МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Комп'ютерних наук
Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Дослідження методів оптимізації пошуку схожих текстів із застосуванням Elasticsearch
(тема)

Виконала:
студентка 2 курсу групи ІПЗм-20-4
Мезенцева Д.В.
(прізвище, ініціали)

Спеціальність 121 Інженерія програмного забезпечення

Тип програми Освітньо-наукова

Керівник доц. Бабій А.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ проф. Дудар З.В.

2022 р.

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 програма
(освітньо-професійна або освітньо-наукова)

Освітня програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«___»_____2022 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентки Мезенцевої Дар'ї Володимирівни
(прізвище, ім'я, по-батькові)

1. Тема роботи Дослідження методів оптимізації пошуку схожих текстів із застосуванням Elasticsearch.

затверджена наказом університету від «17» січня 2022 р. №51Ст

2. Термін подання студентом роботи до екзаменаційної комісії «16» травня 2022 р.

3. Вихідні дані до роботи: технічне завдання, календарний план, пояснювальна записка

4. Перелік питань, що потрібно опрацювати в роботі: мета роботи, аналіз предметної галузі і постановка задачі, дослідження існуючих методів аналізу схожих документів, вивчення можливостей їх використання, дослідження методів оптимізації існуючих методів пошуку схожих текстів.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Вивчення наявної літератури	17.01.2022	виконано
2	Вивчення технології	31.01.2022	виконано
3	Збір вхідних даних	07.02.2022	виконано
4	Проведення експериментів	21.02.2022	виконано
5	Підготовка пояснювальної записки	21.04.2022	виконано
6	Підготовка презентації та доповіді	01.05.2022	виконано
7	Нормоконтроль, рецензування	20.05.2022	виконано
8	Занесення диплому в електронний архів	22.05.2022	виконано
9	Допуск до захисту у зав. кафедри	23.05.2022	виконано

Дата видачі завдання 17.01.2022 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Бабій А.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 92 с., 28 рис., 3 табл., 19 джер.

ПОШУК, СХОЖІ ТЕКСТИ, ОПРАЦЮВАННЯ ПРИРОДНОЇ МОВИ, WORD2VEC, ТЕКСТОВИЙ КОРПУС, ELASTICSEARCH, ВЕКТОРИЗАЦІЯ.

Об’єктом дослідження є моделі та методи пошуку схожості документів. Метою цієї роботи є аналіз різних моделей представлення документів для визначення їх схожості та дослідження методів оптимізації пошуку схожих текстів із застосуванням Elasticsearch.

Популяризація векторного уявлення слів для представлення величезного корпусу слів у вектор компактної довжини, що дозволить здійснювати швидкий пошук за ними – це і є наукова значимість.

При впровадженні цієї системи компанії можуть зробити клієнтів більш лояльними через швидший і точніший пошук відповіді на їх питання – це практична значимість. Крім цього, не варто забувати про економічну ефективність – за рахунок прискорення отримання часу на відповідь можна скоротити ресурси компанії.

SEARCH, SIMILAR TEXTS, NATURAL LANGUAGE PROCESSING, WORD2VEC, TEXT BODY, ELASTICSEARCH, WORD EMBEDDING.

The object of research is the models and methods of document similarity. The purpose of this work is to analyze different models of document representation to determine their similarity and to study methods for optimizing the search for similar texts using Elasticsearch.

Popularization of the vector representation of words to represent a huge body of words in a vector of compact length, which will allow you to quickly search for them - this is scientific significance.

When implementing this system, companies can make customers more loyal because of a faster and more accurate search for an answer to their questions - this is a practical significance. In addition, do not forget about the economic efficiency - due to the acceleration of getting time for the answer, you can reduce the company's resources.

Умови публікації пояснювальної записки

Я, Мезенцева Дар’я Володимирівна (прізвище, ім’я, по батькові), студентка групи ІПЗМ-20-4 здобувач вищої освіти на другому (магістерському) рівні кафедра програмної інженерії, (повна назва кафедри) заявляю: моя кваліфікаційна робота на тему “Дослідження методів оптимізації пошуку схожих текстів із застосуванням Elasticsearch”, (назва роботи) що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання. Я ознайомлений (а) 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

Актуальність та новизна дослідження. Кваліфікаційна робота базується на вивченні часового параметра в пошуковій релевантності та його значущості в існуючих методологіях зважування термінів. Всі існуючі підходи до зважування терміну розраховують вагу тільки на основі частотних розподілів, не розглядаючи різні контексти термінів. За допомогою цього дослідження ми розглядаємо термін вік, який являє часовий контекст для підвищення результатів пошуку та рекомендацій. Попередні дослідження були зосереджені на різних аспектах зважування термінів, таких як розподілення, шаблон випадків, формулювання IDF тощо, але майже жодний з них не працював над віком термінів. Були деякі дослідження, засновані на часових рекомендаціях та моделюванні користувачів.

У цьому дослідженні я спробувала зібрати ці поняття і сформулювати поняття віку терміну і нормалізованого за часом терміну схем зважування. Було використано три різні набори даних різних доменів, тобто новини, уривки веб-відповідей та рекомендації щодо дослідження. Деякі набори запитів і очікуваний набір результатів даються для вибраних наборів даних, які розглядаються як золотий стандарт для оцінки продуктивності різних моделей. У цьому дослідженні було порівняно шість різних схем зважування термінів, що включають дві стандартні методології, тобто TF-IDF, BM25 та їх відповідні часові нормалізовані варіанти. І розширена модель вбудовування тексту, USE і нормалізований варіант.

Метою магістерської роботи є проектування зручної та швидкої системи для пошуку схожих текстів. Ця система дозволить знаходити документи подібних тематик швидко та прозоро для кінцевого користувача та його інтересів.

Завдання дослідження. Для досягнення вищезгаданих цілей необхідно виконати таку роботу:

– обробка тексту. Комп'ютер розуміє текст як упорядкований набір символів. Він не бачить за ним ніякого сенсу, тому необхідно перетворити текст на такий вид, щоб методи машинного навчання могли знайти між ними сенс;

– вивчення теоретичної частини з векторизації слів. Векторизація слів (word embedding) - новий підхід. Він дозволяє уявити слово у вигляді вектора чисел невеликої довжини, який значно зменшує простір всього корпусу слів та надає зручні механізми для роботи з векторами;

– опис застосовуваних алгоритмів машинного навчання;

– проектування алгоритму визначення оптимальних параметрів моделі. Слід розробити процес знаходження оптимальної моделі, як саме до неї підібрати параметри;

Основне завдання полягає в тому, щоб грамотно побудувати систему, яка швидко і якісно видавала б список схожих документів. Кожен документ буде представлений вектором, тегом та id. Завдяки тому, що документ представлений як вектор, порівнювати документи між собою стає швидше. Тег та id дозволяє однозначно пов'язати його з оригінальним екземпляром сутності.

Наукове значення. Популяризація векторного уявлення слів для представлення величезного корпусу слів у компактний вектор довжини, що дозволить здійснювати швидкий пошук по них.

Практична значимість. При впровадженні цієї системи компанії можуть зробити клієнтів більш лояльними через швидший і точніший пошук відповіді на їх питання.

2 АНАЛІТИЧНИЙ ОГЛЯД

2.1 Завдання аналізу текстів

Текстові дані використовуються дуже часто. Можна сказати, що інтернет – це величезна колекція текстів різних форматів. З їх допомогою вирішують велику кількість завдань [1]:

– прогноз рейтингу запису в блозі за текстом. Успішність запису можна оцінити ще до публікації;

– визначення емоційного забарвлення коментаря (позитивний чи негативний). Це може стати в нагоді при аналізі відгуків про банк від клієнтів, при виявленні негативного відгуку з'являється можливість швидко вжити будь-яких дій і підвищити лояльність клієнта;

– визначення тематики наукової статті при додаванні в бібліотеку. Після цього можна або автоматично зарахувати текст до виявленої категорії, або запропонувати автору варіанти вибору;

– кластеризація новин за сюжетами. Про ту ж саму подію пишуть різні новинні сайти і має сенс групувати новини з однаковим змістом;

– пошук слів, схожих за змістом на данне. Наприклад, слова «стілець» і «крісло» трохи відрізняються за змістом, але все-таки означають приблизно те саме. Для комп'ютера обидва слова – це набір символів, і за допомогою простих алгоритмів не можна визначити, що це синоніми. Про те, як робити такі висновки з даних, буде розказано пізніше.

Існують і складніші завдання на текстових даних [2]:

– виділити всі згадки імен у тексті. Це завдання виділення нових сутностей;

– побудова короткої анотації тексту;

– створення моделі, що відповідає на питання. Вона приймає на вхід питання у довільній формі та намагається знайти на нього відповідь. При цьому відповідь будується по великому структурованому корпусу, наприклад, вікіпедії;

– створення тексту, схожого на заданий набір. Було б цікаво використати, наприклад, зібрання творів Л.М. Толстого й у ньому побудувати модель, що генерує тексти, схожі на розповіді автора.

Люди спілкуються та виражають інформацію природною мовою. Дослідження в галузі штучного інтелекту, зокрема обробки природної мови (NLP), стосуються автоматичного аналізу, представлення та генерації людської мови. Генерація мови є в центрі уваги сфери генерування природних мов (NLG). Розробка систем штучного навчання, які можуть розуміти і генерувати природну мову, була однією з давніх цілей штучного інтелекту. Останні десятиліття стали свідками вражаючого прогресу в обох цих проблемах, що породило нові підходи. Зокрема, досягнення глибокого навчання за останні кілька років привели до нейронних підходів до генерації природної мови (NLG) [3].

2.2 Історія аналізу текстів

Раніше ці завдання було важко вирішити за допомогою обчислювальних пристроїв, можна було сказати неможливо, тому що не було достатнього обсягу знань для навчання та технологій, які б обробляти дані за прийнятний час.

За останні 20 років обсяги даних різко збільшилися. У статті «Технологічний потенціал світу для зберігання, передачі та обчислення інформації» Хілберта та Лопез було оцінено світовий технологічний потенціал для зберігання, передачі та обчислення інформації. Для цього було проаналізовано 60 аналогових та цифрових технологій у період з 1986 по 2007 рік. Було підраховано, що за 2007 рік обсяг даних дорівнював 276,12 мільярдам гігабайт. Стрімке зростання обсягу цифрових даних

почалося з 20-х років (див. рис. 2.1). Це дозволило отримати величезний масив текстових даних, починаючи з перетворення книг у текстові файли, поступового переходу газет у формат інтернет-газет до текстових обговорень на форумах та соціальних мережах.

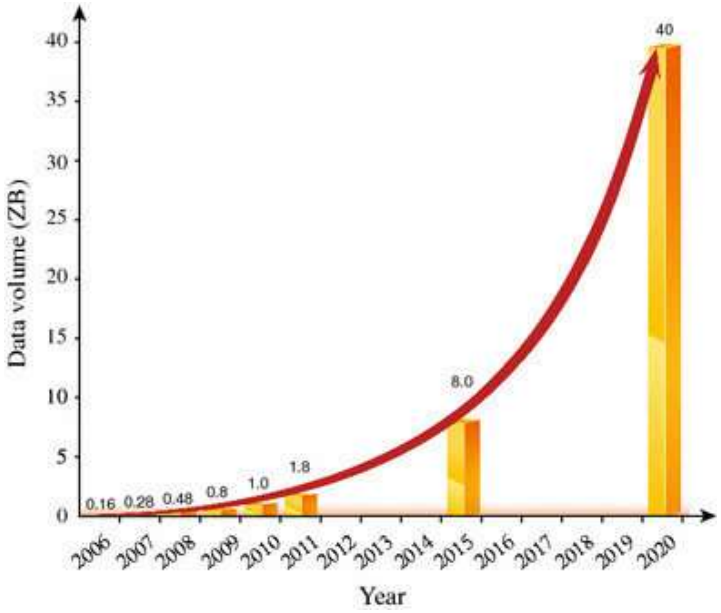

Рисунок 2.1 – Зростання цифрових даних

Технології також не стояли на місці. Для людини 10 років не такий вже й великий термін, проте для технологій це ціла вічність. Вони розвиваються, прискорюються, непомітно стають невід'ємною частиною нашого життя і змінюють його. За останні півтора десятиліття світ технологій зазнав помітних змін. Ще в 2005 досить часто зустрічалися люди, які не вміли користуватися стільниковим телефоном, а сьогодні й самі стільникові телефони стали пережитком минулого – на зміну їм прийшли мобільні пристрої із сенсорним керуванням. Ці технології дозволяють обмінюватися даними, створювати їх, збирати, як ніколи раніше.

2.3 Мета дослідження

Аналізуючи простий підхід TF-IDF, термін зважується на основі розподілу частоти в корпусі. Реальною проблемою в цьому методі є припущення, що частотний розподіл залишається постійним з часом, не розглядаючи різні контексти для різних термінів. Наприклад, розглянемо два терміни, «COVID19» і «нейронні мережі», які мають різні роки походження. І припустимо, що в конкретному корпусі обидва ці терміни зустрічаються рівну кількість раз в документі і всьому корпусі. Загалом, існує, ймовірно, менше документів, що містять термін «COVID19», ніж документи, що містять термін «нейронні мережі», просто тому, що «COVID19» є відносно новим терміном, тоді як «нейронні мережі» - це термін, який використовується з десятиліття. Однак вони були б зважені аналогічно, не розглядаючи різницю в походженні.

Враховуючи цю невизначеність в терміні зважування, я пропоную використовувати нормалізований за часом підхід для зважування термінів, який відображає вік терміну. Оскільки словниковий запас змінюється з часом, наша мета полягає в тому, щоб визначити вік терміну на основі його першого використання та поточного року та розрізнити документи на основі віку використовуваних термінів. Отже, я пропоную зважувати терміни не тільки за їх частотою, а і за часом. Для цього дослідження я розглянула три різноманітні набори даних, а саме trec news, web answer passage та CiteULike, а також очікуваний набір результатів, сформульованих людьми як основна правда. Результати формулюються та оцінюються на основі точності, повноти, F1 та NDCG оцінок. А потім я порівняю ций підхід до класичних схем зважування термінів, тобто TF-IDF, BM25 та USE, щоб підтвердити гіпотезу. Експериментальні результати показують істотні поліпшення над базовими моделями для аналогічних рекомендацій.

3 ДОСЛІДЖЕННЯ ІСНУЮЧИХ МЕТОДІВ АНАЛІЗУ СХОЖИХ ДОКУМЕНТІВ

Інтенсивно досліджується завдання подібності документів. Модель Bag-of-Words була одним з найперших представлень векторних документів фіксованої довжини в усьому словнику, де два документи вважаються подібними, якщо вони мають однаковий набір слів. Однак порядок слів не має значення, і неподібні документи можуть мати однакове представлення, якщо в парі документів зустрічаються ті самі слова. Наприклад, "X повільніше, ніж Y" матиме той самий вектор документа, що й "Y повільніше, ніж X". Кращі моделі Bag-of-Words інтегрували різні функції зважування для вектора документа. Наприклад, ваги TF-IDF були використані, щоб дозволити певним словам впливати на обчислення подібності більше, ніж інші слова, які часто зустрічаються у всьому корпусі. Однак підхід не вдався у виведенні зв’язку між синонімами, наприклад, Лікар не буде схожий на Доктор. Більше проблем виникає, коли слово можна використовувати в кількох контекстах (полісемах), як Ruby та Apache ZooKeeper, що зменшує запам’ятовування.

3.1 Представлення щільних векторів

У цьому розділі ми обговорюємо численні семантичні моделі, які використовуються для генерування векторних уявлень слів, параметризації та різних архітектур, щоб обґрунтувати вибір гіперпараметрів та рішень моделі, використаних пізніше в дипломній роботі.

3.1.1 Word2vec

Word2Vec — це набір алгоритмів для обчислення векторного представлення слів, припускаючи, що слова, які використовуються в подібних контекстах, є семантично близькими. Спочатку створюється словник, а потім розраховується векторне представлення слів. Векторне представлення засноване на контекстній близькості, суть якої полягає в тому, що слова, що знаходяться в тексті поруч з тими ж словами, матимуть близькі координати векторів – слів [4].

Word2Vec представляє слова як вектори у багатовимірному векторному просторі, де схожі слова мають тенденцію бути близькими один до одного. Для того, щоб отримати таке уявлення, було запропоновано дві методики: неперервна торба слів (continuous bag-of-words, CBOW) та неперервний пропуск-грам (continuous skip-gram). В архітектурі неперервної торби слів модель передбачує поточне слово з вікна слів навколишнього контексту. Порядок слів контексту не впливає на передбачування (припущення торби слів). В архітектурі неперервного пропуск-граму модель використовує поточне слово для передбачування навколишнього вікна слів контексту. Архітектура пропуск-граму надає словам найближчого контексту більшої ваги, ніж словам контексту віддаленішого.

Представлення щільних векторів витягує семантичні зв’язки на основі спільного зустрічання слів у наборі даних. Точність представлення цих двох слів залежить від того, скільки разів модель бачить ці слова в одному контексті по всьому корпусу. Більше спільних зустрічей слів і контексту під час навчання змінює приховане уявлення, що дозволяє моделі мати більше успішних прогнозів на майбутнє, що призводить до кращого представлення слова та контексту у векторному просторі.

Слова «йти» і «крокувати» — це синоніми, вони мають схоже значення. Однак для комп'ютера це просто рядки, причому не дуже схожі: вони мають різну довжину,

різні літери. Просто за цими двома рядками комп'ютер не може сказати, що вони мають однаковий сенс.

При аналізі текстів хочеться вміти оцінювати, як схожі різні пари слів. Це можна зробити за допомогою текстових даних. Виявляється, слова зі схожим змістом часто зустрічаються поряд з одними і тими самими словами. Інакше кажучи, схожі слова мають однакові контексти. На цьому принципі заснований метод word2vec.

Отже, потрібно описати кожне слово ω за допомогою вектора розмірності d за формулою:

$$\omega \to \vec{\omega} \in R^d$$

При цьому було б непогано мати компактні уявлення слів, тобто величина d повинна бути невеликою. Також передбачається, що схожі слова повинні мати близькі вектори (наприклад, за косинусною метрикою). Нарешті, до числових векторів можна застосовувати арифметичні операції: складати, віднімати, множити на коефіцієнт [5].

При пошуку вектора ω робота відбуватиметься з ймовірностями за формулою:

$$p(\omega_i|\omega_j) = \frac{\exp(\langle \vec{\omega_i}|\vec{\omega_j} \rangle)}{\sum_w \exp(\langle \vec{\omega_i}|\vec{\omega_j} \rangle)}$$

Ця функція показує, наскільки можна зустріти слово $\vec{w}_i$ в контексті слова $\vec{w}_j$, тобто поруч. Векторні уявлення слів налаштовуватимуться так, щоб ймовірності зустріти слова, що знаходяться в одному контексті, були високими. До функціоналу для кожного слова входять ймовірності зустріти його разом з k словами до і після нього за формулою:

$$\sum_{i=1}^{n} \sum_{j=-k}^{k} logp(\omega_{i+j}|\omega_j) \to max$$

Оптимізувати цей функціонал можна за допомогою градієнтного стохастичного спуску. Word2Vec враховує контекст слова та водночас зменшує обсяг даних. Він здатний це робити завдяки наступному.

Завданням методу CBOW є прогноз слова на підставі прилеглих слів. У skip-gram зворотне завдання – передбачення набору прилеглих слів на підставі одного слова (див. рис. 3.1).

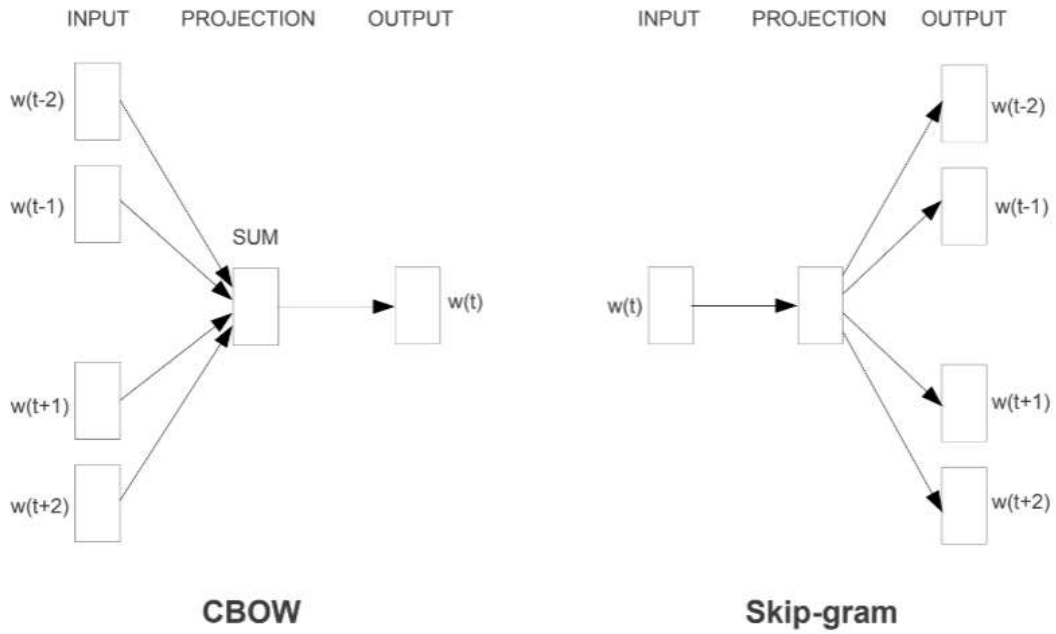

Рисунок 3.1 – Архітектура методів CBOW і skip-gram

У вікні розміру N, модель CBOW використовує контекст w (t ± 1..N) для передбачення поточного слова w(t), тоді як Skipgram передбачає контекст, заданий поточним словом.

Кількість векторних вимірів відображає розмір проекційного шару, показаного на рисунку 3.1. Прихований шар вивчається без нагляду під час тренування нейронної мережі, як у звичайних нейронних мережах. У той час як проекційний шар є матрицею, що складається з ваг прихованих шарів, об’єднаних із вхідним шаром, де матриця є спільною для всіх слів у словнику.

На рисунку 3.2 показано докладну архітектуру моделі CBOW з вікном розміру 1, яку також називають моделлю біграм, з N числом векторних вимірів, що зображують розмір прихованого шару зі словниковим запасом розміру V. Мета полягає в тому, щоб передбачити друге слово у вікні, заданому першим. Всі V слова в словнику закодовані в вхідному шарі, що означає, що для вхідного слова лише одна одиниця в x 1 до x V буде 1, а решта - 0.

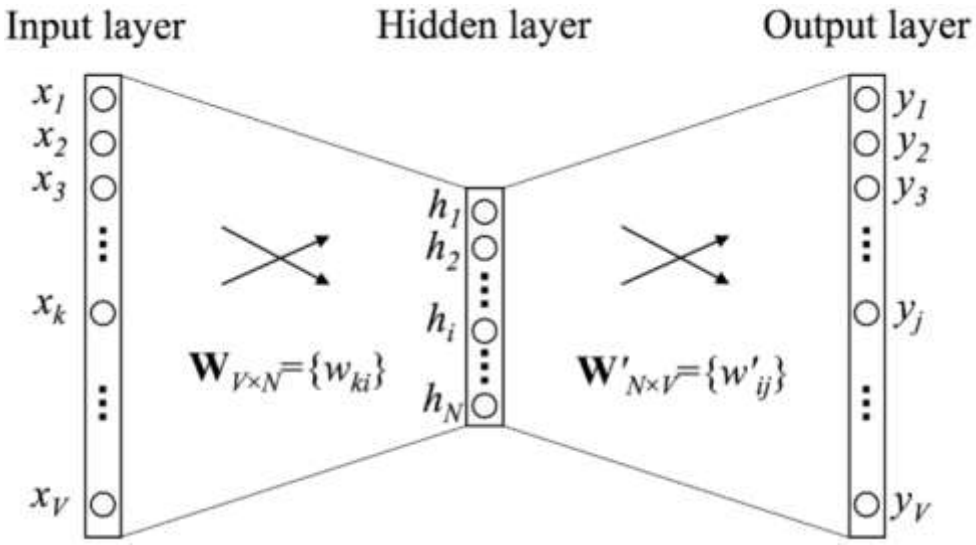

Рисунок 3.2 – Архітектура моделі CBOW з вікном розміру 1

Враховуючи вхідне слово x_i, модель біграм прогнозує друге слово y_i у контекстному вікні розміру 1, використовуючи вхідний і вихідний шари розміру V (розмір словника) і прихований шар розміру N. Матриця ваги $W_{V\cdot N}$ зображує N-мірні ваги слів словника V, де позначається матриця активації, необхідна для обчислення результату.

Жодна стаття про word2vec не обходиться без наступного сюжету (див. рис. 3.3).

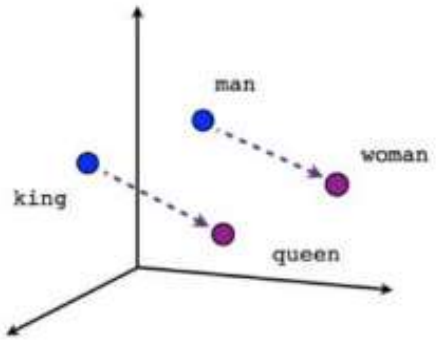

Рисунок 3.3 – Аналогії у векторному просторі

Такі уявлення інкапсулюють різні відносини між словами, такими як синоніми, антоніми або аналогії. Наприклад, вектор «король» відноситься до вектора «королева» як вектор «чоловік» відноситься до вектора «жінка». Іншими словами, отримаємо наступну картину у формулах:

$$\overrightarrow{queen} \approx \overrightarrow{king} + \overrightarrow{somevector},$$
$$\overrightarrow{woman} \approx \overrightarrow{man} + \overrightarrow{somevector}$$

Результати тренування word2vec можуть бути чутливими до параметризації. Нижче наведено деякі важливі параметри тренування word2vec.

Вікно контексту. Розмір вікна контексту обмежує слова, які використовуються в контексті під час тренування, вдвічі більшим. Наприклад, розмір вікна 5 включатиме 5 слів ліворуч і 5 слів праворуч для кожного спостережуваного слова в реченні як контекст. Збільшення розміру вікна може включати контекстні слова, які не мають відношення до поточного слова. Однак, зменшення розміру вікна може охопити відносини між словами та стоп-словами, що також не є бажаним.

Розмірність. Якість вкладання слів зі збільшенням розмірності зростає. Але після досягнення якоїсь точки гранична вигідність згасатиме. Зазвичай розмірність векторів встановлюють у межах між 100 та 1 000.

Недовибирання – це механізм прискорення часу навчання. Замість того, щоб оновлювати вектори всіх слів у словнику на кожній ітерації, оновлюватиметься лише частина словникового запасу. Вибираючи випадкові слова зі словника, ймовірність того, що вони подібні до поточного слова w (t), дуже низька. Недовибирання спрямована на максимізацію подібності між словами, які зустрічаються разом у контексті, і мінімізування подібності між w(t) і вибраними словами замість усього словникового запасу.

Кількість потоків. Реалізація C пропонує паралелізм під час тренування, щоб скоротити час навчання, встановлений кількістю потоків. Однак, аналізуючи код, виявляється, що не існує замикаючих потоків, які пишуть в матриці нейронних ваг, деякі можуть перезаписувати один одного. Методика називається «Асинхронний стохастичний градієнтний спуск», що скорочує час тренування без значного впливу на точність. Однак він не може бути відтвореним, оскільки порядок переписування є випадковим і може не підходити для контрольованих експериментів. Тому в роботі використовується один потік.

3.1.2 Word2VecF

Word2VecF є іншою реалізацією Word2Vec, де вхідними даними є файл, що містить кортежі слів і контекстів (замість файлу з усім текстом), що дає більше контролю над тим, що подавати в мережу, з можливістю використання довільного контексту як показано на рисунку 3.4. Векторизацію моделі можна змінювати, повторюючи певні пари контексту кілька разів протягом тренувальної фази. Крім того, тренування в Word2VecF відбувається помітно швидше, оскільки вилучення

кортежів попередньо обробляється, поряд з підрахунком словникового запасу та підвибіркою.

I	like
I	Icecream
I	Icecream
...	...
I	document_id_5
like	document_id_5
...	...

Рисунок 3.4 – Приклад вхідних даних в Word2VecF

Керуючи вхідними даними в Word2VecF, можна повторювати певні кортежи, щоб додати ваги/зміщення до отриманих вкладень. Довільний контекст можна додати, щоб додати глобальний контекст.

Побудувати вектор документа можна за допомогою середньозваженого вектора слів документа. Однак різниця полягає в тому, що додаючи довільні контекстні слова, наприклад ідентифікатори документів, ми можемо змусити слова в одному документі передбачити вектор документа замість сусідніх слів. Це приклад того, як можна використовувати глобальний контекст, слова більше не прив’язані до сусідніх слів у вікні так, як усі слова в документі.

Word2VecF показує більше слів, які зустрічаються в тому самому документі, що й «Забезпечення якості», тоді як Word2Vec показує слова, які можуть займати те саме або сусіднє місце з «Забезпечення якості».

Word2VecF показує більше слів, які з'являються в тому ж документі, що і «Quality Assurance», в той час як Word2Vec показує слова, які можуть зайняти те ж саме або сусіднє місце, як «Quality Assurance» (див.рис. 3.5).

Word2Vec - Window size 10	Word2VecF - Global Context
selenium	bug
test	programming
manual	jenkins
testautomation	jira
testing	assurance
testmethod	unix
...	...

Рисунок 3.5 – Приклад слів в документі Word2VecF

Коли подібні слова запитуються, наприклад, для "Quality Assurance", Word2VecF має тенденцію отримувати слова, які з'являються в документі, пов'язаному з QA, як-от jenkins і програмування. В той час як модель Word2Vec отримує слова, які можуть з’являтися в тому самому місці у вікні контексту, як-от тестування і selenium.

3.1.3 Doc2Vec

Обидва описані раніше методи (Word2Vec і Word2VecF) створюють вектори слів, які усереднюються для представлення вектора документа. Для того, щоб безпосередньо представляти документи, Ле та Міколов ввели Doc2Vec, де кожен абзац має унікальний вектор у матриці D, а кожне слово має свій власний вектор у матриці W (така ж архітектура локального контексту, що й Word2Vec). Ці вектори усереднюються та об’єднуються, щоб передбачити наступне слово в контексті даного абзацу.

Вектор абзацу використовується лише для слів одного абзацу. Його можна представити як інше слово в контексті, яке фіксується для всіх речень і вікон в абзаці.

Тому він зберігає (або запам’ятовує) тему параграфа. Саме звідси назва архітектури отримала назву «Розподілена пам’ять», як показано на рисунку 3.6.

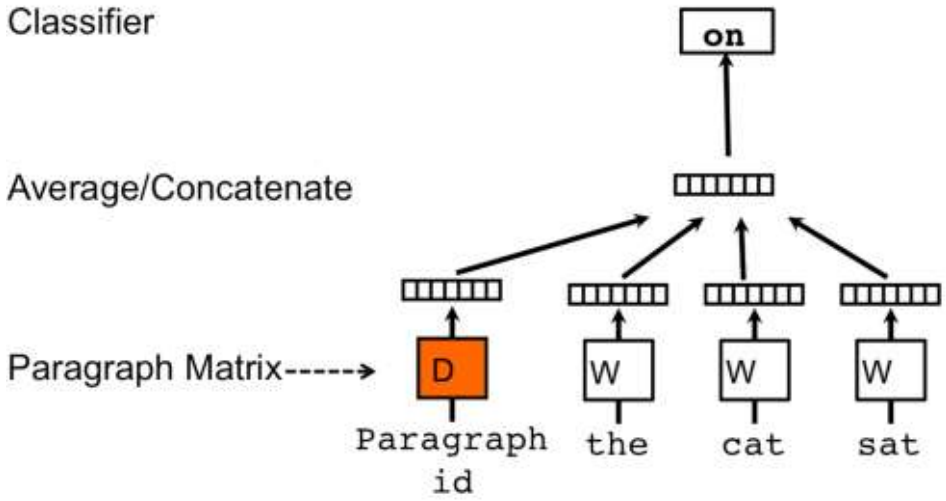

Рисунок 3.6 – Архітектура моделі «Розподілена пам’ять»

Модель розподіленої пам’яті Doc2Vec використовує вектор абзацу разом із словами локального контексту, щоб передбачити слово w(t), вона також діє як пам’ять теми абзацу.

Розподілена торба слів (distributed Bag-of-Words, DBOW), однак, ігнорує розмір контекстного вікна та обчислення векторів слів, оскільки це змушує модель передбачати випадкові вибірки слів у документі з урахуванням вектора документа, як показано на рисунку 3.7. DBOW оновлює лише вектор абзацу, тому йому потрібно менше пам’яті, оскільки він ігнорує вектори слів.

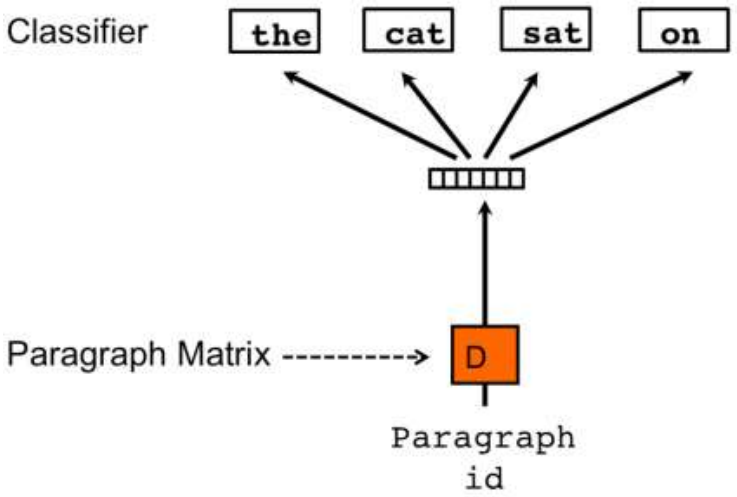

Рисунок 3.7 – DBOW намагається передбачити випадкову вибірку набору слів в абзаці з урахуванням вектора абзацу

Недоліком Doc2Vec є його обчислювальна складність, оскільки матриця абзаців збільшується в розмірі по відношенню до кількості документів, через відсутність підходу від'ємної вибірки при оновленні ваг документа, всі документи залучені в процес навчання. Більше того, щоб створити вектори для невидимих документів, документи потрібно додати до корпусу, а навчання моделі потрібно перезапустити з самого початку, що неможливо в галузі, оскільки воно погано масштабується.

3.2 Подібність та показники оцінки

У цьому розділі ми розглянемо TF-IDF як вагову функцію, яку можна інтегрувати із середньозваженими семантичними моделями. Крім того, будуть описані дві метрики подібності, косинус і відстані Хеллінгера, щоб пояснити їх значення для семантичних і тематичних моделей. Нарешті, ми обговоремо основний показник оцінки в роботі, показник F1, який використовується для порівняння продуктивності моделі під час експериментів.

3.2.1 TF-IDF

TF-IDF - це трохи складніший підхід до формування вектора ознак. Як і в попередньому методі, вважається, що якщо слово часто зустрічається в тексті, і воно не є стоп-словом, то, швидше за все, воно є важливим [6]. Але є й друга тонкість. Якщо слово зустрічається в інших документах рідше, ніж у цьому, то й у цьому випадку, швидше за все, воно важливе для тексту. За цим словом можна відрізнити цей текст від інших.

Якщо врахувати описані вище міркування, то вийде підхід TF-IDF. Значення ознаки для слова w та тексту x обчислюється за такою формулою:

$$TF - IDF(x, w) = n_{dw} log\ (\frac{l}{n_w})$$

Ця формула складається із двох частин. n_{dw} - частка входжень слова w в документ d. Якщо слово часто зустрічається в тексті, воно важливо для нього, і значення ознаки буде вище. Другий множник називається IDF (inverse document frequency, зворотна документна частота), це відношення l, загальної кількості документів, до n_w, кількості документів у вибірці, у яких слово w зустрічається хоча б раз. Якщо це відношення велике, слово рідко зустрічається в інших документах, значення ознаки збільшуватиметься. Якщо слово зустрічається у кожному тексті, то значення ознаки буде нульовим.

3.2.2 BM25

Okapi BM25 або BM25 (де BM позначає Best Matching) також є однією зі стандартних методологій ранжування та зважування термінів, що використовуються для визначення релевантності документа для даного пошукового терміну або запиту. BM25 означає «Найкращий збіг 25» та його 25-ту ітерацію налаштування розрахунку релевантності. Це базується на ймовірнісній структурі пошуку, розробленій Джонсом Робертсоном та іншими в 1970-80-х роках [7].

Імовірнісна модель підкидає релевантність як задачу ймовірності і відображає ймовірність того, що користувач знайде результат, релевантний запиту. BM25 базується на сумці слів пошукової моделі. Подібно до TF-IDF, ця модель також базується на просторовому розподілі частот термінів. BM25 ранги документів засновані на термінах, що з'являються в кожному документі, а не на близькості в документі [8]. Формула BM25 для терміну ваги також базується на схемі зважування TF-IDF, але з варіаціями в тому, як обчислюються терміни TF і IDF. Формула для зважування терміну в BM25:

$$wt_{w,d} = \sum_{i}^{n} IDF(q_i) * \frac{f(q_i, D) * (k1 + 1)}{f(q_i, D) + k1 * (1 - b + b * \frac{fieldLen}{avgFieldLen})}$$

де q_i - термін запиту q в позиції i;

IDF(q_i) - інверсна частота документа q_i.

Хоча термін називається IDF, але є різниця в методі обчислення. Цей термін IDF карає термін, який є більш поширеним і зменшує загальну вартість терміну ваги для відповідного терміну. IDF обчислюється як:

$$IDF(q_i) = \ln(1 + \frac{(docCount - f(q_i) + 0.5)}{f(q_i) + 0.5})$$

де *docCount* - це загальна кількість документів;

f(q_i) - це кількість документів, що має заданий термін q_i.

У статті «BM25 The Next Generation of Lucene Relevance» порівнюється класичний IDF та BM25 IDF, що можна побачити на рисунку 3.8 [9].

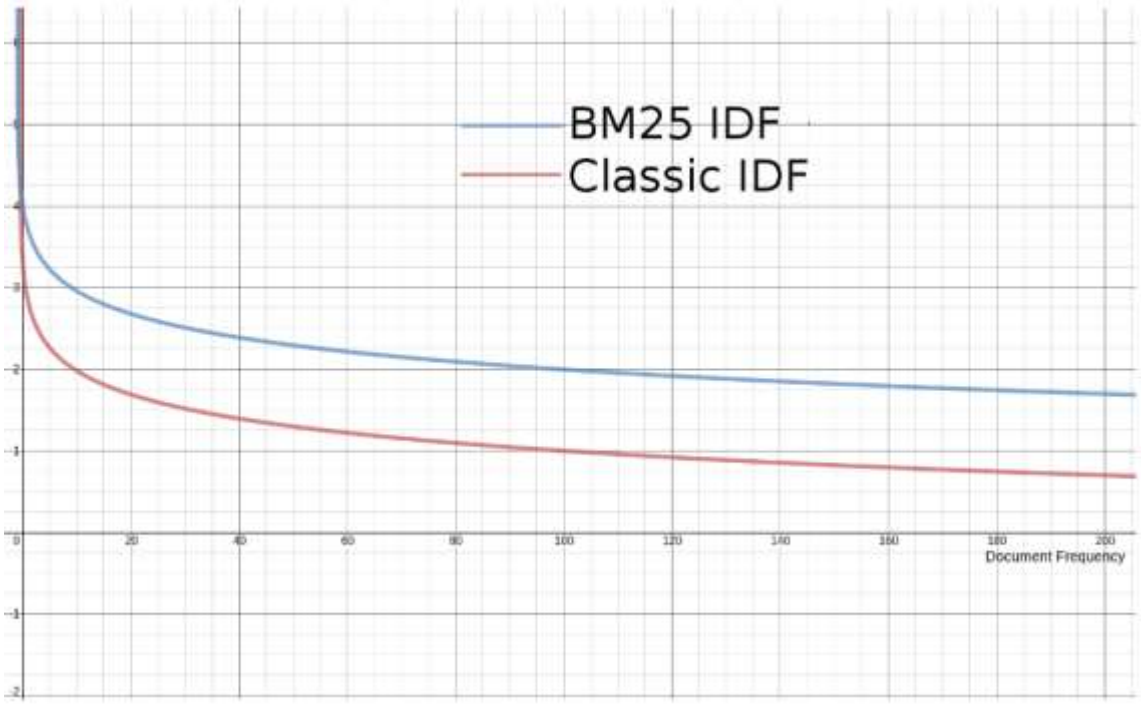

Рисунок 3.8 – Порівняння класичного IDF та BM25 IDF

Вищевказана формула IDF має наступний недолік. Для слів, що входять до більш ніж половини документів з колекції, значення IDF є негативним. Таким чином, за наявності будь-яких двох майже ідентичних документів, в одному з яких є слово, а в іншому - ні, другий може отримати більшу оцінку. Основною зміною обох формул є додаткова 1. У класичному IDF є шанси повернути від'ємні значення, це пом'якшується використанням додаткової функції 1 в логах.

3.2.3 Universal Sentence Encoder (USE)

Universal Reprection Encoder [10] - це не конкретно схема для зважування термінів, а більш просунута модель, яка використовується для векторизації тексту високої розмірності і, отже, використовується в таких задачах, як класифікація тексту, семантика, кластеризація, пошук інформації, рекомендаційні системи та інші пов'язані з текстом завдання.

USE навчений і оптимізований для більш ніж простої векторизації тексту на основі термінів і працює на рівні речення, фраз або малого абзацу, вбудовуючись в щільний вектор. Модель USE навчається на різних текстах природної мови та різних джерелах даних для використання в декількох задачах обробки природної мови. Вхід є англійським текстом змінної довжини, а вихід - вектором розмірності 512. Модель USE тренується з глибоким усередненням мережі (DAN) кодувальника. Деякі з класичних застосувань універсального кодування речень є семантичною подібністю та класифікацією тексту.

Семантична подібність: семантична подібність показує рівень, на який два різних тексти позначають одне і те ж значення. Це в основному використовується, щоб отримати кращі результати в задачах пошуку інформації і всіх пов'язаних завдань, заснованих на розумінні природної мови.

Класифікація тексту: текстові класифікаційні моделі показують кращу точність та ефективність при навчанні з розширеною моделлю векторизації, такою як USE. Користувальницькі бінарні текстові класифікатори можуть бути навчені з моделлю USE, щоб добре виконувати найрізноманітніші завдання класифікації з меншою кількістю помічених даних.

Інтуїція моделі USE полягає в тому, щоб вбудувати речення в щільні вектори, які позначають передачу навчальних завдань до завдань NLP. Ці моделі є ефективними і призводять до хорошої продуктивності та точності в завданнях

навчання передачі для природних мов. Дослідження порівняли різні моделі, що реалізують векторизацію слів та речень для компромісів між точністю та обчислювальними ресурсами. Порівняння вивчає складність моделі, доступність даних та загальну продуктивність завдання. Порівняльний аналіз показує, що навчання передачі разом з векторизацією речення виконує значно краще, ніж модель векторизації слова, а також з меншою кількістю контрольованих навчальних даних.

3.2.4 Показники схожості

Як тільки документи представлені у вигляді векторів, міри подібності використовуються для обчислення подібності між двома заданими документами. Завдання, такі як кластеризація або пошук найближчих сусідів, можуть бути використані для виконання за допомогою значення подібності між документами в корпусі. Існує багато вимірів подібності, таких як коефіцієнт Жаккара, евклідове відхилення, косинусні відстані тощо.

Косинус подібності (cosine similarity). Одним із стандартних показників подібності документа і слів є косинус подібності. Він показує, як два вектори пов’язані в термінах векторного кута замість величини. Для двох векторів документів v1 і v2 їх значення косинусної подібності обчислюється формулою:

$$SIM_C(v_1, v_2) = \frac{v_1 . v_2}{|v_1| * |v_2|}$$

де v1 і v2 - m-вимірні вектори, які представляють документ, а результатом є невід'ємна оцінка подібності між -1 і 1, де оцінка подібності 1 означає, що v1 і v2 ідентичні.

3.2.5 F-міра (F-measure)

Під час пошуку інформації продуктивність різних моделей можна порівнювати за допомогою таких показників, як влучність, повнота та F-міра. Влучність показує частку вилучених документів, які є релевантними. У той час як повнота вимірює частку відповідних документів, отриманих з усіх відповідних документів у наборі даних.

Оскільки влучність і повнота мають зворотну кореляцію, переважно для оптимізації використовується комбінація. Повнота отримає 100%, якщо буде отримано всі відповідні документи плюс нескінченно багато невідповідних документів. Влучність досягне 100%, якщо моделі вдалося отримати, наприклад, лише 5 відповідних документів із нескінченної кількості відповідних.

F-міра (або F1-оцінка) визначається як середнє гармонічне між влучністю та повнотою. Середнє гармонічне використовується, тому що ми маємо справу з співвідношеннями та відсотками, «оскільки воно прагне до найменшого числа, мінімізуючи вплив великих викидів і максимізуючи вплив малих». Іншими словами, і повнота, і влучність повинні мати високе значення, щоб мати високий бал F1, завдяки чому гармонійне середнє краще підходить, ніж арифметичне.

3.2.6 Порівняння методів аналізу тексту

У 2013 році компанія Google представила алгоритм word2vec, який поступово став набувати популярності. Теорія, що за ним приховується, показала свою ефективність. Word2vec ґрунтується на контексті слова, щоб визначити вектор слова. Чим більше прикладів використання тих чи інших слів, тим менша ймовірність

взяття аномального контексту, оскільки розглядаються найчастіше використовувані контексти [11].

У стандартній моделі CBOW, розглянутій вище, ми прогнозуємо ймовірність слів і оптимізуємо їх. Функцією для оптимізації (мінімізації у разі) служить дивергенція Кульбака-Лейблера. Word2vec не стояв на місці, його модернізували. Виявилося, що оптимізувати цю формулу є досить складно. Насамперед через те, що розраховується за допомогою softmax по всьому словнику. Варто зазначити, що багато слів разом не зустрічаються, тому більша частина обчислень у softmax є надмірною. Було запропоновано обхідний шлях, який отримав назву Negative Sampling. Суть цього підходу полягає в тому, що ми максимізуємо ймовірність зустрічі для потрібного слова у типовому контексті (тому, який часто зустрічається в нашому корпусі) і одночасно мінімізуємо ймовірність зустрічі у нетиповому контексті (тому, який рідко чи взагалі не зустрічається). Згідно з дослідженнями, якість роботи моделі з Negative Sampling можна порівняти з якістю роботи моделі с softmax, причому модель з Negative Sampling навчається швидше, ніж модель з Negative Sampling [12].

Числове представлення текстових документів є складним завданням у машинному навчанні. Така вистава може використовуватися для багатьох цілей, наприклад: пошук документів, веб-пошук, фільтрація спаму, моделювання тем і т.д. Трохи згодом з'явилася реалізація word2vec для документів – doc2vec [13].

Doc2vec – дуже гарний підхід. Він простий у використанні та дає хороші результати. Згідно з дослідженнями якості різних моделей, Word2vec показує кращі результати пошуку схожих документів на великому корпусі, ніж LSA, PLSA, LDA. Векторне уявлення дозволяє визначити прихований (латентний) зміст текстів на основі слів, що зустрічаються один з одним [14].

Перевага doc2vec над LSA проявляється під час використання великого корпусу документів. В одній із статей було зроблено порівняння можливостей Skip-gram та LSA для вивчення точних вкладень слів у невеликі текстові корпуси. Щоб

зробити це, була перевірена здатність моделей пророкувати семантичні категорії (такі як напої, країни, інструменти чи одяг) у вкладених підвиборках корпусу середнього розміру. Виявлено, що результати якості Word2vec перевершують LSA, коли моделі навчаються з наборами даних середнього розміру (10 мільйонів слів). Однак, коли розмір корпусу зменшується, продуктивність Word2vec різко знижується, тому LSA стає більш підходящим інструментом. Вважається, що зниження продуктивності Word2vec на невеликих корпусах засноване на тому факті, що моделі, засновані на передбаченнях, потребують великої кількості навчальних даних, щоб відповідати їх великій кількості параметрів [15].

Глибокі нейронні мережі зараз є одним із найпопулярніших методів інтелектуального аналізу даних. Майже у всіх предметних областях вони показують якісніші результати, ніж інші методи машинного навчання.

4 НОРМОВАНИЙ ЗА ЧАСОМ ПІДХІД ЗВАЖУВАННЯ ТЕРМІНІВ

4.1 Розрахунок віку терміну

Часовий фактор (також можна назвати віком терміну) являється роком походження слова (може знайти в різних джерелах) і частоти терміну в документі, обчислюється як відношення обох, що дають таку метрику, як документи на рік. Для заданого терміну *w* у документі $d \in D$, де *D* - це корпус документа *D* з розміром *N*, вік терміну обчислюється формулою:

$$t_{w,D} = \log(df_{w,D}/(y_{diff} + 1))$$

де y_{diff} обчислюється формулою:

$$y_{diff} = y_{current} - y_{origin}$$

де y_{origin}це рік першого вживання слова і $y_{current}$- це нинішній рік [4].

Ми не розглядаємо рік публікації статті і беремо поточний рік з певних причин. Оскільки термін може бути використаний протягом декількох років, так що обчислення віку буде змінюватися кожен раз, і вибір будь-якого з них буде складним викликом. Крім того, це може призвести до певних аномалій та неоднозначностей у віковому обчисленні. Щоб спростити цю логіку і мати однаковий вік для терміну в корпусі, ми розглянемо поточний рік, який є останнім роком статей, опублікованих в корпусі. Ми беремо логарифм термінів для нормалізації значення, так як це може призвести до великого числа виходячи з розміру корпусу. Крім того, ми беремо абсолютне значення журналу, так що ми не маємо від'ємних значень ваги. 1 додається у знаменник, щоб уникнути поділу на нульову помилку у випадку, якщо рік походження збігається з поточним роком.

y_{origin} може бути знайдено з декількох джерел. Розглянемо декілька прикладів та формулювання фактора віку терміну:

а) якщо розробляється рекомендаційна робота, рік походження може бути отриманий з року першої появи в дослідницькій роботі. А поточний рік може стати останнім роком статті, присутньої в корпусі.

б) у випадку веб-персоналізованої пошукової системи, час першого пошуку конкретного терміна може бути використаний для формування часового контексту. Різниця від першого року використання до поточного року може бути використана як різниця року.

в) для більш загальних екземплярів і для того, щоб зробити обчислення більш надійним, рік походження можна простежити до етимології і рік першої появи можна отримати. Це також може бути використано для отримання інших контекстів слова на основі походження слова та використання для поліпшення результатів пошуку та рекомендацій, однак ці аспекти виходять за рамки цього дослідження.

В цій роботі я використовувала рік походження, як загальний і отримувала його з etymonline.com, щоб отримати етимологію та отримати рік першого виникнення. Було використано набір даних з новинами, які використовують терміни, що пояснюють широкий спектр часового контексту. Я також перевірила той же алгоритм на дослідницькому паперовому наборі даних і веб-наборі даних пошуку відповіді. А також, щоб зберегти алгоритм більш надійним і загальним, тестування етимології, здається, є вдалим способом перевірити гіпотезу. Було введено термін віковий фактор в трьох термінах схем зважування, тобто TF-IDF, BM25 і USE, все це пояснюється в наступних розділах.

4.2 Нормований за часом TF-IDF(tTF-IDF)

TF-IDF - це класична методологія зважування термінів, яка зважує терміни на основі двох факторів, частоти термінів та оберненої частоти документа. Обидва ці фактори засновані на просторовому розподілі термінів, що є частотою їх виникнення в корпусі. Розраховується по формулі:

$$wt_{w,d} = tf_{w,d} * log(\frac{N}{df_{w,d}})$$

Реальна проблема в цьому формулюванні полягає в тому, що TF-IDF припускає, що частотна дистрибуція залишається постійною при зміні часу. За короткий проміжок часу, це вірно, однак, протягом більш тривалих періодів часу, це припущення не правдиве. Щоб подолати проблему, що терміни мають часові розподіли частоти, а не просто просторовий розподіл, які не враховуються при використанні TF-IDF, я використовую алгоритм з нормалізованим часом TF-IDF. На рисунку 4.1 наведено алгоритм для tTF-IDF.

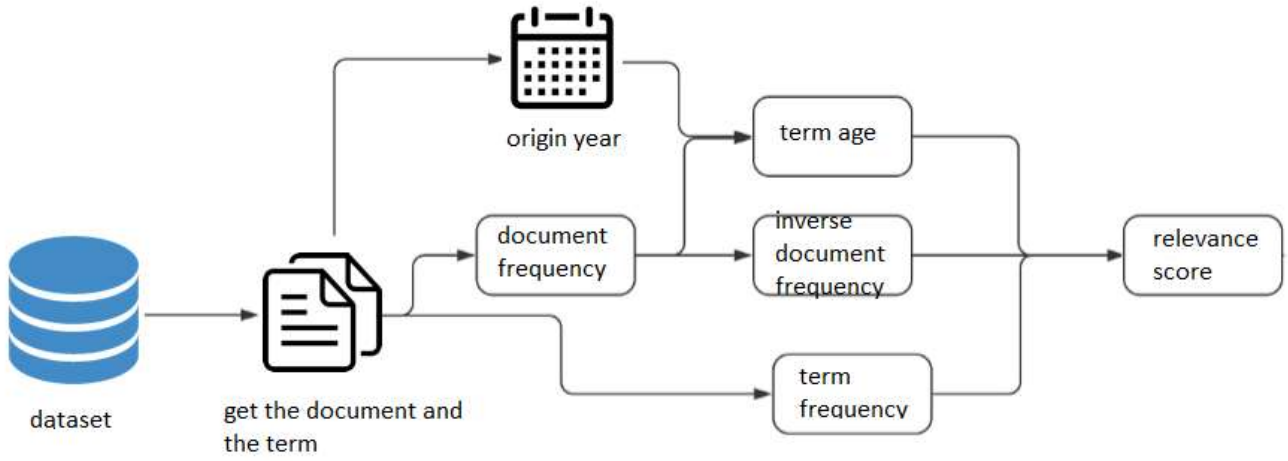

Рисунок 4.1 – Алгоритм tTF-IDF

Формула для обчислень:

$$wt_{w,d} = t_{w,D} * tf_{w,d} * log(\frac{N}{df_{w,d}})$$

З введенням фактору віка терміну, ми маємо намір включити тимчасовий контекст термінів для поліпшення результатів пошуку.

4.3 Нормований за часом BM25(tBM-25)

BM25 - це ще одна стандартна методологія зважування термінів, яка також базується на обчисленнях частоти терміна та зворотної частоти документа, але логіка обчислення цих змін. IDF обчислюється так само, як і в TF-IDF, але з додаванням 1 у знаменнику для подолання ділення на нульову помилку. Для обчислення терміну частоти розглядаються деякі додаткові метрики на основі терміна, такі як довжина поля і середня довжина поля. Деякі інші константи, такі як k1 і b, також розглядаються в цій формулі, які використовуються для нормалізації терміна оцінки зважування. Рахується за формулою:

$$wt_{w,d} = \sum_{i}^{n} IDF(q_i) * \frac{f(q_i, D) * (k1 + 1)}{f(q_i, D) + k1 * (1 - b + b * \frac{fieldLen}{avgFieldLen})}$$

Хоча це долає проблеми, виявлені в TF-IDF і виконується краще в деяких випадках, але питання не розглядати різноманітний контекст терміну в різних аспектах все ще залишається таким самим. Навіть додані метрики, такі як довжина поля і константи k1 і b, базуються на просторовому розподілі термінів. Формулюючи

tTF-IDF, аналогічно, ми розглядаємо формулу для tBM25, що використовує вік терміну в обчислення ваги терміну. Розрахунок за формулою:

$$wt_{w,d} = \sum_{i}^{n} t_{w,D} * IDF(q_i) * \frac{f(q_i, D) * (k1 + 1)}{f(q_i, D) + k1 * (1 - b + b * \frac{fieldLen}{avgFieldLen})}$$

І цей час - нормалізований фактор віку терміну також працює над тим, щоб розглянути тимчасовий аспект термінів.

4.4 Нормований за часом USE(tUSE)

Універсальний кодувальник речень (USE) не є конкретно терміном методології зважування, а більш просунутою схемою. Модель векторизації тексту та речень перетворюють даний текст у вектори високої розмірності, а потім подібність між цими векторами в основному обчислюється за допомогою функції подібності косинусів. Ця подібність косинусів знаходить кут між заданими векторами і чим менше значення кута, тим більш схожими є тексти. Ця формула для знаходження косинусної подібності між високомірними векторами наступна:

$$\cos\theta = \frac{\sum_1^n \vec{a}_i b_i}{\sqrt{\sum_1^n \vec{a}_i^2}\sqrt{\sum_1^n \vec{b}_i^2}}$$

Основною перевагою використання моделі USE в задачі пошуку тексту є те, що вона розглядає семантичний контекст тексту разом з розподілом частот. Це добре працює в більшості текстових пошуків і дає досить хороші результати в наших

наборах даних також. Однак, питання, яке ми вирішуємо в цьому дослідженні, все ще залишається відсутнім в логіці формул. Маємо формулу:

$$\cos\theta = t_{w,D} * \frac{\sum_1^n \vec{a}_i b_i}{\sqrt{\sum_1^n \vec{a}_i^2}\sqrt{\sum_1^n \vec{b}_i^2}}$$

Подібно до інших двох методологій зважування термінів, було помножено фактор віку терміну з функцією регулярної косинусної подібності для текстової подібності та отримано версію tUSE.

4.5 Аналіз алгоритмів

Як ми бачимо, вік терміну залежить не тільки від року походження, але і від кількості документів, в яких цей термін знаходиться (частота документів). Це також можна розглядати як додатковий фактор до значення IDF, але з іншим значенням. Давайте розглянемо вплив частоти документа і різниці часу на значення ваги терміну. Припустимо, що термін є новим і виникає в достатній кількості документів, тоді значення віку терміну, $t_{w,D}$ буде великим і, отже, вага терміну буде великою. І контрастний випадок, якщо термін є відносно новим, і знаходиться в меншій кількості документів, це очікувана поведінка, що термін буде важити менше через його меншу значущість. Аналогічним чином, якщо термін використовується протягом багатьох років і знаходиться в багатьох документах, це відносно знизить значення фактора часу, надаючи йому низьку важливість.

Застереження, яке необхідно прийняти під час реалізації цього алгоритму, полягає в тому, щоб перевірити, чи частіше трапляються невідповідні терміни, які нормалізуються за допомогою IDF, не повинні збільшуватися. Наприклад,

виникнення стоп-слів. Їх знаходять максимальну кількість разів в будь-якому корпусі, і це немає значення в загальних схемах пошуку. Про це можна подбати під час обчислення значення y_{origin}, і такі терміни можуть бути проігноровані, тому вони не збільшують вагу терміну на основі не релевантних термінів. Або якщо стоп-слова не мають великого значення навіть у семантичній логіці системи пошуку, вони можуть бути видалені в самому першому індексі. Це збереже обчислення, а також зменшить шанси на небажану частковість у результатах.

5 МЕТОДОЛОГІЯ

5.1 Опис наборів даних

5.1.1 Trec News датасет

Ця колекція містить 608,180 статей і блогів з новинами за період з січня 2012 по серпень 2017. З метою перевірки нашої гіпотези ми використовуємо зразок ~ 21000 документів з приблизно ~ 2400 релевантними документами. Однак це було зроблено лише для індексу на основі часу через масштабованість та обмеження ресурсів. І вік терміну все ще обчислюється з урахуванням всього корпусу і не впливає на алгоритмічну логіку. Набір даних знаходиться у форматі файлу JSON, що містить наступні поля: id, URL, назва, автор, дата публікації, текст статті, розбитий на абзаци, підпис і біографія автора. Було створено три індекси на сервері Elasticsearch, один з яких містить весь цей набір даних, як присутній в даному файлі JSON, з регулярним відображенням. Другий індекс містить оновлений параметр ваги терміна та містить нормалізовану за часом вагу терміна, який буде використаний пізніше для розрахунку оцінки релевантності та пошуку. І третій індекс, що містить поле векторизації тексту USE разом з параметром ваги терміна.

Для тестування результатів і налаштування основної істини для системи було використано TREC-2018 новини для задачі фонового зв'язування (background linking task). Це завдання полягає в отриманні фонових новинних статей, що надають контекст для основної статті. Читач повинен отримати всю відповідну інформацію про шукану тему з найкращого джерела [16]. Наприклад, якщо заголовок теми пошуку: «Як великі міста США та транзитні системи реагують на напади в Брюсселі», то відповідною статтею буде «Транспортні агентства через США підвищують безпеку після паризьких атак» або «Ісламська держава заявляє про відповідальність за напади в Брюсселі».

Надано 50 таких запитів з наступними полями: номер документа, ID, URL. Це використовується як файл вхідного запиту до системи пошуку інформації. Наведено інший файл даних, який містить найбільш релевантні результати для кожного заданого запиту у вхідному документі. На рисунку 5.1 показано уявлення про набір даних та його структуру.


```
"ID": "Zad5721e-3642-11e1-afdf-67906fc95149",
"URL": "https://www.washingtonpost.com/national/health-science/first-ever-hybrid-shark-discovered
    -off-australia/2012/01/03/gIQAPy00YP_story.html",
"title": "First-ever hybrid shark discovered off Australia",
"author":"Juliet Eilperin",
"published_date": 1325632195000,
"text": " Health & Science First-ever hybrid shark discovered off Australia By Juliet Eilperin
    Scientists have identified the first-ever hybrid <a href="http://ww.demonfishbook.com">shark</a>
    off the coast of Australia, a discovery that suggests some shark species may respond to changing
    ocean conditions by interbreeding with one another. A team of 10 Australian researchers identified
    multiple generations of sharks that arose from mating between the common blacktip shark
    (Carcharhinus limbatus) and the Australian blacktip (Carcharhinus tilstoni), which is smaller and
    lives in warmer waters than its global counterpart. "To find a wild hybrid animal is unusual,
```


Рисунок 5.1 – Структура Trec News датасету

Ці результати перевірені людиною і можуть розглядатися як основна істина для нашого експерименту. Цей файл містить наступні поля: номер документа, ID документа, значення користі, яке визначається як 2^r, де r є шкалою релевантності від 0 до 4. В таблиці 5.1 наведено відповідні значення релевантності золотого стандарту.

Таблиця 5.1 – Золотий стандарт TREC News

query number	id	Relevance (2^r)
321	00f57310e5c8ec7833d6756ba637332e	16
321	02ae7136-006d-11e3-9711-3708310f6f4d	2
321	049739753ce2e539d8dc2165daaf6aa6	4

Номер документа, згаданий у цьому файлі, збігається з номером, вказаним у файлі вхідних запитів. Отже, цей файл встановлює основну істинність для даного

набору запитів, документи, які мають відношення до кожного запиту та масштабу релевантності для цих документів.

5.1.2 Web AP датасет

Ця колекція містить 8027 статей з Інтернету, які є відповідями на 82 запити TREC. Набір даних є підмножиною треку TREC Terabyte 2004 Gov2 набору даних разом з темами тестового запиту, вказаними в треку. Цей набір даних очищений і містить відповідні уривки для 82 тестових запитів, вказаних в основній істині. Довжина середнього проходу в цьому наборі даних становить 45 слів. На рисунку 5.2 показано вигляд набору даних та його структуру.


```
"docum": "GX008-71-13615464-705",
"target_qid": "705",
"orginal_doc_num": "GX008-71-13615464",
"passage": "Press Room FROM THE OFFICE OF PUBLIC AFFAIRS June 27, 2003 JS-516 Assistant Secretary
    for International Affairs Randal Quarles United States Policy Enforcement in the Global Economv
    Remarks to the Marriott School of Management, Brigham Young University, Provo, Utah, June 27, 2003
    Thank you, Dean Hill. Its a pleasure to be back home in Utah and to be able to share with you today
    my thoughts on the global economy and US engagement in international economic affairs. The Role of
    Economics in Foreign Policy From the outset of this Administration, President Bush has stressed
    that his foreign policy is based on three essential, interdependent building blocks --military,
    political, and economic--with economics by no means in third place. In fact, the first National
    Security Presidential Directive named the Secretary of the Treasury for the first time - as a
    formal member of the National Security Council, together with the Secretaries of Defense and State.
    This formal inclusion of economics into foreign policy has certainly resulted in an elevation of
    economic issues. But it has also led to a government inter-agency mechanism--from the cabinet, to
    their deputies, to technical staff--that allows for increased coordination between economic and
    military/political issues. This increased coordination has been evident in many areas, from the
    Reconstruction of Iraq to a heightened emphasis on business-like input-output performance measures
    used in policy evaluation, even in areas where quantitative information is difficult to find or
    measure. Overall Economic Policy Goals So, from the outset of the Administration, international
    economic policy has been central to foreign policy. Two goals have guided the international
```


Рисунок 5.2 – Структура Web AP датасету

Набір даних знаходиться у форматі файлу JSON, що містить наступні поля: унікальний id документа, id цільового питання та прохід. І таблиця 5.2 показує знімок результатів золотого стандарту релевантності.

Таблиця 5.2 – Релевантність золотого стандарту Web AP

Number	docnum	relevance
701	GX268-35-11839875	1
701	GX262-28-10569245	2
701	GX238-57-4348848	3
701	GX231-53-10990040	4

Було створено три індекси з цього набору даних, один з яких містить дані, наведені у файлі JSON. Другий індекс, що містить оновлений параметр ваги терміна як нормалізований фактор часу. І третій індекс з щільними векторами для векторизації USE разом з нормалізованим фактором часу. Нам даються основні результати істинності для даного набору запитів 82 як набір з 50 найбільш релевантних результатів. Результати подаються як id питання, номер документа та релевантність від 1 до 50.

5.1.3 CiteULike датасет

Цей набір даних зібраний з CiteULike і Google Scholar і містить 17013 документів. CiteULike був сервісом, який дозволяв людям створювати свою особисту колекцію аркушів або паперу.

Набір даних подається у форматі файлу CSV з наступними полями: docid (унікальний id документа), title (заголовок нижнього регістру), raw title (оригінальна назва дослідницької роботи) та abstract. Було створено три індекси з цього набору даних, один з яких містить дані, як це є, другий містить параметр нормалізованої ваги

терміна часу та третій індекс, що містить щільні вектори на основі кодування USE разом з метрикою віку терміна. На рисунку 5.3 показано вигляд набору даних та його структуру.


```
"docid": "56",
"title": "functional genomic hypothesis generation and experimentation by a robot scientist",
"citeulike id" : "292",
"raw title": "Functional genomic hypothesis generation and experimentation by a robot scientist",
"abstract": "The question of whether it is possible to automate the scientific process is of both
    great theoretical interest, 2 and increasing practical importance because, in many scientific
    areas, data are being generated much faster than they can be effectively analysed. We describe a
    physically implemented robotic system that applies techniques from artificial intelligence3, 4, 5,
    6, 7, 8 to carry out cycles of scientific experimentation. The system automatically originates
    hypotheses to explain observations, devises experiments to test these hypotheses, physically runs
    the experiments using a laboratory robot, interprets the results to falsify hypotheses inconsistent
    with the data, and then repeats the cycle. Here we apply the system to the determination of gene
    function using deletion mutants of yeast (Saccharomyces cerevisiae) and auxotrophic growth
    experiments9. We built and tested a detailed logical model (involving genes, proteins and
    metabolites) of the aromatic amino acid synthesis pathway. In biological experiments that
    automatically reconstruct parts of this model, we show that an intelligent experiment selection
    strategy is competitive with human performance and significantly outperforms, with a cost decrease
    of 3-fold and 100-fold (respectively), both cheapest and random-experiment selection.
```


Рисунок 5.3 – Структура CiteULike датасету

Нам надається інший файл у цьому наборі даних, який містить посилання на статті для кожного документа. Це служить нашою умовою проблеми для розробки рекомендаційної системи дослідницької роботи. Вхідними даними для цього буде назва статті, а очікуваними результатами буде стаття з посиланням на цю статтю. Оскільки посилання наведені для кожної статті, і це зробить оцінку упередженою і складною для агрегації результатів. Для того, щоб виправити це питання, я випадково відібрала 116 тестових тем, що мають рівно 10 цитат, які будуть використані як наша основна правда.

5.2 Показники оцінки

5.2.1 Точність@10 (Precison@10)

Точність визначається як кількість відповідних результатів у наборі отриманих результатів. Як правило, точність обчислюється для верхнього K числа результатів, заданих як P@K. Це оцінює кількість відповідних результатів, отриманих у верхній K кількості результатів. Було протестовано систему на заданому наборі з 50 запитів (для TREC News), 82 запитів (для Web AP) і 116 запитів (для CiteUlike), тому обчислюємо середню точність, яка займає середнє значення точності, розрахованої для всіх заданих запитів. Ми взяли значення K як 10, тому ми перевіряємо релевантність результатів топ-10 у нашому сценарії. Ми розрахували значення точності для топ-10 отриманих результатів на заданому наборі вхідних запитів і беремо середнє значення результатів для порівняння.

5.2.2 Повнота (Recall) (тільки для TREC News)

Для розрахунку повноти були розглянуті запити, що мають менше 100 результатів, як у випадку новин TREC. Повнота розраховується як кількість відповідного витягнутого документа, поділеного на кількість відповідних документів, присутніх в індексі. Для інших наборів даних значення точності і повноти залишається незмінним, оскільки кількість відповідних результатів є фіксованою.

5.2.3 F-міра (F1 Score) (тільки для TREC New)

Оскільки F1 оцінка використовує як точність, так і значення повноти, тому для обчислення оцінки точності, було використано ті ж результати повноти і використовували фіксований знаменник як кількість отриманих результатів (100 в нашому випадку). Формула, що використовується для F1 оцінки:

$$F1 = 2 * (Precision * Recall)/(Precision + Recall)$$

5.2.4 NDCG@10 міра

Це показник якості рейтингу. Він базується на припущенні, що більш релевантні документи розміщені вище у списку результатів. При обчисленні DCG, більш високий релевантний результат, розміщений на нижній позиції, карається значенням журналу пропорційно позиції. DCG обчислюється в конкретній позиції рангу p, який задано наступною формулою:

$$DCG_p = \sum i = 1 prel_i / \log_2(i+1)$$

де, rel_i - оцінка релевантності документа на позиції *i*.

NDCG обчислюється шляхом розгляду DCG ідеального порядку разом із значеннями:

$$nDCG_p = DCG_p / IDCG_p$$

де $IDCG_p$ є ідеальними значеннями DCG на позиції p. Це розглядається з основної істини або з перевірених людиною результатів, які вказують рейтинг результатів, яким вони віддають перевагу для релевантності. Було взято значення на позиції 10 і розрахувано nDCG для кращих 10 результатів. Ми отримали результати для заданого набору запитів, тому обчислюємо середнє nDCG для обох наборів алгоритмів і порівнюємо результати.

6 РЕАЛІЗАЦІЯ

6.1 Індексування даних

Індексація даних є першим кроком у впровадженні системи інформаційного пошуку і має вирішальне значення для ефективного пошуку. Метод грубої сили пошуку документів - це лінійне сканування всієї бази даних. Щоб уникнути лінійного сканування, ми будемо робити індексацію даних. Для ефективної індексації та пошуку, як запит, так і дані повинні бути в індексному вигляді [17]. Індексація даних відбувається в чотири етапи: по-перше, це збір даних і управління корпусом даних, по-друге - розбір списку документів, по-третє - токенізація термінів в документі і створення логічної структури, а останній крок - створення індексу даних. Цей процес індексації показаний на рисунку 6.1.

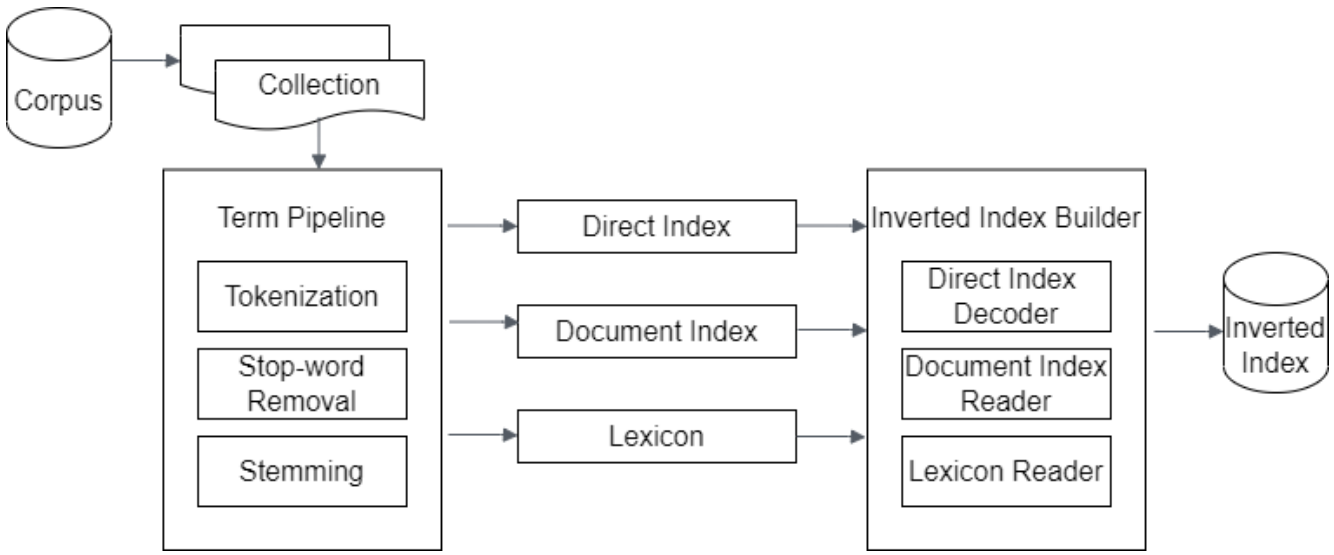

Рисунок 6.1 – Процес індексування даних

Що стосується інформаційної системи пошуку та рекомендацій, індексація даних має наступні переваги:

а) покращує продуктивність при отриманні відповідних документів;

б) оптимізує швидкість інформаційної системи пошуку;

в) допомагає зберегти дані унікальними і скинути дублікати;

г) текстові індекси полегшують пошук великих рядкових значень, в порівнянні з традиційними підходами до бази даних.

Індексація даних є однією з найважливіших частин стратегії реалізації. В інформаційних системах пошуку або рекомендації, затримка для отримання результатів, ефективність результатів, і релевантність результатів для даної проблеми відіграють основну частину рішення. Усіма завданнями можна легко керувати, якщо дані індексуються ефективно.

6.1.1 Архітектури Elasticsearch та Apache Lucene

Elasticsearch побудована за допомогою Apache Lucene і використовує індекси Lucene. Поняття інвертованого індексу використовується при формуванні індексу в Elasticsearch. Інвертований індекс створюється шляхом відображення термінів до документів та їх розташування в документах у вигляді словника [18]. Словники сортуються, тому простіше знайти терміни, а отже і відповідні документи. Протилежно цьому є передній індекс, де терміни пов'язані з конкретними документами, що досить повільно в пошукових операціях. Отже, інвертований індекс використовується при формуванні індексу в Elasticsearch.

Для операції пошуку з декількома термінами, це робиться шляхом пошуку всіх термінів і їх відповідних випадків, і остаточні результати складаються за деякою формою сукупності цих результатів. Основна логіка операції пошуку полягає в тому, щоб знайти відповідні терміни, потім відповідні входження, позиції і, отже, документи. Рисунок 6.2 показує архітектуру Elasticsearch.

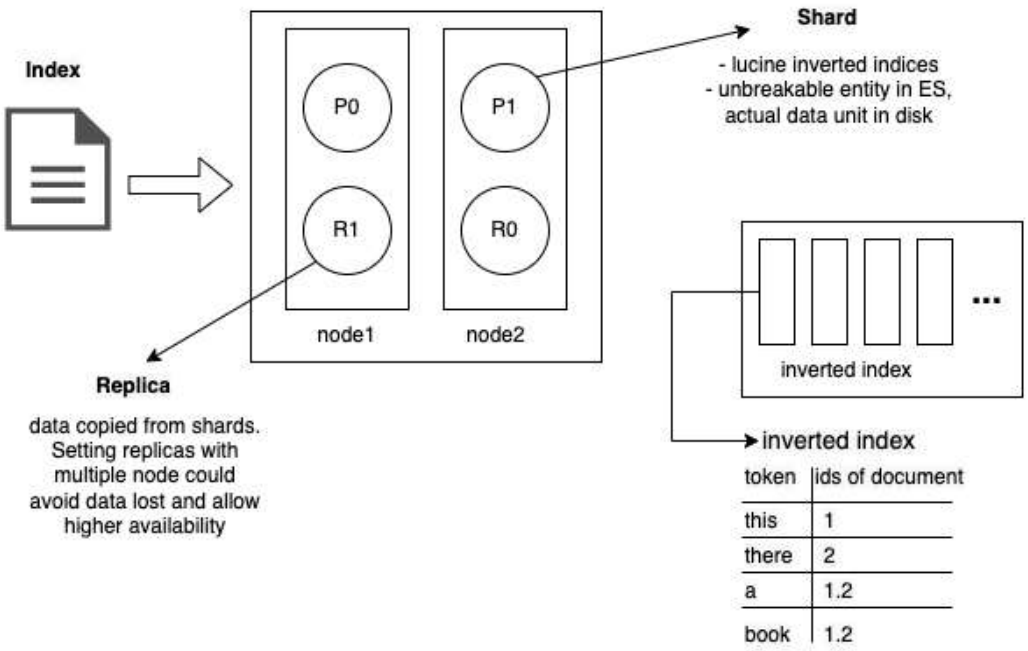

Рисунок 6.2 – Архітектура Elasticsearch

Індекс Elasticsearch створюється з одним або декількома шардами (shards), які можуть мати нуль або більше реплік. Кожен шард є в індивідуальному індексі Lucene. Таким чином, індекс Elasticsearch складається з декількох індексів Lucene, які складаються з декількох сегментів індексу. Операція пошуку в індексі Elasticsearch виконується на всіх шардах та сегментах. Вони об'єднуються, щоб дати кінцевий результат. Подібний процес відбувається під час пошуку декількох індексів. Також можна вважати, що пошук двох індексів з одним шардом збігається з пошуком одного індексу з двома шардами, в обох випадках пошук двох Lucene індексів. Elasticsearch також дуже гнучкий і надає варіанти для оптимізації результатів, надаючи варіанти для визначення індексів Elasticsearch, шардів (і реплік), на які спрямований пошук. Крім того, є параметри для визначення шаблонів індексів, псевдонімів індексів, маршрутів документа та пошуку, розділення даних та стратегій потоку даних.

6.1.2 Реалізація за допомогою Elasticsearch

Ми використовували Python і Elasticsearch для управління цим завданням індексації даних. Оскільки Elasticsearch розроблений з метою текстової системи пошуку, тому основне завдання ефективної індексації полягає у його внутрішній обробці. Скрипт python піклується про збір даних, попередню обробку та передачу даних до сервера у правильному форматі, а частина індексації, що залишилася, виконується Elasticsearch. Малюнок 6.2 пояснює архітектуру процесу індексації даних. Після того, як дані індексуються, вони відображаються як пари ключових значень, а також отримуються таким же чином, як і на малюнку 5.3.

Перший індекс для всіх трьох наборів даних створюється з тією ж структурою відображення, що і у файлах JSON. Регулярна індексація в Elasticsearch зберігає термін зі стандартними метриками. Важливо зауважити, що всі метрики, такі як частота терміну, частота документа, довжина документа, середня довжина документа тощо, обчислюються під час індексації та зберігаються як метадані індексу. Це також відіграє значну роль у пошуку документів, оскільки ваги термінів просто отримуються з метаданих, а не обчислюються на даний момент.

6.2 Розрахунок віку терміну та індексація

Було створено три індекси для кожного набору даних. По-перше, стандартний індекс з усіма стандартними метриками, як описано в останньому розділі. Для другого індексу ми додаємо нормалізований час для фактори ваги терміну або вік терміну, який слід зберігати разом з іншими метриками. Для розрахунку віку терміну виконуються наступні кроки:

а) ітерувати по ID, що зберігаються в першому індексі і отримати основний текст статті;

б) розглянути текст статті документа і розділити його на окремі слова;

в) для кожного слова, отримати рік походження, який є роком першого входження цього слова з etymonline.com. На рисунку 6.3 показано результати пошуку з сайту і «1300» вважається роком походження терміну «university», який ми шукали;

г) розрахувати різницю в кількості років від року першої згадки до поточного року. Було взято базовий рік для корпусу 2017 (TREC News), 2015 (Web AP) і 2019 (CiteULike), так як це рік останніх публікацій. Це було зроблено для рівномірного терміну зважування по всьому корпусу і отримання метричної одиниці такої як документи на рік;

д) тепер ми отримуємо частоту документа, $df_{w,D}$ для певного терміну *w* і розділяємо частоту документа з різницею в році. Оскільки значення відносно велике, я взяла його логарифмічне значення, щоб нормалізувати цей вираз. Крім того, щоб уникнути поділу на нульову помилку у випадках, коли термін щойно придуманий, і рік походження збігається з поточним роком, я додала один до знаменника;

е) нарешті, абсолютне значення обчислення, виконаного на останньому кроці, є параметром віку терміну і обчислюється для кожного терміну в тексті статті.

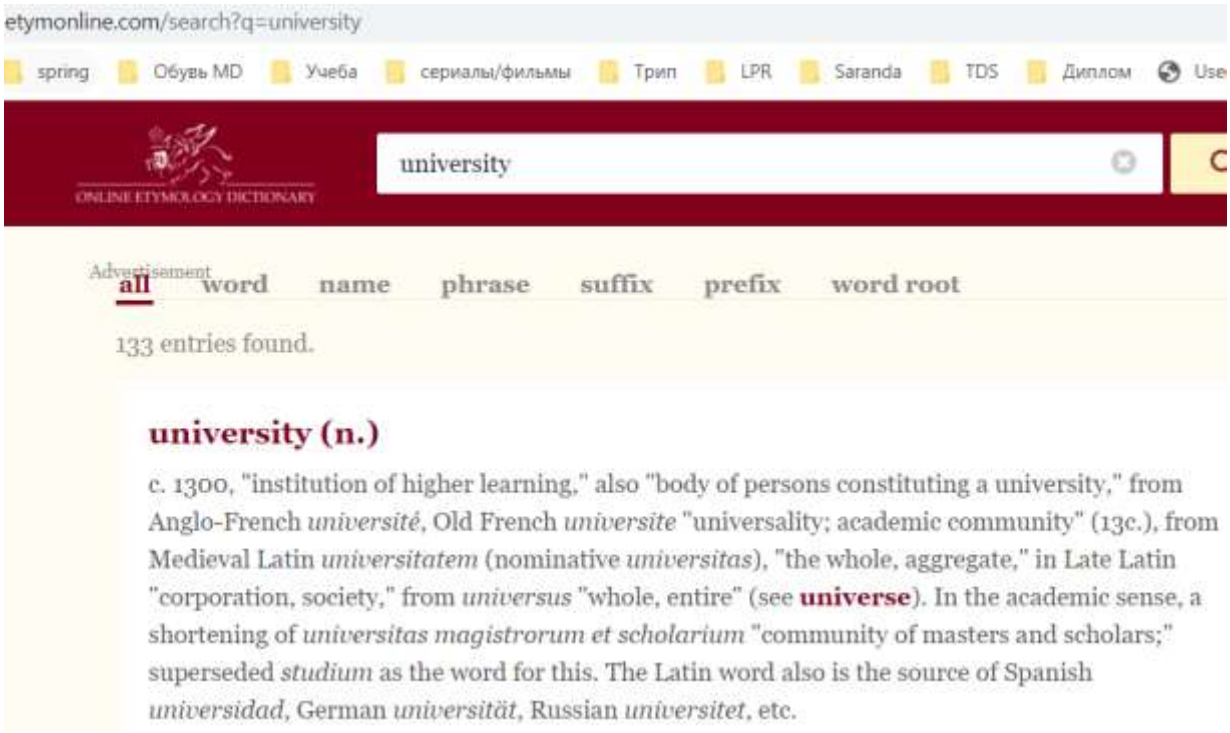

Рисунок 6.3 – Результати пошука з Etymonline

Як тільки ми обчислили вік терміну, ми додаємо це к термінам, використовуючи роздільник. А новостворений текст статті індексується у вигляді індексу корисного навантаження. Оскільки для параметр віку терміну не має стандартного стовпця, ми зберігаємо його як корисне навантаження в індексі, а пізніше використовуємо його в обчисленні оцінки пошуку та релевантності. Оновлений текст документа виглядає так: «*Americans—5.432 to—0.000 rate—5.466 their—1.080 ideology—3.303 on—0.000 a—0.000 scale—3.455 of—0.000 1—0.000 to—0.000 5—0.000.*». На рисунку 6.4 показано структуру відображення індексу корисного навантаження, виділено векторні та аналізаторні поля, що показують доданий аналізатор для корисного навантаження.

```
"mappings": {
    "properties": {
        "author": {...},
        "authorBio" {...},
        "caption" {...},
        "id" {...},
        "published_date" {...},
        "source" {...},
        "text": {
            "type": "text",
            "term vector": "with_positions_offsets_payloads",
            "analyzer": "positionPayloadAnalyzer"
        },
        "title": {
            "type": "text",
            "fields": {
                "keyword": {
                    "type": "keyword",
                    "ignore_above": 256
                }
            }
        },
        "type" {...},
        "url" {...}
    }
```

Рисунок 6.4 – Структура відображення індексу корисного навантаження

Важливо відзначити, що не кожному терміну надається вага терміну, це відбувається з кількох причин. По-перше, якщо рік походження не може бути простежено для терміну, це зазвичай відбувається для власних імен. По-друге, якщо терміни найчастіше використовуються, такі як статті, прийменники тощо, які мають високе значення частоти документа і в результаті можуть похилити результати пошукового запиту.

6.3 Індексація корисного навантаження на основі USE

Для стандартних схем зважування термінів документи, які отримуються, базуються на найбільшій кількості термінів, що збігаються з вхідним запитом. Однак для поліпшення результатів пошуку, є випадки, коли кількість відповідних термінів менше, але все ж, документ може мати відношення до вхідного запиту з урахуванням

семантичного контексту. Для подолання цього питання і підвищення продуктивності пошукової системи або системи рекомендацій використовуються текстові моделі векторизації, які кодують текст або речення в високорозмірний числовий вектор.

Векторизація слів - це сімейство методів обробки природної мови, спрямованих на відображення семантичного значення в геометричному просторі. Вона реалізується шляхом зв'язування числового вектора з кожним словом у словнику, так що відстань між будь-якими двома векторами охоплює частину семантичних відносин між двома пов'язаними словами. Наприклад, «кокос» і «білий ведмідь» є словами, які семантично досить відрізняються, але векторизація представить їх у вигляді векторів, які будуть розташовані на великій відстані один від одного; але слова «кухня» і «вечеря» є спорідненими словами, тому вони будуть розташовані ближче один до одного [19].

У нашому дослідженні ми використали універсальну модель кодування речень і створили індекс для зберігання щільних векторів, а також вік терміну корисного навантаження. Ми використали попередньо підготовлену модель Tensorflow для формування щільних векторів. На рисунку 6.5 показана високорівнева архітектура для створення індексів та пошукова програма.

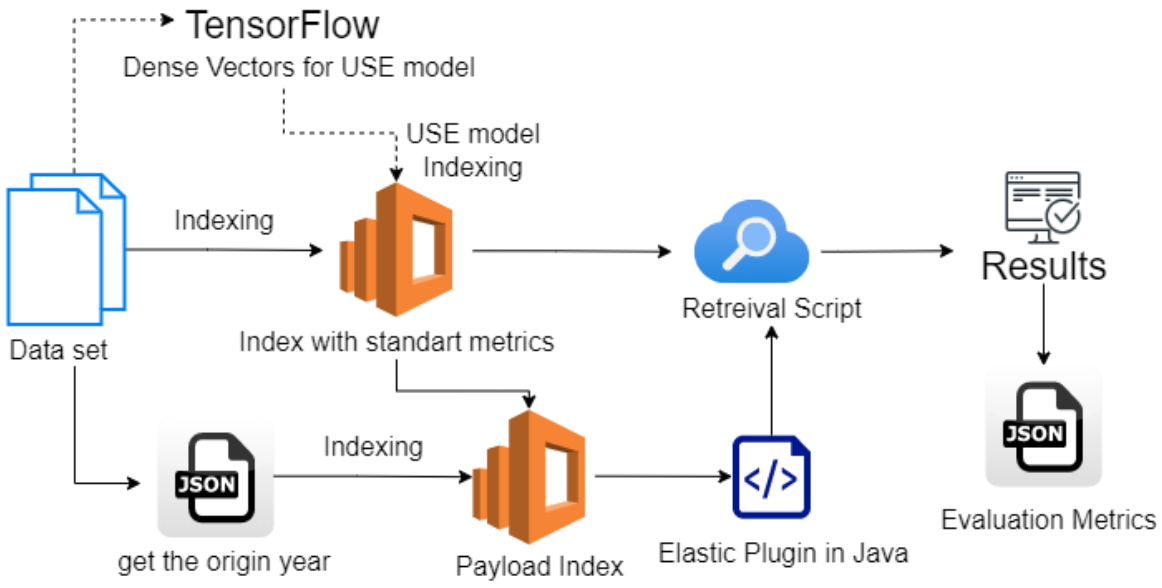

Рисунок 6.5 – Архітектура системи

Ми використовуємо наступний метод для створення індексу на основі USE:

а) ітерувати по ID, що зберігаються в першому індексі, і отримувати основний текст статті;

б) основний текст статті пробігає через попередньо підготовлену модель Tensorflow, створюючи 512 векторів розмірності для основного тексту;

в) параметр віку терміну обчислюється так само, як описано в останньому розділі;

г) така ж модель використовується для обчислення щільного вектора для вхідного запиту під час пошуку тексту.

6.4 Реалізація розрахунку оцінок релевантності

Ми використали Elasticsearch для створення всіх трьох різних індексів для наших наборів даних. Щоб мати рівномірне порівняння з базовими моделями, ми реалізували два різних плагіни. Перший просто отримує стандартні метрики, такі як TF, IDF, довжина документа тощо, щоб обчислити оцінку релевантності та отримати результати. Другий плагін оцінює на основі обчисленого параметра ваги терміна разом зі стандартними метриками. Ці плагіни розроблені на Java і засновані на бібліотеці додатків Apache Lucene. Ці плагіни потім додаються в установку Elasticsearch, а також вказуються під час індексації документів. Нарешті, ми порівнюємо результати, отримані для даного набору запитів від обох цих виконань до наших результатів основної істини.

6.4.1 Реалізація нормованого за часом TF-IDF(tTF-IDF)

Стандартна модель TF-IDF базується лише на двох метриках, частоті термінів та зворотній частоті документа. Ці метрики вже зберігаються в метаданих індексу, просто отримуються безпосередньо під час отримання документів. Формула TF-IDF:

$$wt_{w,d} = tf_{w,d} * log(\frac{N}{df_{w,d}})$$

Отримуємо результати з першого індексу за допомогою цього стандартного метричного розрахунку і формуємо перший набір наших результатів для заданих вхідних запитів. Для другого набору результатів, ми отримуємо результати на основі розширеної версії формули:

$$wt_{w,d} = t_{w,D} * tf_{w,d} * log(\frac{N}{df_{w,d}})$$

Це називається tTF-IDF, де додаткове *t* означає тимчасовий вимір, доданий до існуючої формули. Параметр віка терміну $t_{w,D}$ зберігається як термін корисного навантаження в метаданих документів. Ми використовуємо ці метадані в нашому плагіні, щоб сформулювати і розрахувати оцінки релевантності для отриманих документів [4].

Результати, отримані за допомогою цієї оновленої формули, формують другий набір результатів. Потім ми порівняли перший і другий набір результатів на основі точності, повноти, F1 і NDCG метрик. Детальна інформація наведена в розділі 7.

6.4.2 Реалізація нормованого за часом BM25(tBM25)

BM25 модель також базується на частоті терміна та зворотній частоті документа, але логіка обчислення обох цих показників відрізняється від стандартної моделі TF-IDF. Розрахунок IDF в основному однаковий, лише доданий 1 у знаменнику, щоб уникнути ділення на нуль. Для TF використовуються деякі додаткові метрики довжини поля та середньої довжини поля. Термін обчислення розраховується по формулі:

$$wt_{w,d} = \sum_{i}^{n} IDF(q_i) * \frac{f(q_i, D) * (k1 + 1)}{f(q_i, D) + k1 * (1 - b + b * \frac{fieldLen}{avgFieldLen})}$$

Перший набір результатів для BM25 отримується за допомогою цього стандартного обчислення. Для другого набору результатів, ми отримуємо результати на основі розширеної версії формули:

$$wt_{w,d} = \sum_{i}^{n} t_{w,D} * IDF(q_i) * \frac{f(q_i, D) * (k1 + 1)}{f(q_i, D) + k1 * (1 - b + b * \frac{fieldLen}{avgFieldLen})}$$

Це називається tBM25, де додаткове *t* означає часовий вимір, доданий до існуючої формули. Параметр віку терміну $t_{w,D}$ зберігається як термін корисного навантаження в метаданих документів. Ми використовуємо ці метадані в нашому плагіні, щоб сформулювати це і розрахувати оцінки релевантності для отриманих документів.

6.4.3 Реалізація нормованого за часом USE(tUSE)

Для моделі на основі USE разом з пошуком схожості тексту було використано функцію подібності косинусів. Подібність косинусів вже інтегрована в Elasticsearch, і ми використовуємо те ж саме при отриманні результатів. Для цієї моделі, вхідний текстовий запит спочатку виконується через ту ж попередньо підготовлену модель векторизації речення, яка робить числовий вектор. Тепер, щоб обчислити оцінку релевантності, ми обчислюємо векторну подібність між вхідним вектором і щільним вектором індексованих документів. Подібність косинусів обчислює кут θ між двома заданими векторами. Розраховується по формулі:

$$\cos\theta = \frac{\sum_1^n \vec{a}_i b_i}{\sqrt{\sum_1^n \vec{a}_i^2}\sqrt{\sum_1^n \vec{b}_i^2}}$$

Це формує стандартний перший набір результатів для наших заданих вхідних запитів. Як і в попередніх моделях, ми множимо фактор віку терміну (зберігається в корисному навантаженні метаданих) в стандартному обчисленні, щоб отримати формулу tUSE:

$$\cos\theta = t_{w,D} * \frac{\sum_1^n \vec{a}_i b_i}{\sqrt{\sum_1^n \vec{a}_i^2}\sqrt{\sum_1^n \vec{b}_i^2}}$$

7 РЕЗУЛЬТАТИ І ОБГОВОРЕННЯ

7.1 Порівняння точності, повноти і F-міри

У 2 з 3 алгоритмів модифікація з параметром віку терміна значно покращила продуктивність. При вимірюванні p@10, tTF-IDF перевершив TF-IDF в середньому на 47%, а tUSE перевершив USE в 2 з 3 наборів даних в середньому на 14,3%, але виконав на 50% гірше в третьому наборі даних (див. рис. 7.2 і 7.4). Нормалізований за часом BM25, однак, виконав на 32% гірше BM25 (див. рис. 7.3).

У загальному випадку оцінки пошуку інформації, повнота і F1 міра дають різні результати, ніж точні метрики, якщо задана кількість відповідних результатів не фіксована. У наших наборах даних тільки новини TREC мають різну кількість відповідних результатів, наданих як основна правда. В інших випадках, якщо кількість даних відповідних результатів є постійною, то всі три значення, тобто точність, повнота і F1 оцінки залишаються незмінними. У випадку набору новин TREC, для обчислення результатів повноти та F1, ми отримуємо найкращі 100 результатів для заданих наборів запитів. Для рівномірності в метричному обчисленні ми також обчислюємо точні оцінки. Результати показників наведені на рисунку 7.1.

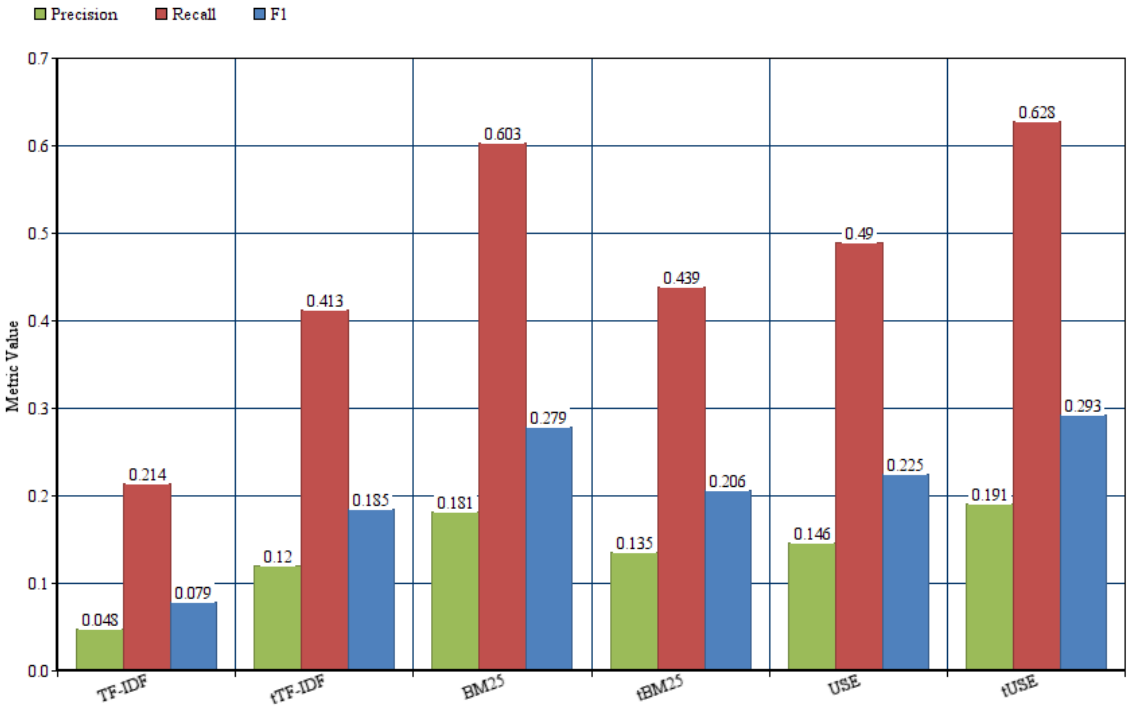

Рисунок 7.1 – Точність, повтота і F-міра для TREC News

Повнота і F1 оцінки також показують значне поліпшення часу нормалізованих моделей над стандартними. Оцінюючи з точки зору повноти, ми бачимо 93% поліпшення моделі tTF- IDF над TF-IDF і 28% поліпшення моделі tUSE над моделлю USE. При цьому tBM25 виконували на 27% гірше стандартної моделі BM25.

7.1.1 tTF-IDF vs TF-IDF

Порівнюючи з точки зору точності та оцінки NDCG, модель tTF-IDF перевершила модель TF-IDF у всіх трьох наборах даних. Глибоко занурившись в результати, для набору новин TREC ми бачимо значне поліпшення 115% в параметрі точності. У той час як для CiteULike tTF-IDF перевершив TF-IDF на 11% і для набору даних WebAP, tTF-IDF перевершив 15% (див. рис. 7.2).

Рисунок 7.2 – Порівняння точності між TF-IDF та tTF-IDF

На основі вивчення корпусу та отриманих результатів для кожного набору даних можна отримати наступні висновки:

а) наявність часового контексту в корпусі новин важніша в порівнянні з іншими текстовими корпусами. Причина такого припущення в тому, що новини висвітлюють події іншої епохи, маючи більш високі шанси показати співвідношення часу в порівнянні з будь-яким іншим текстом;

б) розмір корпусу відрізняється і тому варіація віку терміну є ще одним фактором. Оскільки вік терміну також залежить від частоти документа, тому розмір корпусу також впливає на термін вік опосередковано;

в) вік терміну може не додавати значної релевантності до набору даних CiteULike і WebAP, але значно допомагає ввести вік терміну для поліпшення результатів. І результати можуть бути покращені шляхом тестування з різними факторами нормалізації для віку терміну, а також tTF-IDF.

7.1.2 tBM25 vs BM25

Часова нормалізована модель BM25 не працювала добре для будь-якого з наборів даних. tBM25 модель виконала на 9,4% гірше для набору новин TREC, 44% гірше для набору даних CiteULike і на 42,8% гірше для набору даних Web AP (див. рис. 7.3).

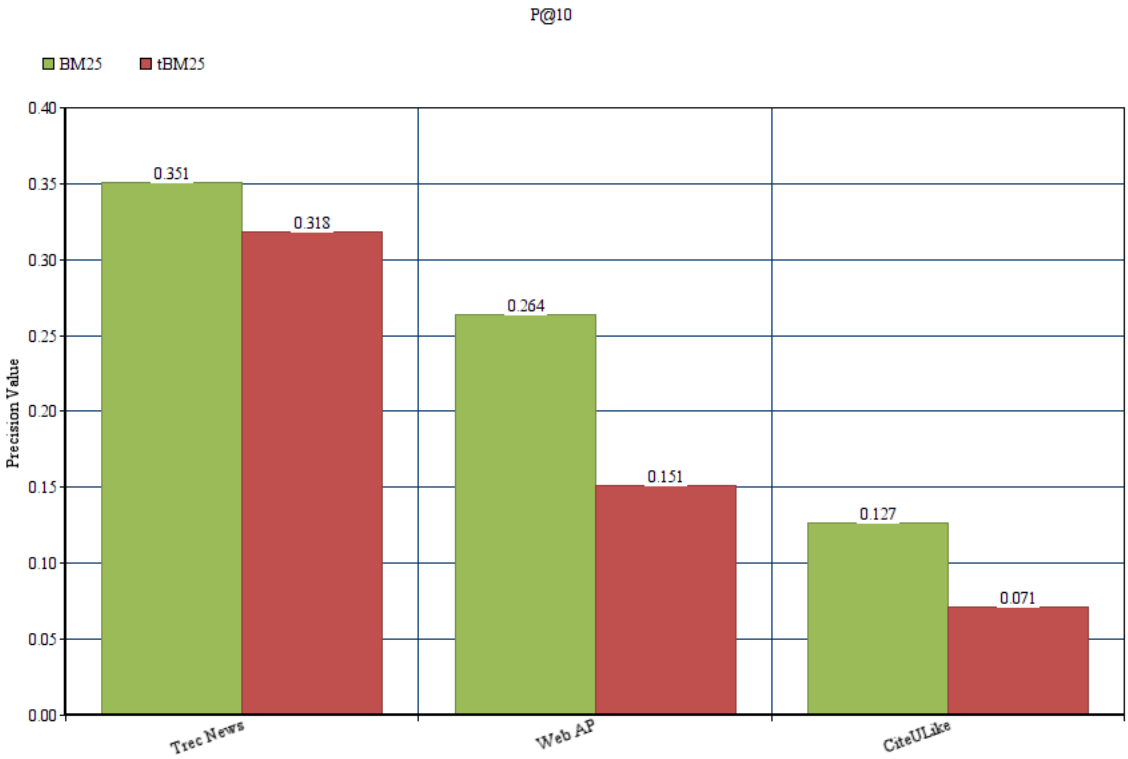

Рисунок 7.3 – Порівняння точності між BM25 та tBM25

На більш детальному аналізі моделі BM25 ми бачимо 27 з 50 запитів у задачі новин TREC, які дають кращі або майже схожі результати у випадку нормалізації BM25 моделі в порівнянні з класичним підходом. Припускаючи, що новинний текст має часовий контекст, ці результати також перспективні і над ними потрібно працювати для імпровізованих результатів у подальшій роботі. На основі отриманих результатів і вивчення можна отримати наступні висновки для tBM25 моделі:

а) BM25 і tBM25 модель працює краще, ніж класична модель TF-IDF і tTF-IDF для всіх випадків;

б) нинішнє формулювання віку терміну не покращує алгоритм пошуку BM25. Різні формулювання віку терміну або фактори нормалізації повинні бути перевірені для перевірки значущості віку терміну в алгоритмі BM2;

в) причиною гіршої продуктивності tBM25 моделі також може бути додаткова метрична довжина поля і середня довжина поля, що використовується в BM25 формулі. Можливо, що поточний метод формулювання віку терміну погано вписується в показники довжини поля;

г) налаштування інших постійних параметрів в BM25 формулі, тобто k1 і b також можна спробувати імпровізувати на tBM25 моделі.

7.1.3 tUSE vs USE

Нормалізована модель універсального кодування речень (tUSE) перевершила модель USE у 2 з 3 наборів даних. Для набору даних TREC News ми спостерігали 18% поліпшення моделі tUSE над стандартним USE. І в CiteULike, ми отримуємо 10.67% поліпшення в нормалізованому часі версії. Хоча він працює на 50% гірше для набору даних Web AP. Порівняння точності між USE та tUSE наведено на рисунку 7.4.

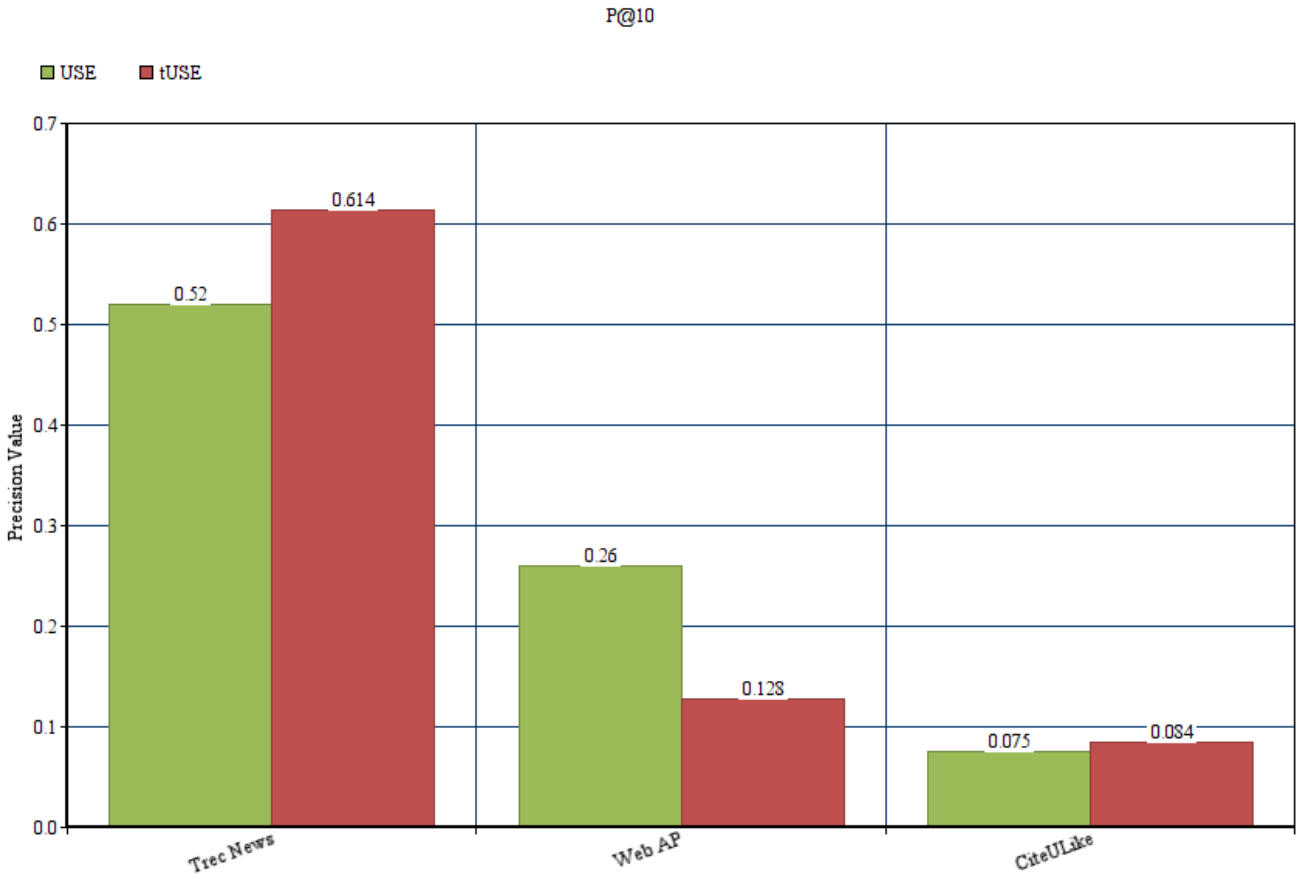

Рисунок 7.4 – Порівняння точності між USE та tUSE

Аналізуючи модель tUSE в наборі даних WebAP, ми бачимо, що вона погано працює проти моделі USE. Однією з можливих причин цього може бути розмір використовуваного корпусу, тобто корпус новин TREC має приблизно 600k документів, тоді як набір даних Web AP має лише 6k документів, що в 100 разів менше. Однак, це висновок на основі отриманих результатів і не був перевірений. Можуть бути й інші можливі причини, такі як розмір документів, розмір використовуваних запитів, кількість правильних іменників у запитах тощо. Або, ймовірно, вік терміну може не бути відповідною метрикою для цього набору даних. Ці можливі причини все ще потрібно проаналізувати, перш ніж підтвердити висновок про ці контрастні результати. На основі вивчення корпусу та отриманих результатів для кожного набору даних можна отримати наступні висновки:

а) з TF-IDF, BM25 та USE модель USE найкраще підходить для отримання результатів для запитів англійською мовою. І, мабуть, модель tUSE працює краще для 2 з трьох наборів даних;

б) в даний час реалізована нормалізація часу погано працює з набором даних Web AP у випадку оцінки релевантності на основі USE;

в) для задачі фонового зв'язку в наборі новин TREC, USE дав p@10 значення 0.52, тоді як tUSE дав значення p@10 - 0.614, що набагато краще в порівнянні з TF-IDF (0.114) і tTF-IDF (0.245). З цього спостереження видно, що пошук на основі семантики є більш перспективним, ніж пошук на основі ключових слів для таких завдань.

7.2 Аналіз оцінки NDCG

NDCG використовується для оцінки рейтингової міри отриманих результатів. Ми використали 10 найкращих результатів для перевірки результатів NDCG. NDCG@10 призводить до аналогічних результатів, як видно для precision@10, проілюстровано в розділі 7.1. Ми обчислили оцінки NDCG тільки для наборів даних TREC News і WebAP, для набору даних CiteULike, NDCG не можна обчислити, оскільки це базується на посиланнях, що використовуються в дослідницькій роботі, і немає рейтингу, вказаного для цитувань. Для новин TREC ми бачимо, що tTF-IDF перевершив TF-IDF на 59%, а tUSE перевершила модель USE на 10,76%. При цьому для tBM25 працює на 18,3% гірше моделі BM25. Ці результати порівняння наведені на рисунку 7.5.

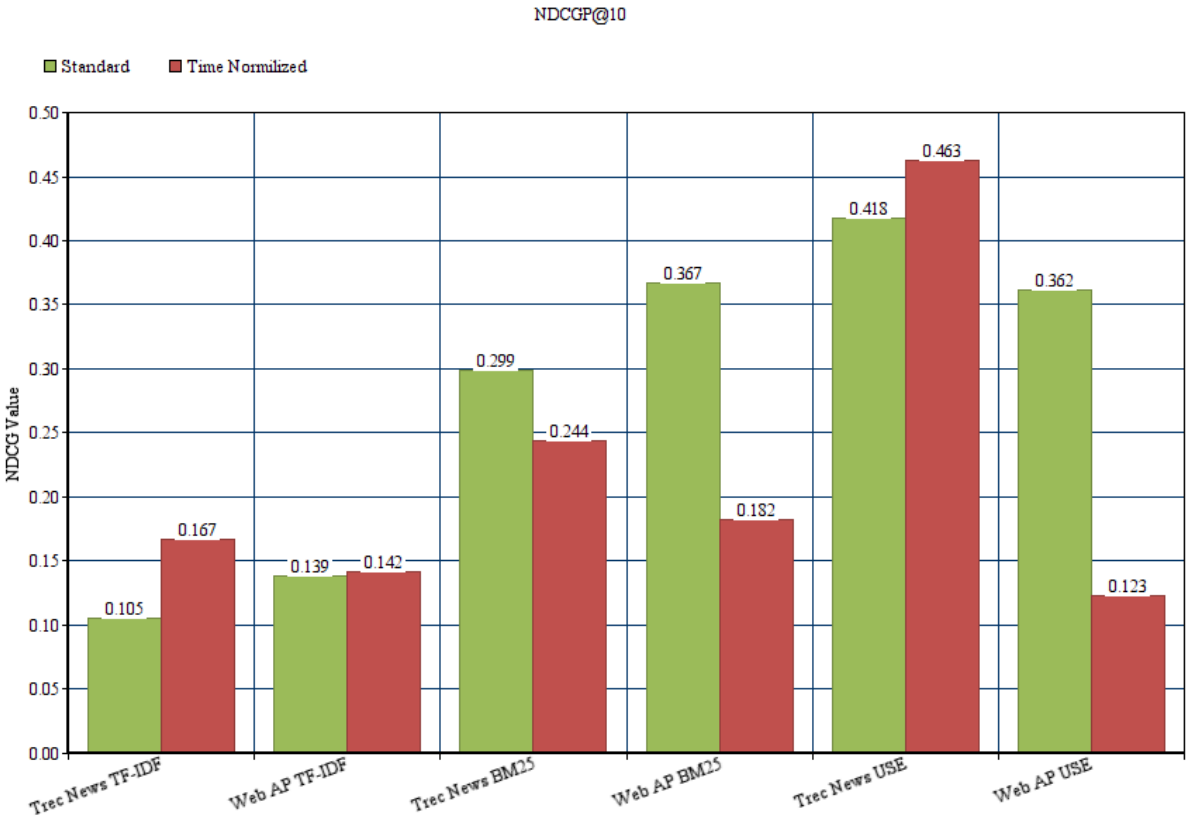

Рисунок 7.5 – Порівняння оцінки NDCG@10

Судячи з цих оцінок NDCG та метрики p@10, очевидно, що нормалізовані моделі часу є перспективними та мають широкий спектр застосувань у системах пошуку інформації та рекомендацій.

7.3 Аналіз розподілу віку термінів

Дивлячись на розподіл слів та роки їх походження, ми спостерігаємо широкий діапазон походження від 1100 року до 2015 року, проілюстрований на рисунку 7.6.

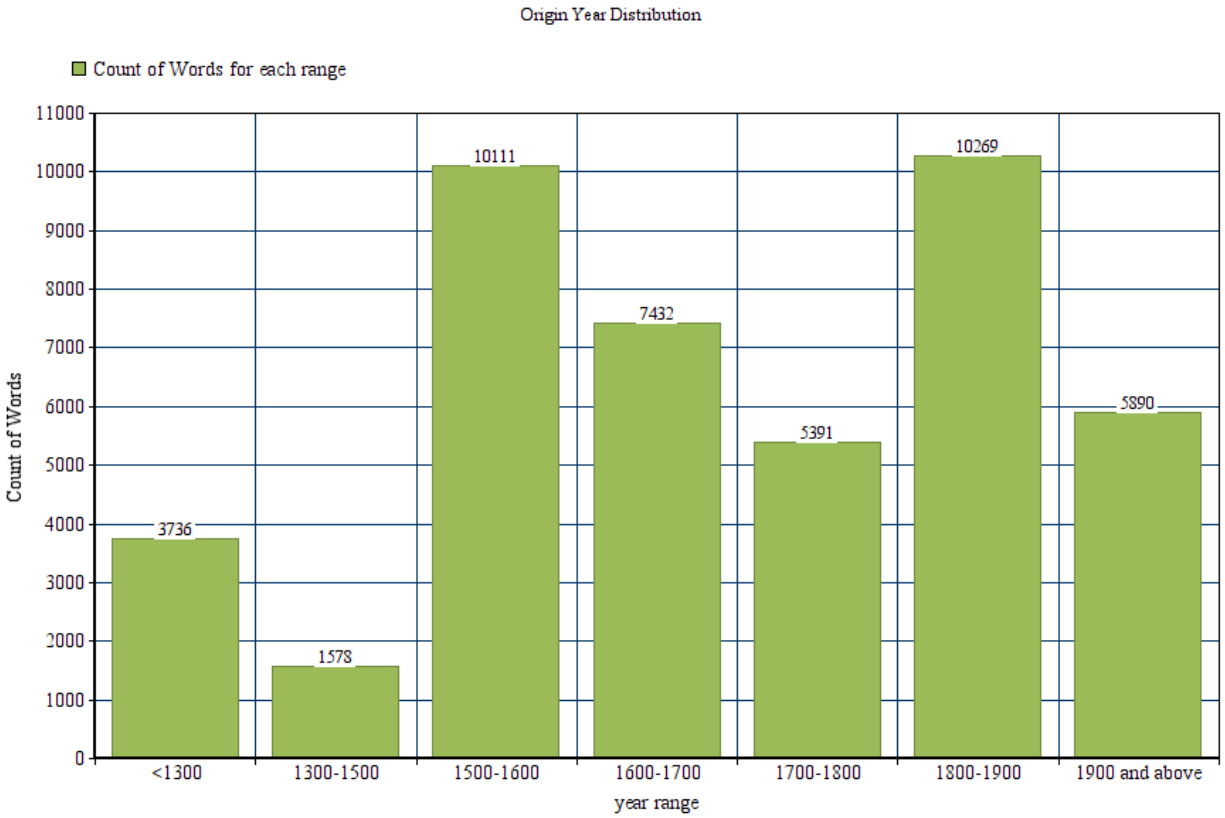

Рисунок 7.6 – Розподіл року походження в роках

Більшість термінів використовуються в діапазоні 1800-1900 року, за якими слідують 1500-1600 діапазону року. Для набору новин TREC ми спостерігали частоту документа від 2 до 550k. Об'єднуючи їх, ми отримуємо термін вік від 0 до 9,71, з середнім значенням 0.9308 і середньою величиною 0.1568. Аналогічно для інших наборів даних, ми спостерігали подібну модель року походження, але зміну частот документа. Отже, варіація віку терміна спостерігається в наборі даних CiteUlike і Web AP. Статистика розподілу віку терміну наведена в таблиці 7.1.

Таблиця 7.1 – Розподілення віку терміну

Min.	Mean	3rd quartile	Max.	Median	Std. dev.	Variance
0	0.9308	1.5198	9.7120	0.1568	1.3363	1.7856

Ще одна цікава статистика для перегляду - це слова, що ми шукали та їх частота. Було проаналізувано шаблон цих термінів у задачі зв'язування новин TREC для знайдених слів. У наборі слів розмір терміну показує частоту використовуваного терміну. Найбільш частими термінами в пошукових запитах є «America», «Cuba», «Ohio», «Change», «deadly», «rise», «new», «death», «cancer», «female», «food», «Earth» і ще деякі терміни.

ВИСНОВКИ

У кваліфікаційній роботі були розглянуті алгоритми зважування термінів, а саме базові алгоритми та модифіковані, що використовують фактор віку термінів. Результати виявили велику значимість часового розподілу разом з існуючим просторовим розподілом термінів. Схема з часовим розподілом використовується на основі дати походження слова і використання в корпусі документа. Цей фактор використовується разом зі стандартними значеннями зважування (такими як TF та IDF) для оцінки релевантності в системі пошуку інформації. Алгоритм нормалізації часу є надійною моделлю, яка може вписуватися в кілька випадків використання. Алгоритм було протестовано на наборі даних з новинами, з запитами, які намагаються знайти зв'язки з документами, пошуковим набором даних веб відповідей та дослідницькими документами, а сама реалізація базується на основі стандартних методологій зважування термінів - TF-IDF, BM25 та USE моделей.

Експерименти, проведені на системі релевантності пошуку, показують, що tTF-IDF і tUSE моделі значно краще працюють, ніж класичні алгоритми TF-IDF і USE з точки зору середньої точності, повноти, F1 і NDCG. Одним з основних недоліків цієї роботи була часова модель BM25, яка не працювала добре для будь-якого з розглянутих наборів даних. Модель BM25 на основі часу повинна бути проаналізована більш детально, щоб отримати краще розуміння обмежень і вивчення введення віку терміну для таких моделей. Можливою причиною може бути те, що новизна терміну не є коректним параметром для BM25 моделі або метод нормалізації погано узгоджується з існуючими метриками, які використовуються, або потрібно перевірити іншу нормалізацію або масштаб. Ці методи потрібно ретельно вивчити, щоб обґрунтувати введення параметру новизни в BM25 метод зважування термінів.

Ще одним важливим експериментом, який може бути проведений в майбутньому, є перевірка різних параметрів нормалізації часу і масштабування часу

для розрахунку віку терміну. Було протестовано тільки один варіант нормалізації на основі отримання віку терміну, що дало значні результати для наших перевірених наборів даних. Вони можуть бути розширені для вивчення різних факторів нормалізації часу і впливу на отримані результати.

Було проведено кваліфікаційне дослідження, яке в основному орієнтоване на завдання пошуку інформації, однак є багато інших додатків для зважування терміну, де новизна терміну може бути застосована. Параметр віку терміну може бути введений в інших задачах, таких як класифікація тексту, моделювання користувачів, рекомендаційні системи та інші пов'язані текстові моделі. Новизна терміну має широку сферу в просторі моделювання користувачів, шляхом побудови моделей користувачів на основі короткострокових контекстів і довгострокових контекстів. Це використовується в подальшему для рекомендацій або поліпшення результатів пошуку. На цьому етапі це дослідження може бути розширено для більш широкого обсягу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Michael Hoey. Lexical Priming: A New Theory of Words and Language. London: Routledge. 2005. ISBN 0-415-32863-2.

2. M. Hilbert, P. Lopez, “The Worlds Technological Capacity to Store, Communicate, and Compute Information,” Science, vol. 332, no. 6025, pp. 60–65, Oct. 2011.

3. Babii A. et al. Neural Natural Language Generation: A Survey on Multilinguality, Multimodality, Controllability and Learning //Journal of Artificial Intelligence Research. – 2022. – Т. 73. – С. 1131-1207.

4. Gruzdo I. et al. Applıcatıon of Paragraphs Vectors Model for Semantıc Text Analysıs //COLINS. – 2020. – С. 283-293.

5. Is word vectors and document vectors in Doc2Vec comparable?

[Електронний ресурс] – URL: https://www.researchgate.net/post/Is_word_vectors_and_document_vectors_in_Doc2Vec_comparable.

6. Term Recency for TF-IDF, BM25 and USE Term Weighting / Divyanshu Marwah [Електронний ресурс] – URL: https://www.scss.tcd.ie/publications/theses/diss/2020/TCD-SCSS-DISSERTATION-2020-071.pdf

7. S. Robertson and H. Zaragoza, The probabilistic relevance framework: BM25 and beyond . Now Publishers Inc, 2009.

8. G. Amati, BM25 , - С. 257–260. Boston, MA: Springer US, 2009.

9. BM25 The Next Generation of Lucene Relevance / OpenSource Connections [Електронний ресурс] – URL: https://opensourceconnections.com/blog/2015/10/16/bm25-the-next-generation-of-lucene-relevation/

10. D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, and C. Tar, “Universal sentence encoder,” arXiv preprint arXiv:1803.11175 , 2018.

11. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “ Distributed representations of words and phrases and their compositionality,” NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems, vol. 2, - C. 3111–3119, 2013.

12. Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” ICML'14 Proceedings of the 31st, vol. 32, - C. 1188-1196, 2014.

13. Чудесный мир Word Embeddings: какие они бывают и зачем нужны? / Блог компании Open Data Science / Хабр. [Електронний ресурс] – URL: https://habr.com/ru/company/ods/blog/329410/

14. M. Campr and K. Ježek, “Comparing Semantic Models for Evaluating Automatic Document Summarization,” Text, Speech, and Dialogue Lecture Notes in Computer Science, - C. 252–260, 2015.

15. E. F. Altszyler, M. F. Sigman, S. F. Ribeiro, and D. F. Slezak, “Comparative study of LSA vs Word2vec embeddings in small corpora: a case study in dreams database,” Conscious Cogn, pp. 178–187, 2017.

16. S. Huang, D. Harman, and I. Soboroff, “The trec news track,” 2018.

17. H. Kaur and V. Gupta, “Indexing process insight and evaluation,” in 2016 International Conference on Inventive Computation Technologies , vol. 3, - C. 1–5, 2016.

18. A. Brasetvik, “Elasticsearch from the bottom up, part 1,” 2013.

19. Назаренко Д. С., Афанасьєва І. В., Голян Н. В. Neural network approach for emotional recognition in text //Біоніка інтелекту. – 2019. – Т. 1. – №. 92. – С. 9-13.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

3. Babii A. et al. Neural Natural Language Generation: A Survey on Multilinguality, Multimodality, Controllability and Learning //Journal of Artificial Intelligence Research. – 2022. – Т. 73. – С. 1131-1207.

4. Gruzdo I. et al. Applıcatıon of Paragraphs Vectors Model for Semantıc Text Analysıs //COLINS. – 2020. – С. 283-293.

19. Назаренко Д. С., Афанасьєва І. В., Голян Н. В. Neural network approach for emotional recognition in text //Біоніка інтелекту. – 2019. – Т. 1. – №. 92. – С. 9-13.