу даних у головній апроксимації можна розглядати як автомат типу

$$( L, U_{\text{зап}}, U_{\text{від}} ),$$

де $L$ – множина становищ автомата,

$U_{\text{зап}}$ – алгебра запитів і $U_{\text{від}}$ – алгебра відповідей.

Далі позначається операція $L \; X \; U_{\text{зап}} - U_{\text{від}}$, значення якої лягає в нааступному. Якщо $1 \in L$ – становище бази даних, $U_{\text{зап}} \in U_{\text{зап}}$ – деякий запит у базі даних, то $1 * U_{\text{зап}} = U_{\text{від}}$ є відповіддю на зауважений запит $U_{\text{зап}}$ у належному становищу бази. Представлення бази даних у вигляді алгебраїчної структури позначається корисним у різних випадках.

Наприклад, у випадках необхідності визначення відхилення композицій і декомпозицій баз даних, а також з ізоморфізму й еквівалентності.

Ґрунтуючись на реляційній моделі баз даних можна показати переваги алгебраїчного підходу до опису інформації. Основні поняття реляційної алгебри були вперше введені Кодом [16]. Було показане, як формулювати запити реляційних баз даних на мовах реляційної алгебри. У якості операцій реляційної алгебри були введені операції селекції, проекції, теоретико-множинного об'єднання й сполучення. Основною перевагою реляційної алгебри є те, що вона замкнута щодо всіх реляційних операцій. Це означає, що результатом будь-якої операції є нове взаємовідношення, яке має точно такий же статус, як і вихідне, у тому розумінні, що всі алгебраїчні операції застосовні до отриманого відношення. Ніяка алгебраїчна операція не може створити об'єкт, що виходить за рамки алгебри, у той же час у мовах, заснованих на різниці, для формулювання деяких складних запитів необхідно формулювати допоміжні підзапити, створювати нові конструкції. При застосовані алгебри немає необхідності в виготовленні нових конструкцій при описі нових взаємовідносин, тому можна створювати запити будь-якої складності.

При проектуванні реляційних баз даних знання про предметну область представляються у вигляді взаємин деякої безґрунтовної арності, такий спосіб демонстрації знання є годі ефективним і при проектуванні експертних систем різноманітного призначення [13]. Кожному взаємозв'язку взаємно однозначно можна поставити у відповідність кінцевий предикат, котрий, у свою чергу, кодується послідовністю нулів і одиниць. Таким чином, можливий перехід від взаємин на кінцевих множинах до двійкових код кінцевої довжини.

Інформація, що залишається в доволі великій множині двійкових кодів, вважає існування значних логічних залежностей між кодами. Так, декотрі коди можуть виражатися друг через друга за допомогою логічних операцій, що свідчить про крайність наявної інформації. Аналіз таких залежностей вимагає розробки ефективних математичних способів, що дозволяють описувати відношення між кодами на правильній логіко-алгебраїчній мові.

У якості операцій в алгебрі двійкових кодів розглядаються функції алгебри логіки, застосовувані до елементів алгебри кодів побітно. Використовується

також поняття базисних кодів, що представляють собою елементи, застосуванням до яких різних наявних операцій можна дістати довільний код. Розглядаються нескоротні щодо деяких операцій системи двійкових кодів, коли жоден елемент системи не може бути отриманий застосуванням до інших суперпозиції даних операцій. Серед великого класу всіляких перетворень інформації, представленої двійковими кодами, природно виділити клас лінійних щодо диз'юнкції або кон’юнкції перетворень кодів.

Під лінійним щодо операції диз'юнкції перетворенням, що заснований на множині двійкових кодів, розуміється оператор $A$, що переправляє один код в інакший і задовольняючий наступним двом умовам.

$$\begin{cases} A(0) = 0 \\ A(X \vee Y) = A(X) \vee A(Y), \end{cases}$$

де 0 = (0,…,0), $X = (x_1, ..., x_n)$.

Відповідно оператор, лінійний відносно кон’юнкції, позначається двоїстим видом. Відповідні умови виписуються в другому виді.

$$\begin{cases} A(1) = 0 \\ A(X \wedge Y) = A(X) \wedge A(Y). \end{cases}$$

У наслідку додавання лінійних операторів до вже наявних операцій алгебри двійкових кодів ми отримали алгебраїчну систему, що владує поруч захопливих властивостей. Отримана алгебра поряд з диз’юнктивною, кон’юнктивною алгеброю й алгеброю постановочних операцій надає можливість користуватися зручною алгебраїчною мовою для формального запису умов, яким повинна задовольняти описувана система відношень. Зокрема, це зручно, коли представлена двійковими кодами інформація має властивості лінійності й однорідності.

## 1.7 Постановка задач дослідження

На основі методів, що проаналізовано, а саме, алгебри лінійних предикатних операцій, їх властивостей і способів завдання, а також методів їх застосування при описі різних закономірностей, методів вирішення кванторних лінійних рівнянь. Можливість застосування лінійних предикатних операцій при зв'язному висновку для опису формалізованій інформації лінгвістичних закономірностей і вирішення задач розпізнавання й класифікації об'єктів розкриває актуальність даної теми.

Необхідно розробити алгоритмічну й програмну реалізацію методу вирішення кванторних лінійних рівнянь на базі алгебри лінійних предикатних операцій, формального апарата лінійно-логічних операторів і методів вирішення логічних рівнянь.

Основні задачі дослідження:

- аналіз формальних способів інтелектуальних систем;
- аналіз методів алгебраізації логіки;
- вивчення теорії лінійно-логічних операторів;
- програмна реалізація методу вирішення кванторних лінійних рівнянь для оцінювання якості підмножини даних СПВС (з набором обмежень), і, далі, спробувати узагальнити отриману оцінку на всю множина питань і відповідей медичної тематики.

Розроблені алгоритми й програмні засоби можуть бути використані при виготовленні лінгвістичного забезпечення автоматизованих інформаційних систем, в інформаційно-пошукових системах, при вирішенні задач логічного висновку в базах даних і експертних системах, а також при вирішенні задач розпізнавання й класифікації об'єктів. Метод вирішення логічних рівнянь, реалізований у програмній системі, дозволить скоротити об'єм необхідних обчислень.

# 2 ОПИС ПРОВЕДЕНИХ ТЕОРЕТИЧНИХ ДОСЛІДЖЕНЬ

## 2.1 Модуль виправлення орфографічних помилок і друкарських помилок

У завданнях аналізу й обробки колекції текстових документів зазвичай не враховують найчастіші слова, тому що вони зустрічаються практично в кожному документі, а також службові частини мови, тому що вони вносять незначний семантичний внесок у документ. Такі слова прийнято називати стоп-словами. Службові слова в будь-якому природному тексті також перебувають серед найбільш частих слів, тому в якості передобробки документів досить відкинути тільки найчастіші слова. Іноді мети розв'язуваного завдання або специфіка даних вимагають відкинути тільки деякі певні слова. Тоді складається спеціальний словник, яким керується система передобробки документів при відкиданні стоп-слів.

Поряд із частими словами відкидаються й найбільш рідкі слова – ті, що зустрічаються в колекції документів одиниці раз і пошук при роботі, наприклад, статистичних методів. У [11] показане, що частота термінів у колекціях текстів природньою мовою убуває згідно зі статечним законом (закон Ципфа), тобто, досить швидко. Експерименти на даних СПВС показують, що 61% термінів зустрічається в колекції всього 1 раз. Відкидання таких слів суттєво скорочує обсяг словника колекції.

Текстові дані, що згенеровані рядовими користувачами мережі Інтернет (UGC), часто містять велика кількість помилок, помилок, некоректних слововживань і інших ефектів, які ускладнюють аналіз і обробку текстових даних. Більше того, у питаннях і відповідях медичної тематики мається на увазі активне згадування складних професійних медичних термінів (симптомів, захворювань, найменувань лікарських препаратів, назв терапевтичних процедур), що тягне збільшення числа помилок і друкарських помилок у тексті. Усе це знижує якість автоматичної обробки документів. Наприклад, слово темпиратура, не буде

сприйматися аналізатором тексту як температура. Методи, розроблені в рамках роботи, здебільшого засновані на словниках медичних термінів. Щоб мінімізувати число помилок розпізнавання таких термінів у даних, був розроблений модуль виправлення орфографічних помилок і друкарських помилок.

Модуль надає на вхід словник $W_{ref}$ – множина пар $(\omega_1, \rho(\omega_1)), i \in [1; N]$, де $\omega_i$ – одне зі слів, на які будуть виправлені слова із друкарськими помилками, а $p(\omega_i)$ – імовірність зустріти $\omega_i$ у колекції документів (формула 2.1). Названо слова з $W_{ref}$ еталонними.

$$W_{ref} = \{(\omega_1, \rho(\omega_1)), (\omega_2, \rho(\omega_2)), \dots, (\omega_N, \rho(\omega_N))\} . \quad (2.1)$$

Для ефективного пошуку підходящого еталонного слова модуль зберігає в пам'яті словник $W_{ref}$ у вигляді інвертованого індексу триграм – буквених під-послідовностей слів, що враховують початок і кінець слова за допомогою додаткових символів: $ та _ відповідно. Визначено інвертований індекс формально. Нехай W – це множина термінів колекції документів D, деякий термін $\omega \in W$ перебуває в документі $d \in D$ на позиції *i*. Определяється слово позицією терміну ω пари (d, i) (d – ідентифікатор документа).

Інвертований індекс – це відображення $W \rightarrow L$, де кожний термін $\omega \in W$ переходить у список $I \in L$ усіх його слів-позицій у документах D.

Інвертований індекс, який реалізований у модулі – це індекс триграм усіх термінів словника $W_{ref}$, тобто $D = W_{ref}$, а W – це множина можливих триграм. Позначена вона Fuzzy Index. Поняття слів-позицій скорочене до ідентифікатора документа, у якості якого використовується термін з $W_{ref}$. Приклад Fuzzy Index для словника Wref = {abcd, bcd, bcde} наведений на рис. 2.1.

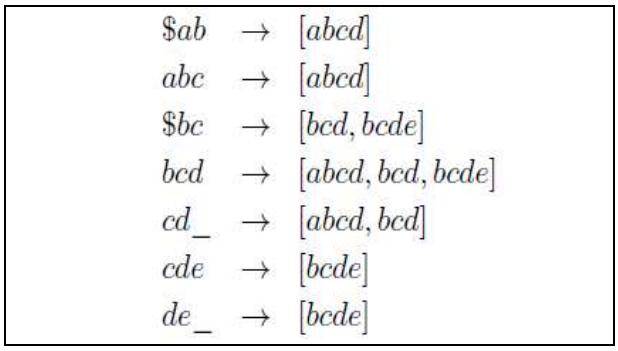


Рисунок 2.1 – Приклад інвертованого індексу триграмм

Після ініціалізації модуля словником коректних слів $W_{ref}$ і побудови інвертованого індексу триграм Fuzzy Index, наступний пошук коректного терміну для слова з помилкою або друкарськими помилками відбувається в два етапи. На першому етапі проводиться пошук слів-кандидатів для слова із друкарськими помилками.

Метод CalcEditDistance, що викликається в алгоритмі, підраховує відстань Левенштейна [29] між двома словами, яка визначається в такий спосіб. Виправленням у слові називають одну з наступних операцій:

- додавання символу;
- видалення символу;
- заміна одного символу на іншій.

Відстань Левенштейна (відстань редагування) між словами $\omega_1$ і $\omega_2$ – це мінімальна кількість виправлень, необхідних для перетворення слова $\omega_1$ у слово $\omega_2$. На другому етапі необхідно перевірити слова-кандидати на придатність і вибрати в підсумку шукане слово. Алгоритм повертає множину слів з найменшою відстанню редагування до слова із друкарськими помилками. Тоді модуль не повинен повертати слово, яке він уважає коректним. Для реалізації цієї логіки вводиться адаптивне граничне значення $T_{edit}$, яке є функцією від довжини nω слова ω із друкарськими помилками.

виправлення· орфографічних· помилок· і· друкарських· помилок·

**Вхід:** *mispelledWord*

**Вихід:** *candidates*

1: *mwsgrams* := *Getsgramsfor (mispelledword)*

2: *words* := {}

3: **for** *g* Є *mwsgrams* **do**

4: ····· *indexwords* := *Fuzzy Index.Get(g)*

5: ····· *words* := *words U indexwords*

6: **end for**

7: *candidates* := $arg\ min_{\omega \in words}$ *CalcEditDistance(w, mispelledWord)*

$$T_{edit}(n_\omega) = \begin{cases} 2, n_\omega > 6 \\ 1, 4 \le n_\omega \le 6 \\ 0, n_\omega < 4 \end{cases}$$

Рисунок 2.2 – Алгоритм пошуку коректного слова

$$T_{edit}(n_w) = \begin{cases} 2, & n_w > 6 \\ 1, & 4 \leqslant n_w \leqslant 6 \\ 0, & n_w < 4 \end{cases} . \qquad (2.2)$$

У деяких випадках (наприклад, коли в словнику $W_{ref}$ немає підходящого слова) ця відстань може виявитися настільки великою, що кандидат на виправлення значно відрізняється від слова із друкарською помилкою.

Якщо найменша відстань редагування, що повертається алгоритмом, менше $T_{edit}$ то список кандидатів ухвалюється, інакше відкидається.

Зі списку слів кандидатів необхідно вибрати слово, найбільш підходяще для виправлення слова із друкарським помилкою. Для цієї мети вводиться функція correct: $W \rightarrow W_{fixed}$, заснована на припущенні, що найбільш підходящим кандидатом є слово, з більшою ймовірністю, що зустрічається в колекції (формула 2.3):

$$correct(w) = \arg \max_{w \in candidates} p(w) \qquad (2.3)$$

З формули 2.2 випливає, що модуль виправлення орфографічних помилок і друкарських помилок не застосовується до слів довжини менш ніж 4. Це пов'язане

з тим, що для коротких слів алгоритм на рис. 1.4 повертає занадто багато кандидатів на виправлення. У цьому випадку для визначення вірного кандидата недостатньо функції correct – потрібні більш точні методи, що враховують контекст слова, що виправляється, і корпусу, у якім проводиться виправлення. Розробка подібних методів є окремою галузями досліджень автоматичної корекції текстів .

Словник $W_{ref}$ для виправлення помилок і друкарських помилок у медичних розділах СПВС сформований у першу чергу за допомогою національного корпуса мови. За основу взятий частотний словник НКРМ, що містить словоформи частоти не менш ніж 3 [27].

Тому що національний корпус складений по більшій частині з текстів художніх і публіцистичних добутків, його лексика частково не відповідає контенту, генерованому користувачами медичних розділів СПВС. Тут спостерігається дві проблеми:

– медичні розділи СПВС містять вживання спеціальних медичних термінів – симптомів, захворювань, ліків тощо;

– Ugc-контент здебільшого формується за допомогою неформального спілкування користувачів, що спричиняє використання спеціального сленгу.

Таблиця 2.1 – Статистика виправлень помилок

| Множина слів | Потужність |
|---|---|
| Wref WCQA WCQA \ Wref | 908 608 505 257 270 510 |
| Wfixed | 95 960 |

Як сленг СПВС, так і спеціальні медичні терміни з великою ймовірністю відсутні в корпусі НКРМ, тому, для формування релевантного словника необхідно доповнити його відповідними джерелами. Зокрема, для рішення проблеми 1 Wref доповнювався лексикою з довідника фельдшера [32], державного реєстру лікарських засобів [33] і міжнародної класифікації хвороб [34].

Проблема 2 припускає збагачення словника Wref лексикою, яка може вважатися некоректної з художньої або публіцистичної точки зору, однак для цілей автоматичної обробки текстів і розпізнавання слів необхідно враховувати даний лексичний зсув. У цьому випадку використовувався частотний словник тієї ж колекції питань і відповідей, на якій і проводилися експерименти по виправленню помилок і друкарських помилок: у словник Wref додавалися словоформи частотою не менш 10 (це 16% повного словника колекції), у припущенні, що слова, що досить часто зустрічаються в тексті, імовірно, не містять помилок або друкарських помилок. Константа 10 отримана шляхом ручного перегляду зрізів словника на різних частотах на предмет коректності абсолютної більшості слів певної частоти. Підсумкова множина $W_{ref}$ містить 908 608 пара (w,p(ω)), з яких 882 385 пара взята з корпуса НКРМ, а 20 052 і 6 171 пари – з медичних довідників і даних СПВС відповідно. Тому що словник $W_{ref}$ складено на основі декількох джерел інформації, існує проблема порівнянності показника p(ω) для слів з різних джерел. Дана проблема вирішена у два етапи:

– призначення пріоритету кожному джерелу. Медичні довідники вважаються більш пріоритетним джерелом, чому НКРМ, який, у свою чергу, має більший пріоритет, чому корпус СПВС;

– модифікація власної частоти слова в джерелі з урахуванням пріоритету.

На практиці це реалізоване в такий спосіб. У якості p(ω) слова ω зі словника СПВС використовується його власна частота зустрічальності в корпусі. Власні частоти слів джерела НКРМ паралельно зрушуються так, щоб мінімальна частота слова НКРМ була більше, ніж максимальна частота слова зі СПВС. Таким чином, частота будь-якого слова НКРМ більше частоти будь-якого слова СПВС. Аналогічна операція проводиться із власними частотами медичних словників щодо нових частот словника НКРМ.

З таблиці випливає, що більше половини (270 510) слів корпуса не входить у словник $W_{ref}$, що виконує роль множини «коректних» слів. Зокрема, показане, що майже одна п'ята частина словника WCQA піддалася виправленню помилки або друкарські помилки (множина виправлених слів позначене як $W_{fixed}$ :

$$W_{fixed} \subset W_{CQA} \setminus W_{ref}).$$

Для наочної демонстрації роботи модуля найбільш часті виправлення вручну розділені на 4 групи, які показані в табл. 2.2 разом з відповідними прикладами.

Таблиця 2.2 – Деякі типи помилок

| Тип друкарської помилки | Оригінальне слово | Виправлена версія |
|---|---|---|
| Орфографічні помилки | запалення<br>симтомы<br>ожега | запалення<br>симптоми<br>опіку |
| «Склеювання» сусідніх слів | вбрюшной<br>кмышце<br>личногои | черевний<br>м'язу<br>особистого |
| Заміна українських букв на латинські аналоги | коли<br>тожно<br>починаю | коли<br>можна<br>починають |

Зокрема, можна відзначити приклади виправлення помилок у медичних термінах {запалення, анальгін), що свідчить про користь застосування модуля в рішенні специфічних завдань.

Ріст обсягу даних СПВС неминуче спричиняє погіршення якості даних у середньому. Часто це пов'язане з тим, що в загальній масі контенту зменшується частка фактологічних питань – таких, відповідями на які служать конкретні факти, наприклад: Яка висота гори Кіліманджаро?. Поряд із цим, збільшується частка нетривіальних питань, що припускають складні відповіді, дискусії, збір думок або досвіду. З іншого боку, значний ріст числа користувачів зменшує середній рівень професіоналізму в співтоваристві, через що якість відповідей на фактологічні питання також падає. Дана проблема породжує потребу в методах оцінки якості питань і відповідей з метою їх подальшої фільтрації, ранжирування або перевикористання в суміжних додатках.

```
DisMed1, DisMed2 :
Питання:   Порадьте гарні краплі від [нежитю]?
Відповідь:      Спробуйте [Санорин] або [Назол] Адванс
```

Рисунок 2.3 – Приклад пари «питання-відповідь», що задовольняє паттернам

Нехай Q – множина питань. Визначити для q ∈ Q предикат $Med_k$: $Med_k$ = TRUE, якщо q задовольняє виразу « Як вилікувати захворювання d?», де d ∈ D – захворювання з множини D, що заздалегідь відомо. Визначимо предикат Medk'. Medk' (a) = TRUE, якщо у відповіді *a* згадується *j* до різних препаратів з множини ліків M. Тоді для пари ( q , a ) , q ∈ Q, a ∈ A, де *a* – відповідь на запитання q, можна визначити патерн Dis $Med_k$ як булеву функцію Q x A → {0,1}:

$$Dis\ Med_k(q,a) = Dis(q)\ \&\ Med_k(a) \quad . \tag{2.4}$$

Пара (q,a) задовольняє паттерну $DisMed_k$, якщо $DisMed_k$(q,a) = TRUE. Патерни DisMed1 і $DisMed_2$, демонструють пари, що задовольняють їм, але не задовольняє патернам $DisMed_3$, $DisMed_4$ , тощо.

## 2.2 Попередня оцінка якості питань і відповідей

Потрібно розробити метод автоматичної оцінки пари, що задовольняють патерну Dismedk'. {(q,a)\Dismedk(q,a) = TRUE}, розробити методику ручної оцінки таких пар для валідації автоматичного методу, реалізувати відповідне програмне забезпечення, оцінити якість автоматичного методу за допомогою ручного асессмента.

Сучасні дослідження сервісів Yahoo! Answers [20] і Twitter [22] формулюють загальну гіпотезу про те, що великі соціальні співтовариства чуйно

реагують на зовнішні зміни, і це можна відстежити за даними за допомогою статистичних методів.

Експерименти проводилися на зменшеній колекції – 95 002 питання з відповідями з категорії Хвороби, ліки за період із квітня 2019 по березень 2020 року. З 133 163 унікальних користувачів-авторів питань і відповідей 74 760 (56,1%) мають публічний профіль, що дозволяє відновити для них деяку інформацію, наприклад вік, стать або регіон проживання.

Один з експериментів присвячений зіставленню за часом сплесків питань певної тематики деяким подіям реального миру. Для відновлення тем у даних СПВС застосовувалося тематичне моделювання. Модель будувалася з використанням методу латентного розміщення Діріхле (LDA), реализация GibbsLDA++ з параметрами T = 100; ос = 0,5; (3 = 0,1 як найбільш стійкими згідно з результатами роботи [55], де T – число тем, а, (3 – параметри розподілу Діріхле. У якості документа розглядалася конкатенація питання й відповідей на нього. Ручний огляд тем показав, що більшість із них (71 тема) є значущими. З 100 тем вручну було відкинуто 29, що містять в основному службові слова, числа тощо. Приклади деяких значимих тем наведені в табл. 2.3.

Таблиця 2.3 – Приклади медичних топиків, отриманих за допомогою LDA

| Тема 1 | Тема 2 | Тема 3 | Тема 4 | Тема 5 |
|---|---|---|---|---|
| грип | ніс | кашель | рак | печінка |
| 37 | текти | легеня | пухлина | жовч |
| 5 | крапля | бронхіт | клітка | дієта |
| 38 | нежить | пневмон | стадія | сечовий |
| грип | ЛОР | астма | випадок | підшлунко |
| застуда | промива | сухий | небезпеч | орган |
| організ | слизуват | сироп | родимка | гострий |
| високий | пазуха | подих | рівномір | хронічний |

Також теми являють собою імовірнісні розподіли слів і не мають однозначних назв. Кожний стовпчик показує 10 самих імовірних слів у відповідній темі.

Для зіставлення була обрана доступна статистика захворюваності гострою респіраторною вірусною інфекцією (ГРВІ). Захворюваність ГРВІ в країнах за 2018-2019 рр. опублікована в щомісячному бюлетені Euroflu всесвітньої організації охорони здоров'я. Демонструє зіставлення даних про захворюваність ГРВІ й динаміки питань на тему, пов'язану з термінами «грип», «ГРВІ», «застуда» (коефіцієнт кореляції Пірсона $r_{flu} \sim 0{,}45$).

Позитивні значення кореляції свідчить про наявність часткової прямої лінійної залежності між сплесками тем і подіями реального миру. Недостатньо сильні показники залежності можуть бути обумовлено декількома причинами:

– мале число доступних публічних профілів користувачів;

– неповна відповідність теми 1 захворюванню ГРВІ (слово застуда вказує на запитання про простудні захворювання);

– наявність сторонніх факторів, що впливають на підсумковий розподіл питань (наприклад, вплив дощової погоди на сплеск нежитю може бути відсутнім, якщо температура повітря висока).

Наявність у даних структури « питання-відповіді» дозволяє визначати різні типи інформаційних потреб користувачів. Виділяють «діагностичні» медичні питання й підрозділяють їх на 2 класу по типу інформаційної потреби:

– evidence-directed – визначення захворювання за ознаками й симптомам.

– hypothesis-directed – підтвердження конкретного діагнозу й пошук інформації про лікування.

проведений експеримент по витягові пар « питання-відповідь», у яких питання належить класу hypothesis-directed. Витягали пари (q,a), $q \in Q, a \in Aq$, що відповідають патерну $DisMed_i$, який, приблизно, добре апроксимує hypothesis-directed-питання. Дані такого типу корисні, наприклад, для завдань побудови, що асистують діагностичних діалогових систем.

## 2.3 Метод автоматичної оцінки якості даних СПВС

Для рішення завдання, пропонується метод автоматичної оцінки більших обсягів питань і відповідей, що задовольняють шаблону. Метод заснований на простої моделі якості пара питань і відповідей. Крім того, при реалізації методу використовується таблиця відповідності ліків хворобам, складена на основі даних інтернет-джерел [28].

Завдання, розв'язувана в даному розділі, обмежує розглянута множина питань і відповідей до пар, що задовольняють патерну (формула 2.1), тобто питань про лікування захворювань і відповідей зі згадуваннями лікарських засобів. Якісною парою в даному контексті буде вважатися та, у якій згадані у відповіді ліки дійсно рекомендовані до застосування при згаданому в питанні захворюванні. Інакше кажучи, захворювання входить у низологічну класифікацію лікарського засобу. Пара «питання-відповідь» уважається якісної, якщо коректних пар « хвороба-ліки» більше, ніж некоректних.

При такому підході зачіпається тільки один з аспектів комплексної оцінки якості питання й відповіді, яку міг би зробити медичний фахівець, тому не можна говорити про якість пари тільки на підставі моделі (2.3). Перевага ж автоматичної оцінки полягає в тому, що вона дозволяє швидко оцінити дані такого обсягу, для якого ручна оцінка вже занадто дорога, і на основі великого числа «слабких» оцінок зробити загальний висновок про якість усієї колекції. Крім того автоматична оцінка дозволяє порівнювати між собою по якості різні набори більших даних.

У контексті методу автоматичної оцінки якості залишається нез'ясованим питання вибору параметра A; у патерні $Dismed_k$ (формула (2.1)). Для вибору до з колекції даних СПВС були земплійовані 1000 питань із відповідями, що задовольняють патерну Dismedi. Ручна перевірка пар « питання-відповідь» показала, що серед пар, що задовольняють Dismed\ тільки 53% пара відповідає вираженню « Як вилікувати захворювання d?», яке і є основною властивістю, що

моделюється функцією (2.1). Тому що критерій Dismedi показав занадто низьку для цілей дослідження точність, він був посилений до Dismed2- При такому способі відбору вже 79% пара відповідає потрібному вираженню, однак Dismed захоплює суттєво менше даних, чому Dismedi (255 питань із 1000 або 25,5%). Подальше збільшення до сильно зменшує число даних, що витягають, не збільшуючи точність значно, тому, для проведення експериментів обране A; = 2, що як забезпечує задовільну точність моделювання, незважаючи на те, що серед пар, що витягають, можна знайти й хибно позитивні.

Алгоритм автоматичної оцінки якості пари « питання-відповідь» медичної тематики представлений на рис. 2.4. Показано, що складність обробки всієї колекції питань і відповідей за допомогою даного методу лінійно залежить від числа відповідей.

```
1:  Quality := 0
2:  if ∃!d ∈ D : q.Contains(d)
3:    if ∃M_a ⊂ M : ∀m ∈ M_a a.Contains(m) & |M_a| ⩾ k
4:      estimation := 0
5:      for m ∈ M_a do
6:        estimation := estimation + IsCorrect(d, m)
7:      end for
8:      if estimation > |M_a| / 2
9:        Quality := 1
10:     end if
11:   end if
12: end if
```

Рисунок 2.4 – Алгоритм обчислення функції Quality(q, a)

Для забезпечення необхідної ефективності колекція питань і відповідей попередньо обробляється: проводиться лематизація або стеммінг повного словника колекції й кожний документ перетвориться в хеш-таблицю слів (що є технічною реалізацією концепції «мішка слів», широко застосовуваної в області інформаційного пошуку). Тоді функція Contains являє собою просту перевірку елемента на приналежність множині. Реалізація, заснована на хеш-таблицях, має середню складність 0(1).

Технічно функція Iscorrect (формула (2.2))також являє собою перевірку на приналежність пари (<i, т) таблиці лікарських засобів, рекомендованих до застосування при певних захворюваннях, що реалізується на основі хеш-таблиць. Таким чином, складність функції Iscorrect становить 0(1) у середньому.

У силу одиничності d (див. рис. 2.4, рядок 2) алгоритм має тільки один прохід по множині D. Реалізація перевірок у рядках 3 і 6 також має на увазі єдиний прохід по словникові M, тому що всі операції перевірки залежать тільки від поточного елемента $m \in M$. Це значить, що на одній ітерації можна виконати перевірку на приналежність $m$ множині $M_a$, і, в випадку успіху, обчислити значення функції Iscorrect. Таким чином, загальна обчислювальна складність функції Quality рівна O(|D| + |M|) у середньому.

Множини D і M обчислюються один раз до початку обробки колекції документів, тому величину |D| + |M| можна вважати константою. Для автоматичної оцінки якості всієї множини питань (Q) і відповідей (A) необхідно виконати функцію Quality для всіх пар колекції, число яких очевидно дорівнює числу відповідей. У підсумку, обчислювальна складність буде зростати лінійно залежно від числа відповідей: O(c|A|), де константа $c \rightarrow |D| + |M|$.

Запропонований метод (2.3) автоматичної оцінки більших обсягів даних. Цей підрозділ описує експеримент по оцінюванню питань і відповідей СПВС, що задовольняють формулі (2.1).

В експерименті використовувалися словники захворювань і медикаментів. Дані СПВС пройшли попередню обробку. Усього з 95 002 питань виділено 8 285 пара «питання-відповідь», що задовольняють патерну Dismed2, для 13 хвороб, на яких фокусувалось дослідження.

Частка позитивних оцінок методу варіюється від 0,69 у пар, що згадують нежить і отит, до 0,91 у пар зі згадуванням діареї. У середньому по захворюваннях метод дає позитивну оцінку в 80% випадків, що може свідчити про загальну адекватність і задовільну якість медичних питань СПВС, типу «Чим лікувати захворювання (d?).

# 3 ОПИС ПРОВЕДЕНИХ ТЕОРЕТИЧНИХ ДОСЛІДЖЕНЬ

## 3.1 Методи персоналізації пошуку

Класичні алгоритми пошуку моделюють користувача пошукової системи за допомогою даних про його минулу активність. Найпростішим прикладом служить історія минулих запитів користувача. У справжній роботі пропонується метод моделювання користувача пошукової системи, що використовує додаткову знеособлену інформацію про стан його здоров'я в специфічному сценарії пошуку по медичних документах.

Алгорити, досліджувані в роботі, засновані на вбудовуванні етапу персоналізації в стандартний сценарій роботи пошукової системи. При реалізації даного етапу використовувалися наступні підходи:

- модифікація (розширення) вихідного запиту пацієнта;
- переранжування шляхом змішування декількох пошукових видач.

При пошуку інформації, що стосується питань стану здоров'я, користувач інтуїтивно виходить із деяких знань про свій організм, апріорі маючи у виді, наприклад, свій вік, стать, що попередні захворювання тощо. Класичні пошукові системи в загальному випадку не мають такі знання в процесі пошуку й витягу релевантних сторінок. Це може приводити до видачі результатів, які не задовольняють інформаційну потребу користувача. Основною метою методу персоналізації, розробленого в рамках дослідження, є постачання пошукової системи апріорними знаннями про пацієнта, що вносять істотний внесок у пошук релевантних йому сторінок. Для цього використовується техніка розширення пошукового запиту.

Під розширенням пошукового запиту розуміється його модифікація шляхом додавання додаткових термінів. Початкове формулювання запиту при цьому залишається незмінної.

Споконвічний запит розширювався полями медичної карти пацієнта. У звичайній ситуації медичні карти користувачів відсутні в публічному доступі,

тому для експериментів використовувалися дані, надані міжнародною ініціативою по оцінці інформаційного пошуку CLEF (доріжка eHealth, створена для незалежної оцінки методів пошуку по медичних документах) в 2019 році.

Алгоритм персоналізації пошуку за допомогою розширення початкового запиту користувача полями його медичної карти представлений на рис. 3.1. Ключовою ідеєю пропонованого підходу є додавання до початкового запиту Initialquery декількох типів даних контексту користувача, представлених різними полями його медичної карти Medreport. Значення fieldvalue кожного поля fieldname витягає з медичної карти й, проходячи етап попередньої обробки (рядок 5 на рис. 3.1), додається до запиту Query з певною вагою weight для дотримання балансу важливості термінів початкової й додаткової частин. Запит Query у цьому випадку є змінюваною копією початкового запиту користувача Initialquery.

Оптимальні ваги визначаються перебором для кожного типу інформації, що додається. При додаванні до запиту великої кількості полів добір ваг є досить складним завданням з погляду обчислювальної потужності. У силу цих обмежень експерименти проводилися з додаванням одного або двох полів, тому що тестування додавання трьох полів вимагає вже значно більших обчислювальних ресурсів.

```
Вхід: Searchengine, Initialquery, Medreport, Medreportfieldnames
Вихід: Searchresuits
Query := Initialquery
for fieldname Є Medreportfieldnames do
weight := fieldn ame.Getw eight()
rawfieldvalue := Medreport.Extr act Field{ fieldname)
fieldvalue := rawfieldvalue.Preprocessq
Query. Addfield(fieldvalue, weight) end for Searchresults := Searchengine.Search{Query)
```

Рисунок 3.1 – Алгоритм персоналізації пошуку за допомогою розширення запиту користувача

Неможливо вмонтувати всю наявну про користувача інформацію в процес пошуку тільки шляхом розширення запиту – це приводить до значного збільшення довжини запиту й відповідному до падіння якості пошуку. Одним з

можливих рішень проблеми є змішування декількох пошукових видач в одну й наступне переранжування результатів.

Запропонований метод, що домішує до видачі по початковому запиту кілька видач по запитах, розширених полями медичної карти користувача.

Запит, розширений одночасно більшим числом полів медичної карти пацієнта, може вийти досить довгим. Дослідження показують, що збільшення довжини запиту в середньому поліпшує якість пошуку [36]. Поряд із цим, при обробці подібних запитів виникає ряд проблем. Середня довжина запиту в пошукових системах мережі Інтернет становить за різними оцінками 2 3 слова, що змушує розроблювачів сучасних систем оптимізувати метрики якості пошуку для коротких запитів. У випадку з довгими запитами пошукові системи можуть поводитися по-різному: у процесі обробки обрізати запит, намагатися шукати всі терміни в одному документі (що спричиняє падіння повноти пошуку) тощо. Крім того, у пошукових системах часто використовується модель «мішка слів» (bag of words), згідно з якою в документах і запиті не враховується порядок слів. Ця обставина може стати причиною падіння точності пошуку у випадку витягу документів по підмножині слів запиту. Така ситуація можлива через те, що довгий запит породжує велику кількість підмножин, багато з яких не є релевантними.

У якості рішення вищеописаної проблеми пропонується замість одного запиту, розширеного більшим числом полів, робити кілька запитів, розширених малим числом полів. Отримані пошукові видачі документів пропонується змішувати й упорядковувати за допомогою додаткової функції ранжирування, у загальному випадку відмінної від тієї, що використовується усередині основної пошукової системи.

У якості функції ранжирування при змішуванні видач використовувався метод Combsum:

$$CombSUM(d) = \sum_{i=1}^{n} \alpha_i score(d, L_i), \sum_{i=1}^{n} \alpha_i = 1 \quad . \qquad (3.1)$$

Тоді в кожному випадку пошукова система буде виконувати більш конкретне завдання, а у фінальній видачі буде врахований не тільки основний запит користувача, але й персональні дані його організму. Даний метод нормалізує й комбінує показники документів, що ранжуються, видавані базової пошуковою системою.

## 3.2 Метод вирішення кванторного лінійного рівняння

Ґрунтуючись на теорії лінійних логічних операторів, розроблено алгоритм вирішення рівняння.

Нехай потрібно знайти вирішення наступного предикатного рівняння

$$Q(Y) = \exists X \, ( P(X) \wedge ДО(Y, X)). \qquad (3.2)$$

Предикати $Q(Y)$ і $P(X)$ задані на галузі $U=(U_1,..., U_n)$, що має $n$ елементів, бінарний предикат $K(Y, X)$ заданий на множині $U \times U$. Потрібно облічити предикат $P(X)$, вважаючи відомими предикати $Q(Y)$ і $ДО(Y, X)$.

Враховуючи, що предикатна змінлива $X$ пов'язана квантором існування, рівняння (3.2) буде преобразовано:

$$Q(Y) = \bigvee_{j=1}^{n} (P(u_j) \wedge K(Y, u_j)). \qquad (3.3)$$

Рівність (3.3) виконується тільки в тому випадку, коли вона вірна для будь-якого значення предикатної змінної $Y$, що пробігає множину $U$. Таким чином, отримано наступні $n$ рівностей:

$$Q(u_i) = \bigvee_{j=1}^{n} (P(u_j) \wedge K(u_i, u_j)) \ . \qquad (3.4)$$

для будь-якого $i \in 1,\ldots, n$.

Значення предикатів $Q(ui)$ і $P(uj)$ позначено відповідно через $y_i$ і $x_j$, де $y_i, x_j \in \{0, 1\}$ та $i, j \in 1,\ldots, n$. Значення бінарного предиката $K(u_i, u_j)$ позначено через $k_{ij} \in \{0, 1\}$, $i, j \in 1,\ldots, n$. З урахуванням ведених позначень рівність (2.3) прийме вигляд

$$y_i = \bigvee_{j=0}^{n} (x_j \wedge k_{ij}). \tag{3.5}$$

Якщо для довільних предикатів $P(t)$ і $Q(t)$ виконується співвідношення $\psi: P(t) \to X$, то вірно також і $\psi: (P(t) \vee Q(t)) \to X \vee Y$. Операторне рівняння вигляду:

$$K(X) = Y, \tag{3.6}$$

де $K$ – лінійний логічний оператор, заданий на просторі $E_{\vee}^{n}$, з матрицею оператора

$$K = \begin{vmatrix} k_{11},\ldots\ldots\ldots, k_{1n} \\ \ldots\ldots\ldots\ldots\ldots \\ k_{i1},\ldots, k_{ij},\ldots, k_{1n} \\ \ldots\ldots\ldots\ldots\ldots \\ k_{n1},\ldots\ldots\ldots, k_{nn} \end{vmatrix}. \tag{3.7}$$

Таким чином, предикатне рівняння (3.2) еквівалентно операторному рівнянню (3.6), визначеному на логічному просторі $E_{\vee}^{n}$. Згідно із ідеєю про суцільний тип матриці лінійно-логічного постійного оператора, що діє із простору $E_{\vee}^{n}$ в себе, для оборотності необхідно й достатньо, щоб у кожному рядку й стовпчику матриці такого оператора був один і тільки один елемент, дорівнює одиниці. Якщо матриця (3.7) задовольняє зазначеним вище умовам ідеї, то вирішення рівняння (3.6) буде наступним

$$X = ДО^{-1}(Y). \tag{3.8}$$

Матриця зворотного оператора збігається із транспонованою матрицею оператора *K*. Таким чином, вирішення операторного рівняння (3.8) у матричному типі буде наступним

$$X = RT * Y. \tag{3.9}$$

У результаті вирішення предикатного рівняння (3.2) можна виписати у вигляді:

$$P(X) = \exists\, Y\,(Q(Y) \wedge K(X, Y)). \tag{3.10}$$

У випадку, якщо оператор *K* не є регулярним, вирішення предикатного рівняння виписати у вигляді (3.10) не можна, однак, використавши алгебраїчним записом предикатного рівняння (3.2), буде шукати вирішення рівняння в такий спосіб.

Операторне рівняння (3.6) напишемо у вигляді системи логічних рівнянь. Припустимо, що вектор *Y* не одиничний.

$$\begin{cases} \bigvee\limits_{j=1}^{n} (k_{1j} \wedge x_j) = y_1 \\ \bigvee\limits_{j=1}^{n} (k_{ij} \wedge x_j) = y_i \\ \bigvee\limits_{j=1}^{n} (k_{nj} \wedge x_j) = y_n \end{cases} \tag{3.11}$$

Нехай одиниці вартують в $Y$ на місцях $(d_1,...,d_{y^{(1)}}) = D$, а нулі на місцях $(z_1,...,z_{y^{(1)}}) = Z$, $D \cap Z = \varnothing$, $D \cup Z = N$, $N$=(1,..., $n$). Множина місць, на яких вартують нулі вектора *X*, будемо позначати як $L = (l_1,...,l_{X^{(1)}})$. Символом * будемо позначати місця, на яких можуть стояти нулі або одиниці. Символи * стоять на місцях $(m_1,...,m_{X^{(*)}}) = M$.

Алгоритм.

Крок 1. Ініціалізація. $i := z_1$.

Крок 2 Формується множина, що містить рівні нулю координати вектора $X$.

Крок 2.1. $j := 1$.

Крок 2.2 Якщо $K[\, i, j\, ] = 1$, то $X[\, j\, ] := l_1$.

Крок 2.3 Організується перебір індексів $j$ від 0 до $n$.

Крок 3. Індексу $i$ прирівнюється наступний елемент із множину $Z$ і перехід до п.2.1 доти, поки не будуть обрані всі елементи множини $Z$.

Крок 4. Формується множина $M$. Отримується деякий логічний вектор $X$, що містить нулі і символі *.

Крок 5. Перевірка системи (3.11) на несуперечність.

Крок 5.1 Підставлено знайдений вектор у систему.

Крок 5.2 Організуєтья вирішення отриманої системи згідно з формулою

$$\bigvee_{j=1}^{n} (k_{ij} \wedge x_j) = y_i .$$

Крок 5.3 Якщо система несумісна, то вектор не є вирішенням системи.

Крок 6. Формування вирішення системи.

Крок 6.1 У вектор $X^{(*)}$ підставляться замість першого символу * одиниця, а замість інших символів нулі.

Крок 6.2 Перехід до п. 5.

Крок 6.3 Якщо сформований логічний вектор є вирішенням системи, він запам'ятовується в масив вирішень.

Рок 6.4 Організуються різні підстановки нулів і одиниць замість символів * з переходом при кожній новій комбінації до п. 5.

7. В всі отримані вирішення системи зберігаються в масив, якщо масив вирішень не порожній. А якщо ні, то в результаті створюєтья повідомлення про суперечливість системи.

## 3.3 Вирішення задач логічного результату в базах даних

Приведений приклад ілюструє можливість використання теорії лінійно-логічних операторів і методу вирішення кванторного предикатного рівняння для обробки й зберіганні інформації в базах даних. Припустимо, що база даних поміщає інформацію про чотири заводах, що випускають деталі для машин. Нехай завод $z_1$ випускає деталі $d_1$ і $d_2$, завод $z_2$, випускає деталі $d_2$ і $d_3$, завод $z_3$ випускає деталі $d_1$ і $d_4$, завод $z_4$ випускає деталі $d_3$ і $d_4$. Наявний взаємозв'язок "завод-деталь" легко описується наступним бінарним предикатом:

$$P_1(z,d) = \begin{cases} 1, \text{ якщо завод z виготовляє деталь d,} \\ 0, \text{ в противному випадку,} \end{cases}$$

де $z \in \{z_1, \ldots z_4,\}$ і $d \in \{d_1, \ldots d_4,\}$. Таким чином, інформацію про заводи можна зберігати у вигляді формульного запису предиката $P_1(z, d)$. Далі, нехай потрібно дістати інформацію про те, які заводи виготовляють деталь $d_1$. Відповідний предикат, що розкриває дане вимогу, записується в другому вигляді:

$$P_2(d) = \begin{cases} 1, \text{ якщо } d = d_1, \\ 0, \text{ в інших випадках,} \end{cases}$$

У наслідку предикат $P_3(z)$, відповідний до шуканої інформації, позначається кванторним рівнянням вигляду:

$$\exists d\, P_1(z, d) \wedge P_2(d) = P_3(z)$$

і задає наступний взаємозв'язок:

$$P_3(z) = \begin{cases} 1, \text{ якщо завод z виготовляє деталь } d_1, \\ 0, \text{ в інших випадках.} \end{cases}$$

Вирішення даного кванторного предикатного рівняння отримується із вирішення відповідного операторного рівняння $A*X=Y$ у лінійно-логічному просторі $E_{\wedge}^{n}$. Матриця оператора $A$ має такий вигляд:

$$\begin{vmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{vmatrix},$$

а вектор $X$=(1 0 0 0). У наслідку дії оператора $A$ на вектор $X$ дістаємо вектор $Y$ рівний (1 0 1 0) і означаючий, що деталь $d_1$ служать заводи $z_1$ і $z_3$. Таким чином, операція шукання інформації, що цікавить, у базі даних заміняється операцією операторного множення. Тепер нехай необхідно облічити, які заводи випускають деталі $d_1$ або $d_3$. Використовуючи адитивну властивість лінійного логічного оператора $A$, маємо

$$A*X_1 \vee A*X_3 = A(X_1 \vee X_3) = A*X_4.$$

Вектора $X_1$ і $X_3$ створюються по предикатах, які формулюють відповідно деталі $d_1$ і $d_3$. Таким чином, відповіді на більш складні запити в базі даних також зіходять із вирішення операторного рівняння. Використовуючи алгоритм вирішення операторного рівняння, описаний у попередньому підрозділі, можна відшукувати по заданих заводах деталі, які вони виготовляють. Наприклад, нехай необхідно облічити які деталі виготовляє завод $z_2$. Отже, логічний вектор $Y$=(0 1 0 0). У наслідку вирішення операторного рівняння вигляду

$$\begin{vmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{vmatrix} * \begin{vmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{vmatrix} = \begin{vmatrix} 0 \\ 1 \\ 0 \\ 0 \end{vmatrix},$$

відносно $X$ дістаємо вектора (0 1 0 0) і (0 0 1 0). Отже, завод $z_2$ випускає деталі $d_2$ і $d_3$.

Далі припустимо, що в базі даних втримується інформація про те, які деталі застосовуються в певних машинах. Припустимо, у машині $m_1$ застосовуються деталь $d_2$, у машині $m_2$ застосовуються деталі $d_2$ і $d_3$ у машині $m_3$ застосовуються деталі $d_2$ і $d_3$ у машині $m_4$ застосовуються деталі $d_3$ і $d_4$. Даному взаємозв'язку "машина-деталь" відповідає бінарний предикат $K_1(m, d)$, обумовлений у такий спосіб:

$$K_1(m,d) = \begin{cases} 1, \text{ якщо в машині m використовується деталь d,} \\ 0, \text{ в противному випадку,} \end{cases}$$

де $m \in \{m_1,..., m_4\}$ і $d \in \{d_1,..., d_4\}$.

Аналогічно раніше розглянутому случаю, з бази даних легко відтягати інформацію про те, у яких машинах які деталі використовуються, шляхом вирішення відповідного кванторного предикатного рівняння, заміняючи його операторним рівнянням. Маємо наступне рівняння:

$$\exists\, d\, K_1(m, d) \wedge K_2(m) = K_3(d)$$

де унарні предикати $K_2(m)$ і $K_3(d)$ задають відповідно конкретну машину й конкретну деталь. Розв'язуючи дане рівняння щодо предиката $K_2(m)$ або предиката $K_3(d)$, ми й будемо відтягати з бази даних необхідну інформацію. Припустимо, що тепер необхідно вирішити більш складну задачу, а саме: облічити, які заводи

виготовляють деталі для заздалегідь заданої машини. Даній умові відповідає наступна система кванторних предикатних рівнянь:

$$\begin{cases} \exists d\ P_1(z,d) \wedge P_2(d) = P_3(z), \\ \exists d\ K_1(m,d) \wedge K_2(m) = P_2(d). \end{cases}$$

Дану систему відобразимо у вигляді одного рівняння:

$$\exists d\ P_1(z,d) \wedge (\exists d\ K_1(m,d) \wedge K_2(m)) = P_3(z).$$

Кванторному предикатному рівнянню (2.14) відповідає операторне рівняння вигляду:

$$B*T=X,$$

у лінійно-логічному просторі $E_{\wedge}^{n}$. Логічні вектори $T$ и $X$ побудовані відповідно по бінарних предикатах $K_2(m)$ і $K_3(d)$. Лінійний логічний оператор $У$ має матрицю вигляду

$$\begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix},$$

побудовану по бінарному предикату $K_1(m,\ d)$. Отже, предикатному рівнянню відповідає операторне рівняння вигляду:

$$A\ (\ B\ (T)\ ) = X.$$

Отримане рівняння перетворено до наступного вигляду:

$$C\,(T) = X,$$

де лінійний логічний оператор $C$ дорівнює суперпозиції операторів $A$ и $B$. У цьому випадку:

$$C = \begin{vmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{vmatrix} * \begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{vmatrix}.$$

У такий спосіб доволі великий список шукання у базі даних зведений до обчислення матриці оператора $C$ за допомогою операції множення матриць операторів $A$ и $B$. Наприклад, облічимо які заводи виготовляють деталі для машини $m_1$. Відповідний логічний вектор $T$ має такий вигляд (1 0 0 0). Таким чином, маємо:

$$\begin{vmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{vmatrix} * \begin{vmatrix} 1 \\ 0 \\ 0 \\ 0 \end{vmatrix} = \begin{vmatrix} 1 \\ 1 \\ 0 \\ 0 \end{vmatrix}.$$

У наслідку отримуємо, що виробниками деталей для машини $m_1$ є заводи $z_1$ і $z_2$.

# 4 ОПИС РОЗРОБЛЕНОЇ ПРОГРАМНОЇ СИСТЕМИ

## 4.1 Опис об'єктної моделі розробленої системи

Ще одним напрямом в інформатизації є створення систем інтегрованого знання й розробка методів дійової, розумової навігації по зазначених системах, у тому числі через світові комп'ютерні мережі. У цей час діло в програмно-апаратних способах, які ефективно маніпулюють природно-язиковою інформацією, перетворилася на таку необхідність, від якої починає залежати ефективність громадських інститутів і виробничих систем. Не випадково, що серед найпопулярніших програмних засобів фігурують саме програми, орієнтовані на обробку природно-язикових об'єктів: текстові й лінгвістичні редактори й процесори, програми автоматичного виправлення граматичних помилок, автоматичного редагування, природно-язикового індексування й шукання, а також програми машинного перекладу, оптичного розпізнавання текстів і т.п. Та й у самі операційні системи останнім часом усе наполегливіше починають впроваджуватися природно-язикові модулі.

На початку програмної системи – дані про моделюючу систему: розмірність задачі, семантичні ознаки (вхідний вектор даних), система логічних рівнянь і набір результуючих змінливих. Після обробки цих даних програмною системою, у ході якої діється визначення несуперечності системи логічних рівнянь і відключення несутніх ознак, на виході отримуємо весь масив вирішень системи логічних рівнянь.

Усі ці задачі неможливо розв'язати без залучення універсальної математичної мови. Уже кілька десятиліть ведуться розробки в цій галузі, були виконані роботи з алгебраізації логіки, розроблений спеціальний математичний апарат для формульної вистави відношень і операцій над ними, який називається алгеброю кінцевих предикатів. Центральне місце в алгебрі предикатів займають відношення, вони відображають властивості предметів і зв'язки між ними. Але

дотепер не існує зручного методу формульної повідоми довільних відношень, що дозволяє їхнім програмно реалізовувати.

## 4.2 Логічна структура й основні функції програми RPU

Масив вирішень виводиться в текстовому полі праворуч у верхній панелі вікна програми.

Блок-схема програми представлена на рис. 4.1.

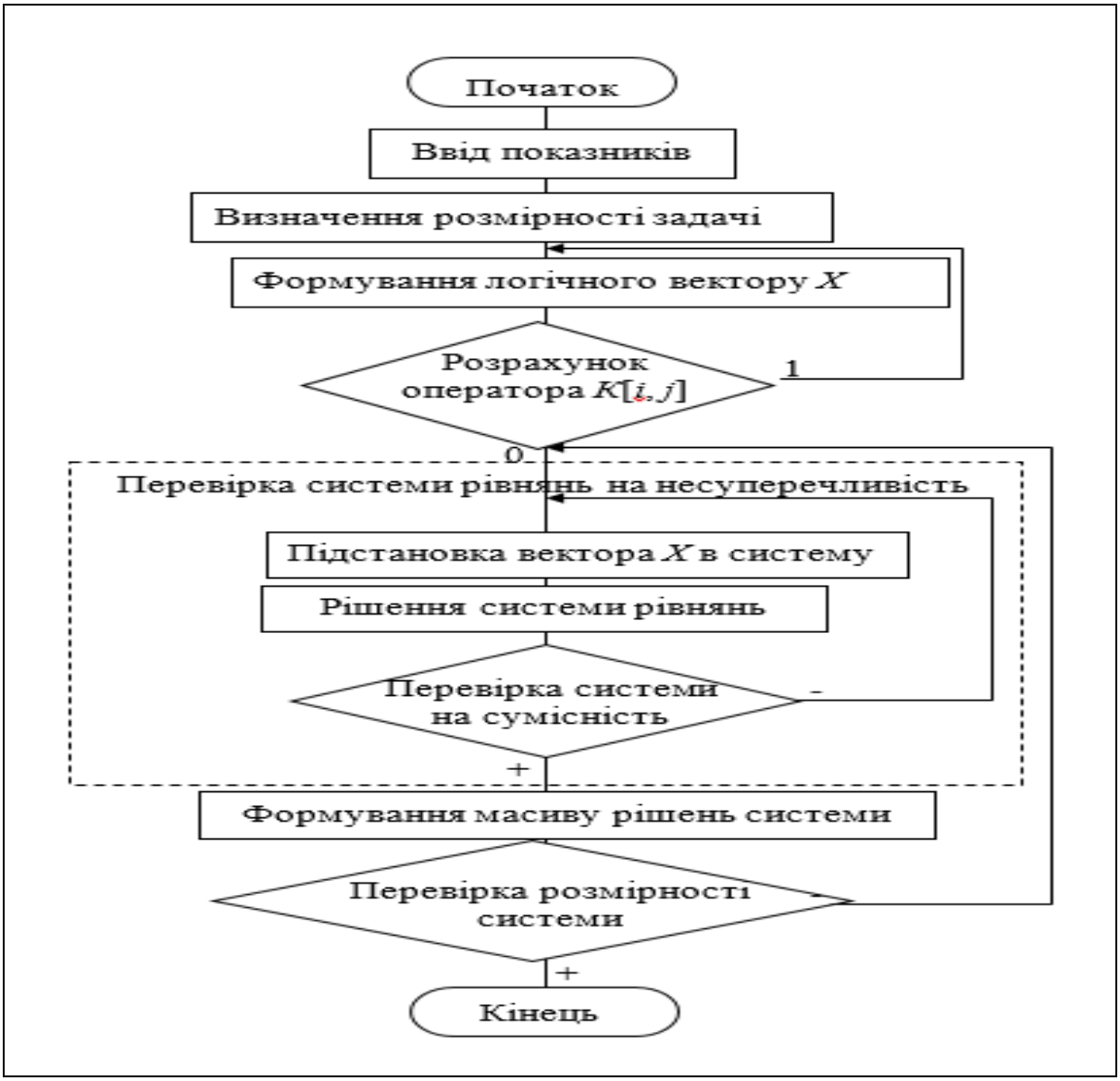


Рисунок 4.1 –Схема програми RPU

Дані, ведені в матрицю, перетворяться зі строкової вистави в числове й зберігаються як масив цілих чисел.

При вирішенні рівняння можуть бути отримано три типи результатів:

- один або кілька можливих варіантів вихідного вектора P;
- повідомлення "система несумісна", що припускає, що при заданих векторі *Q* і матриці вихідного вектора *P*(*x)* не існує;

Програма RPU призначена для рішення кванторних предикатних рівнянь описаним вище методом. Програма реалізована в системі швидкої розробки доповнень з використанням стандартної бібліотеки компонент (VCL) і існуючих розширень, що забезпечує можливість її платформної незалежності, простоту використання й наочність.

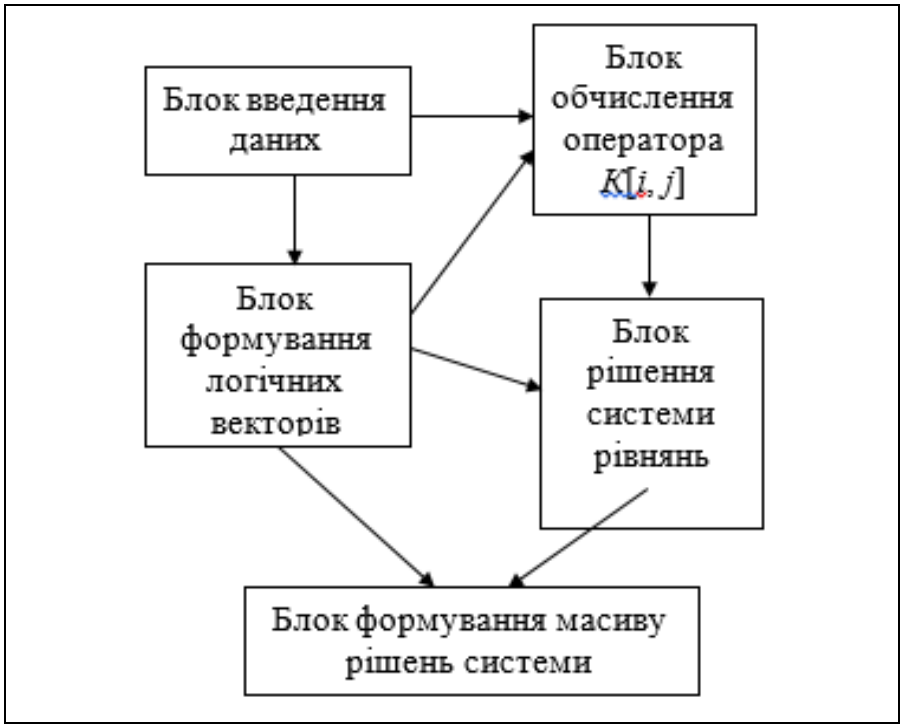


Рисунок 4.2 – Логічна структура програми RPU

Програма розв'язує предикатне рівняння

$$Q(Y) = \exists X \, (P(X) \wedge K(Y, X))$$

щодо предиката $P(X)$ відповідно описаному вище алгоритму, працюючи з логічними векторами.

Основне вікно програми презентовано на рис. 4.3.

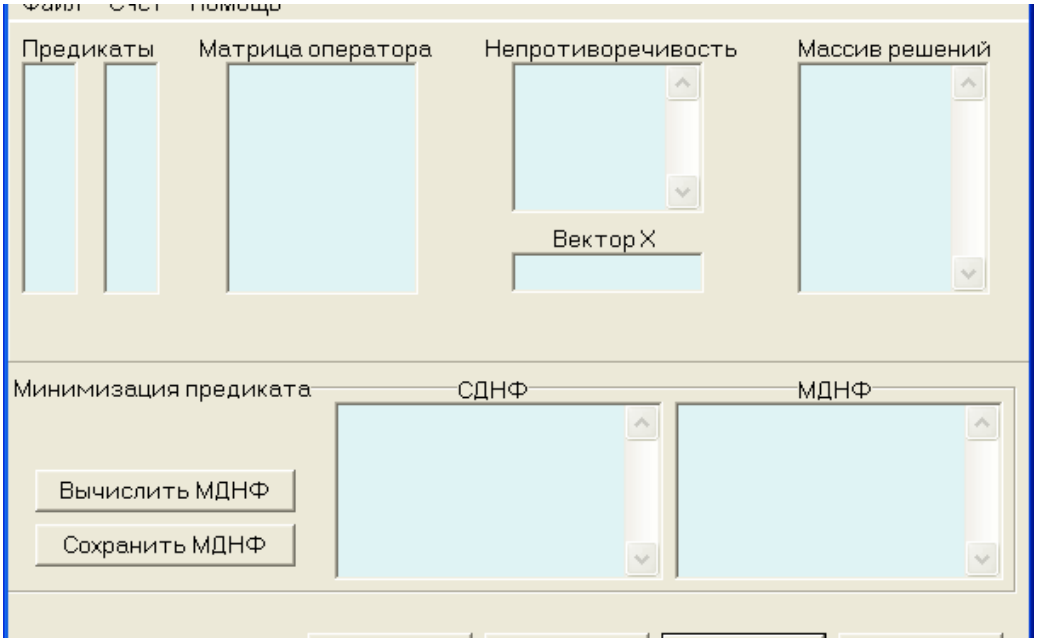


Рисунок 4.3 – Основне вікно програми RPU

По натисканню на кнопці «Старт» виникає ініціалізація програмних змінливих. По натисканню на кнопці «Такт» виникає обчислення пунктів 1-4 алгоритму, описаного в підрозділі 2.2. Результати обчислень заноситься в текстові поля «Несуперечність» і «Вектор X» (див. рис. 4.4).

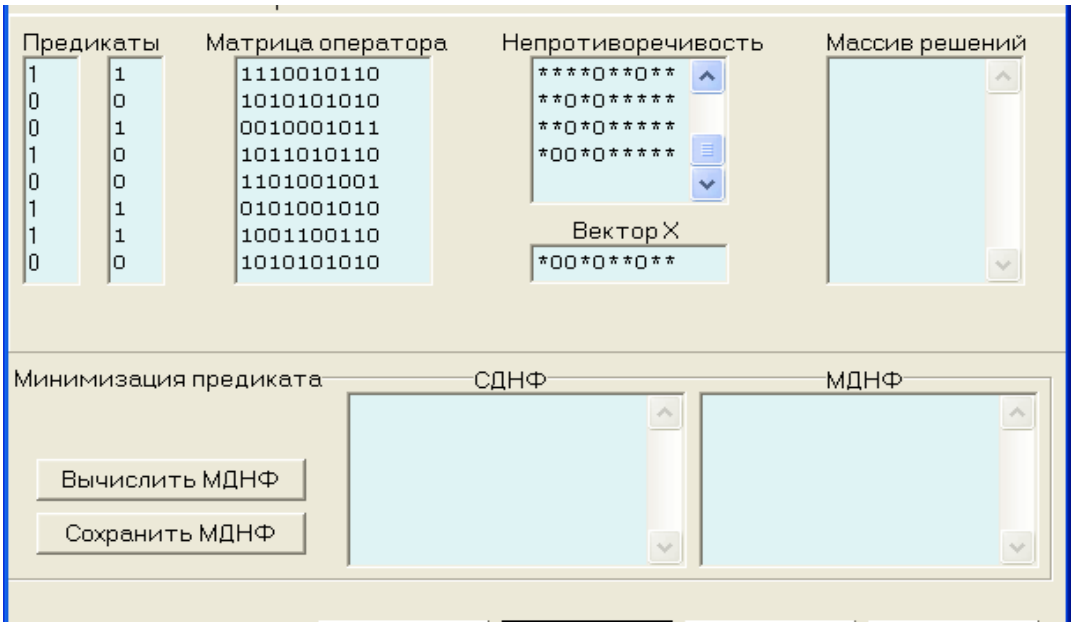


Рисунок 4.4 – Обчислення логічного вектора X

При обранні пункту меню «Несуперечність» обчислюється несуперечність початкових даних. Якщо система предикатів сумісна, програма реагує відповідним повідомленням рис. 4.5. А якщо ні, то видається повідомлення про неспільність системи рівнянь.

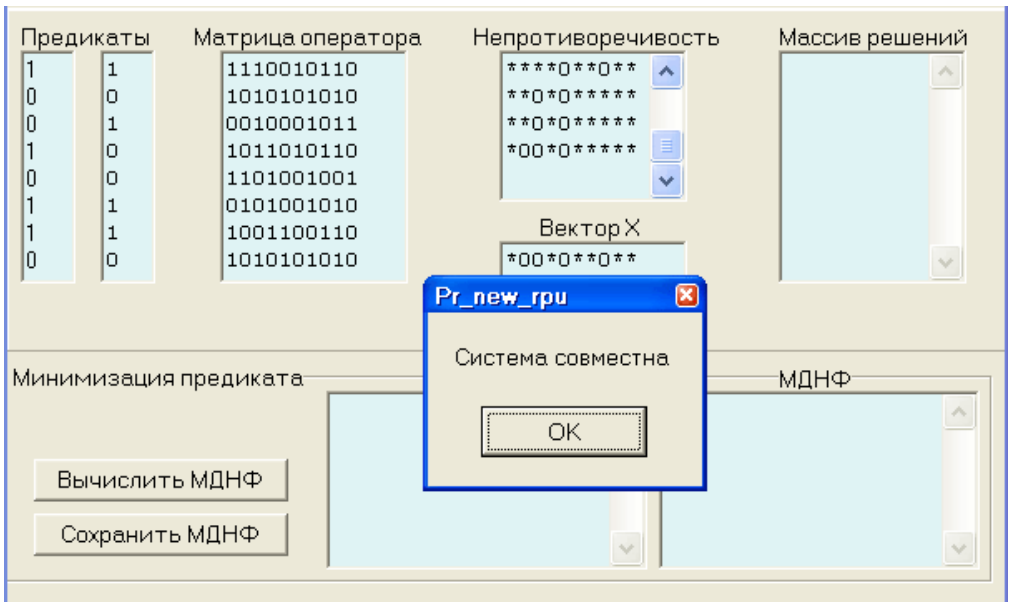


Рисунок 4.5 – Перевірка несуперечності системи рівнянь

Якщо система предикатів сумісна, то можна шукати рішення отриманої системи.

Постійне підвищення ступеня комп'ютеризації створило актуальну потребу в розробці нової теоретичної й практичної бази в галузі формального опису відмінних фізичних інформаційних об'єктів. Швидке зростання об'ємів даних у комп'ютерах, їх структурне ускладнення комп'ютеризація, що швидко прогресує, і інформатизація вимагають постійного підвищення продуктивності електронних обчислювальних машин, збільшення швидкодії й оперативної пам'яті.

# 5 ОПИС МОЖЛИВОСТІ ВИКОРИСТАННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Якщо система предикатів сумісна, то можна шукати рішення отриманої системи. Для цього, згідно з пунктами 5-6 розробленого алгоритму, підставляємо знайдений вектор у систему й організуємо рішення отриманої системи згідно з формулою

$$\bigvee_{j=1}^{n}(k_{ij} \wedge x_j) = y_i.$$

де *kij* – елементи матриці оператора,

$x_j$ – елементи знайденого вектора,

$y_j$ – елементи вектора предиката *Y*.

Згідно з пунктом 6.3 алгоритму, якщо сформований логічний вектор є рішенням системи, запам'ятовуємо його в масив рішень.

Перебираючи всі варіанти логічного вектора *X*, певні в алгоритмі, вишукуємо повне рішення (якщо воно існує) заданої системи рівнянь (рис. 5.1).

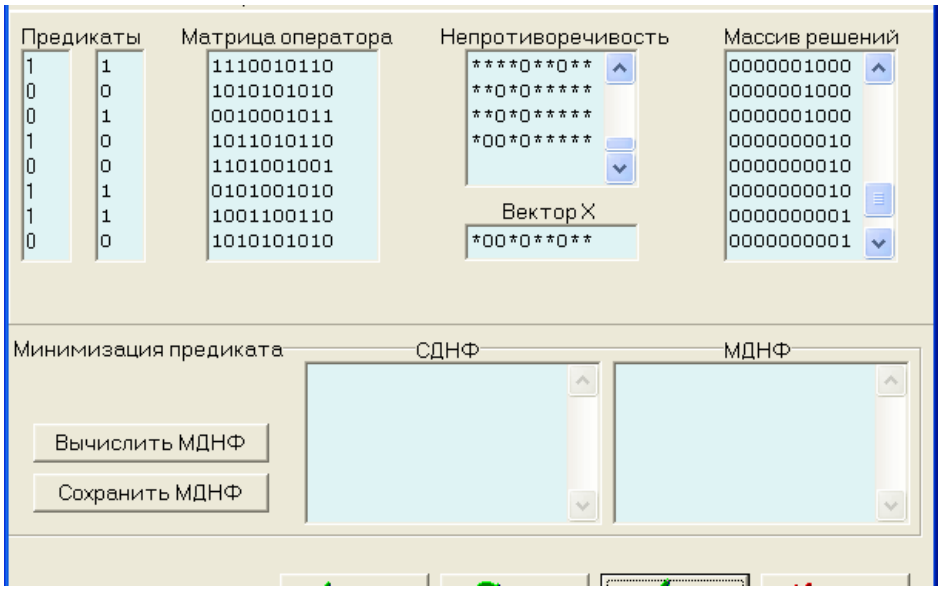


Рисунок 5.1 – Формування масиву вирішення заданої системи рівнянь

Очікування того, що роль універсального інформаційного посередника будуть виконати мови програмування, не виправдалися, оскільки виразилося, що по своїй зручності й гнучкості жодна штучна мова не може зрівнятися із природнім. У цей час уже розроблений методологічний і технічний підходи до створення й використанню інформаційних систем. Наявні в цей час інтелектуальні інформаційні системи здатні виконувати функції, що раніше значилися винятково прерогативою людини: доводити математичні теореми, перекладати тексти з однієї мови на іншій, діагностувати захворювання й виконувати багато інші функції.

Результати цілеспрямовані саме на створення сучасних принципових вирішень при побудові методів формальної вистави відношень за допомогою апарата алгебри кінцевих предикатів, орієнтованих на реальний обрахунок можливостей сучасної комп'ютерної й обчислювальної бази й нових вимог до інформаційних технологій, можуть бути застосовані для моделювання будь-яких логічних структур, що вимагають великого діапазону обчислень у реальному темпі часу.

Хоча, в головному, робота була цілеспрямована на моделювання природно-мовних структур, отримані алгоритми й програма мають гарні перспективи застосування й в інших галузях. Для опису заданої предметної галузі системою предикатних рівнянь необхідно правильно вибрати набір семантичних ознак і їх значень. Семантичні ознаки можуть бути отримані на підставі аналізу об'єктів і їх властивостей у рамках цієї предметної галузі. Приписуючи заданому предмету деяка семантична відзнака, ми ставимо йому у відповідність певне значення цієї ознаки. Потім можна застосовувати розроблену в дипломі програмну систему для моделювання будь-якої предметної галузі.

Можливість програмної реалізації формул, які описують предикати або відношення, важлива при проектуванні АСУ, при розробці природно-язикового інтелектуального інтерфейсу.

# ВИСНОВКИ

В процесі виконання кваліфікаційної роботи магістра, розроблені в рамках даного дослідження методи, загалом кажучи, можуть бути застосовні до будь-якої тематики, для якої існує власна термінологічна база. Застосування методів до нових даних у більшості випадків має на увазі формування підходящих тематичних словників, а також добір відповідних параметрів і граничних значень у конкретних додатках.

У даній роботі було розроблено алгоритми та програма, що реалізує метод вирішення кванторних лінійних рівнянь в галузі виправлення друкарських помилок у СПВС. Програма написана в системі швидкої розробки, з використанням стандартної бібліотеки компонент (VCL) і існуючих розширень, що забезпечує можливість її платформної незалежності, простоту використання й наочність.

Програма, створена в даному проекті на базі теорії лінійно-логічних операторів і методу вирішення кванторного предикатного рівняння, може бути використана для вирішення задачі логічного результату в базах даних, тобто для обробки й зберігання інформації в базах даних, а також для створення природно-язикових інтерфейсів та систем запитів-відпоідей.

У роботі були вивчені й проаналізовані формальні способи інтелектуальних систем: способи представлення знань залежно від конкретних галузей застосування систем; формальні мови, що дозволяють представляти знання в пам'яті комп'ютера. Розглянуті галузі застосування алгебраїчних методів в автоматизованих системах.

Знайдено практичні додатки сучасної абстрактної алгебри в базах даних і інтелектуальних системах. На базі застосування алгебраїчних методів у теоретичнім програмуванні були розроблені різні транслятори з мов високого ступеня й різні алгоритмічні алгебри.

Показана перспективність використання розглянутого методу вирішення логічних рівнянь в інформаційних системах, зокрема, у базах даних. Він забезпечує можливість одержання не тільки інформації, що безпосередньо зберігається в базі даних, але й похідної інформації, отриманої на базі базисної інформації. Задачі одержання похідної інформації безпосереднім образом пов'язана із задачею результату в інтелектуальних системах, при цьому, у випадку застосування алгебраїчного підходу до опису похідної інформації, виділяється певна алгебраїчна система – алгебра запитів, у термінах якої похідна інформація записується через базисну.

Для того щоб довільний запит можна було виражати яким-небудь видом через базисні запити, на множини запитів уводяться алгебраїчні операції, що дозволяють оперувати із запитами. Аналогічно подібні алгебраїчні операції повинні бути введені й на множини відповідей. У цьому випадку способами наявних алгебраїчних операцій відповідь на довільний запит можна обчислити у відповідності зі структурою запиту, записати його через запити, відповіді на які вже відомі. Розглянуті в роботі множини запитів і відповідей є алгебрами запитів і відповідей.

Отримані наступні результати:

– проведений детальний аналіз проблемної галузі;

– проведений порівняльний аналіз методів вирішення логічних рівнянь;

– алгоритми, що використовують тематичні особливості текстових даних, значно поглиблюють їхнє розуміння, дозволяючи досягти більшої якості в порівнянні з методами аналізу текстів без прив'язки до конкретної теми;

– спроектована й розроблена програмна система, що реалізує метод вирішення кванторних лінійних рівнянь;

– метод вирішення кванторних лінійних рівнянь застосований для вирішення задачі логічного результату в системі питань-відповідей.

На підставі цього можна зробити висновки, що отримані результати можуть бути використані при виробництві лінгвістичного забезпечення автоматизованих інформаційних систем, в інформаційно-пошукових системах, при вирішенні задач

логічного результату в базах даних і експертних системах, а також при вирішенні задач розпізнавання й класифікації об'єктів.

Розроблений наближений алгоритм оцінки якості медичних розділів питально-відповідних сервісів. Аналіз результатів проведених експериментів показав, що питання й відповіді медичної тематики мають прийнятна якість для сценаріїв повторне використання й витягу знань.

Алгоритми розроблені в рамках даного дослідження, можуть бути застосовні до будь-якої тематики, для якої існує власна термінологічна база. Застосування методів до нових даних у більшості випадків має на увазі формування підходящих тематичних словників, а також добір відповідних параметрів і граничних значень у конкретних додатках.

Результат чисельного експерименту на даних СПВС продемонстрував більш високий рівень якості ранжирування користувачів за допомогою запропонованого методу в порівнянні зі стандартною рейтинговою системою сервісу.

Для рішення підзадачі орфографічної корекції слів при їхній нормалізації для подальшого застосування методів автоматичної обробки текстів реалізований і адаптований для медичної тематики модуль виправлення орфографічних помилок і друкарських помилок.

Область аналізу даних медичної тематики досить молода, при цьому має істотний потенціал по застосуванню в майбутніх системах обробки й видачі інформації. Оцінка якості текстових даних медичної тематики в інтернеті є важливим завданням, тому що інформація поганої якості може завдати потенційної шкоди здоров'ю користувача.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Fox S., Duggan M. Health Online 2021. – 2013. – Режим доступа: http://www.pewinternet.org/2021/01/15/health-online-2021/.

2. Интернет как источник получения потребителями информации о здоровье, медицине, препаратах // Дайджест HealthIndex360. — Synovate Comcon Healthcare, 2020. — Т. 19.

3. Methods of multidimensional classification in problems of linguistic localization / Shubin I., Kozyriev A. // Proceedings of the III International Conference "Innovative Technologies in Science and Education". November 14, 2019 in Amsterdam, The Netherlands, 2019. pp 398-402

4. Beloborodov A., Kuznetsov A., Braslavski P. Characterizing Health-Related Community Question Answering // Proc. of the 35th European Conf. on IR research (ECIR'13): LNCS.. Vol. 7814. 2020. , P. 680-683.

5. Beloborodov A., Braslavski P., Driker M. Towards Automatic Evaluation of Health-Related CQA Data // Proc. of the 5th International Conf. of the CLEF Initiative (CLEF'14): LNCS. - Sheeld, UK. Vol. 8685. 2020. , P. 7-18.

6. Sieg A., Mobasher B., Burke R. Inferring User's Information Context from User Pro les and Concept Hierarchies // Classification, Clustering, and Data Mining Applications. | 2019., | P. 563-573.

7. Chirita P., Firan C., Nejdl W. Personalized Query Expansion for theWeb // Proceedings of SIGIR'07 Conference. | 2017. | P. 7-14.

8. Speretta M., Gauch S. Personalized Search Based on User Search Histories // Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence. - 2019 . P. 622-628.

9. Chetverikov G., Puzik O., Vechirska I. Multiple-valued structures of intellectual systems //Proceedings of the with Internations Computer Sciences and Information Technologies (CSIT). 2016, 7589907. -pp. 204-207

10. Ould-Amer N., Mulhem P., LIG at CLEF 2015 SBS Lab //Working Notes of CLEF'2019 Conference. | 2019.

11. Chikina V., Dudar Z., Shabanov-Kushnarenko S. The modeling of some intelligence functions // Radio electronics and computer science. - Kharkiv: KNURE, 2003. – № 3. – pp. 166-172.

12. The Methods of Adaptation in Computer-Based Training Systems / Shubin I., Umiarov K.// Proceeding of 2015 Information Technologies in Information Business Conference (ITIB) 7 – 9 October, 2015, IEEE Catalog Number CFP15D13-PRT pp. 64-67.

13. O. Lassila. Introduction to RDF Metadata// World Wide Web Consortium, – November 1997. http://www.w3.org/TR/Note-rdf-simple-intro-971113.html.

14. The Architecture Domain// World Wide Web Consortium, - November 2003. http://www.w3.org/Architecture.

15. The Interaction Domain// World Wide Web Consortium, - November 2003. http://www.w3.org/Interaction/.

16. M.F. Bondarenko, Z.V. Dudar, N.T. Protsay, V.V. Cherkashyn, V.A. Chykyna, Y.P. Shabanov-Kushnarenko, “Algebra of predicates and predicate operations” Radio electronics and informatics, no. 1, 2004, pp. 5-22.

17. The protégé-2000 project// Stanford, California, - 2000-2014. http://protege.stanford.edu/index.html.

18. O. Lassila. The XML Family of Specifications: A Practical Guide// Addison-Wesley, - 2002. http://shop.barnesandnoble.com/booksearch/isbninquiry.asp?isbn=0201703599

19. Semantic Web Development // World Wide Web Consortium, - January 2000. http://www.w3.org/2000/01/sw/

20. Semantic Web // World Wide Web Consortium, - September 2016. http://www.w3.org/2016/sw/.

21. Naming and Addressing: Uris, Urls // World Wide Web Consortium, - October 2013. http://www.w3.org/Addressing/

22. G.G. Chetverikov, I.D. Vechirska, S.S.Tanyanskiy, “The methods of algebra finite predicates in the intellectual system of complex calculations of telecommunication companies,” International Conference Proceedings Crimean Microwave and Telecommunication Technology (CriMiCo), 6959425, 2014, pp.

23. Shubin, I., Snisar, S., Zhyrnov, V., Slavhorodskyi, V.// Practical Application of Formal Representation of Information for Intelligent Radar Systems 2018 International Scientific-Practical Conference on Problems of Infocommunications Science and Technology, PIC S and T 2018 - Proceedings, 2019, pp. 433-436, 8632103

24. DTD for XML Schema // World Wide Web Consortium, - August 2002. http://www.w3.org/TR/xmlschema-1/#nonnormative-schemadtd

25. XSL Transformations (XSLT), W3C Recommendation // World Wide Web Consortium, - November 1999. http://www.w3.org/TR/1999/ Rec-xslt-19991116.

26. Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation // World Wide Web Consortium, - February 1999. http://www.w3.org/TR/2019/Rec-rdf-syntax-19990222/.

27. Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation // World Wide Web Consortium, - March 2020. http://www.w3.org/TR/2020/Cr-rdf-schema-20000327/.

28. OWL Web Ontology Language // World Wide Web Consortium, - September 2004. http://www.w3.org/2014/OWL/.

29. OWL Web Ontology Language Reference, W3C Recommendation // World Wide Web Consortium, - February 2014. http://www.w3c.org /TR/owl-ref/.

30. Web Ontology // World Wide Web Consortium, - March 2014. http://www.w3.org/Help/siteindex.html#webont.

31. DAML+OIL Reference Description // World Wide Web Consortium, - March 2020. http://www.w3.org/TR/daml+oil-reference

32. SWRL: A Semantic Web Rule Language Combining OWL and Ruleml // World Wide Web Consortium, - November 2003. http://www.ruleml.org.

33. Integration of information Systems: Bridging Heterogeneous Databases. / Editing by Amar Gupta, Sloan School of Management, Massachusetts Institute of Technology.-1996.

34. G. Wiederhold. Mediators in the architecture of future information systems// IEEE Computer, -March 2016. - p. 38-49.

35. Gruber T. Towards principles for the design of ontologies used for knowledge sharing// International Journal of Human-Computer Studies, 43: 907-928,- 2014.

36. N. Guarino. Some ontological principles for designing upper level lexical resources. In Proceedings of the First International Conference on Language Resources and Evaluation, Granada, 2008. pp. 3–15