Issue 12, 2022

https://doi.org/10.23939/

УДК 004.9

ІНФОРМАЦІЙНА СИСТЕМА ВИДОБУВАННЯ ІНФОРМАЦІЇ З ВІДКРИТИХ WEB-РЕСУРСІВ

Петро Здебський[1], Андрій Берко[1], Любомир Чирун[2]

[1] Національний університет "Львівська політехніка", кафедра інформаційних систем та мереж, вул. С. Бандери, 12, Львів, Україна
[2] Львівський національний університет імені Івана Франка, кафедра прикладної математики, вул. Університетська, 1, Львів, Україна
E-mail: petro.v.zdebskyi@lpnu.ua, ORCID: 0000-0002-0478-2308
E-mail: Andrii.Y.Berko@lpnu.ua, ORCID: 0000-0003-2892-9519
E-mail: Lyubomyr.Chyrun@lnu.edu.ua, ORCID: 0000-0002- 9448-1751

Мета роботи – створення проєкту інформаційно-довідкової системи знаходження відповіді на запитання на основі найвищого ступеня порівняння за допомогою текстового контенту з відкритих англомовних вебресурсів. Приклади таких запитань: "What is the best book ever?", "What is the most popular IDE for Python". Результатом функціонування інформаційно-довідкової системи є рейтинговий список відповідей на основі частоти появи кожного із варіантів відповідей. До кожного елемента списку також додано числову характеристику ймовірності переваги конкретної відповіді над іншими. На основі цієї метрики ранжують отримані результати. Така інформаційно-довідкова система працює із запитаннями, на які немає однозначної відповіді, цим вона відрізняється від класичних інформаційних систем пошуку відповідей на запитання типу QA-систем. Останні основані на гіпотезі, що є єдина істинна відповідь на запитання. Часто такі системи працюють із загальновідомими фактами. Прикладними запитаннями, на які вони відповідають, можуть бути, наприклад, дата народження відомої людини або кількість населення певної країни. Натомість запропонована інформаційно-довідкова система відповідає на суб'єктивні запитання, наприклад, "Яка найкраща книга у жанрі фентезі?", або "Яка найкраща мова програмування?". Система ґрунтується на популярності тієї чи іншої відповіді. Ключовими словами для формування відповіді на запитання також є власні назви на основі аналізу N-грам.

Ключові слова: інформаційна система; проєкт; QA система; вебзастосунок; пошук контенту; подібність текстових фрагментів; Part-of-speech tagging; N-грама; TF-IDF; TextRank.

Вступ

У зв'язку із бурхливим розвитком інформаційних технологій (ІТ) і безперервним збільшенням обсягів інформації в інтернеті все більшої актуальності набувають питання ефективного інформаційно-інтелектуального пошуку (ІІП) і доступу до релевантного текстового контенту. Найчастіше стандартний ІІП із використанням ключових слів не дає бажаного результату, оскільки такий підхід не враховує мовленнєвих та змістових взаємозв'язків між словами/термами користувацького запиту. Тому нині активно розвиваються ІТ опрацювання природних мов (англ. Natural Language Processing, NLP) й основані на них запитально-відповідальні системи (англ. Question-Answering Systems, QAS) [1–5]. QAS – це інформаційно-інтелектуальна система (ІІС), що є гібридом інформаційних, пошукових, довідкових та інтелектуальних систем, яка використовує інтерфейс природною мовою. Головна мета QAS – отримати конкретну відповідь на поставлене запитання, а не списки документів і/або посилань на джерела із масою непотрібної інформації, які надають більшість сучасних ІІП-систем [6–9]. На QAS подається запит, сформульований природною

мовою, після чого він опрацьовується на основі NLP-методів, і генерується відповідь [10–12]. Як базовий підхід до задачі пошуку відповіді на запитання зазвичай застосовують таку схему: спочатку QAS так чи інакше (наприклад, пошуком за ключовими словами) відбирає документи, що містять контент, пов'язаний із поставленим запитанням, потім фільтрує їх, виділяючи окремі текстові фрагменти, що потенційно містять відповідь, після цьоого з відібраних фрагментів генерує модуль і синтезує відповідь на запитання [13–20]. Community question answering (CQA) сайт/система відповідей на запитання – це відомі інтернет-ресурси, де користувачі взаємодіюють, обмінюючись запитаннями та відповідями на них. CQA – це також задача, яка полягає у знаходженні відповідей на запитання на Q&A-форумах, наприклад, Stack Overflow або Quora. Спільноти відповідей на запитання, такі як Yahoo! Answers або Stack Overflow, належать до групи успішних і популярних програм Web 2.0, які щодня використовують мільйони користувачів, щоб знайти відповідь на складні, суб'єктивні чи залежні від контексту запитання. Для ефективного отримання відповідей CQA-системи повинні оптимально використовувати колективний інтелект усієї онлайн-спільноти, а це неможливо без належної підтримки спільної роботи за допомогою ІТ. Тому CQA став цікавим і перспективним предметом дослідження у галузі інформатики. Незважаючи на збільшення із кожним роком кількості публікацій щодо QAS-систем, сучасні науковці досі оминають дослідження CQA-систем на основі найвищого ступеня порівняння.

Аналіз останніх досліджень та публікацій

Серед відомих аналогів – система передбачення найкращих відповідей у Q&A спільнотах, описана в [21]. Завдання системи – визначити, яка відповідь на конкретне запитання є найкращою. Вона використовує спільноти, у яких користувачі обмінюються запитаннями та відповідями. В роботі розглянуто два підходи для знаходження найкращої відповіді. Перший оснований на контенті відповіді, а другий на характеристиках користувача, який відповідає на запитання (рис. 1).

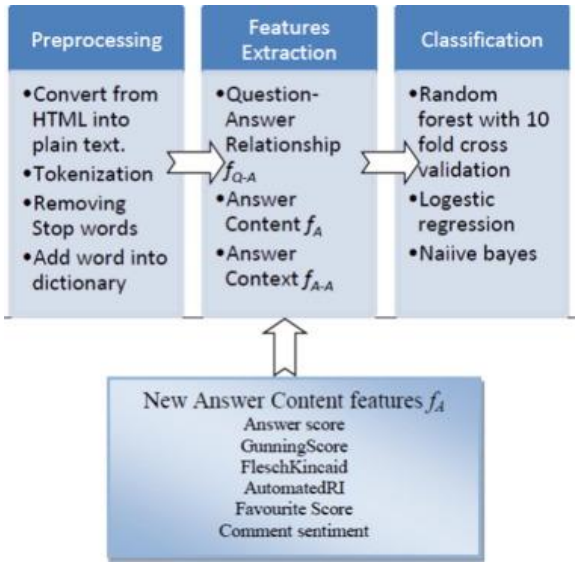

Рис. 1. Модель на основі контенту та нових доданих функцій

У підході, основаному на контенті відповіді, генеруються характеристики контенту (features), які потім передають для класифікації у моделі машинного навчання (англ. Machine Learning, ML), такі як Random Forest, Logistic Regression, Naiive Bayes. У підході, що ґрунтується на характеристиках користувача, використовують евристики активності, рівня впевненості, експертності користувача (рис. 2) тощо. Найкращою вважають гібридну модель як поєднання обох підходів.

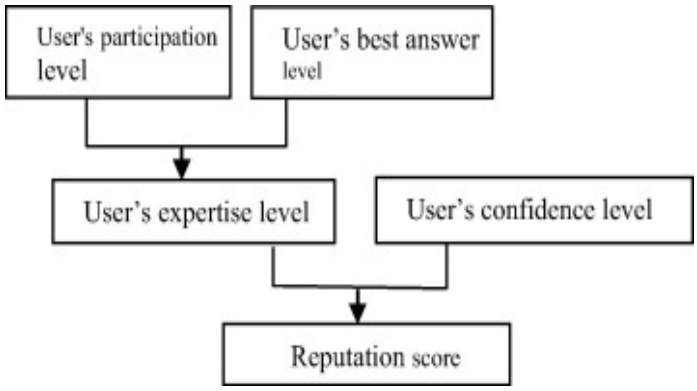

Рис. 2. Оцінка репутації користувача

Ще одним відомим подібним аналогом знаходження найкращої відповіді є система, описана в [22]. Недоліком цієї системи порівняно із системою, запропонованою у цій роботі, є те, що вона рейтингує відповіді, які можуть бути великими за обсягом, а не виводить відсортований список знайдених результатів. Перевагою розглянутої системи є можливість працювати із широким спектром запитань, порівняно із системою, яка знаходить відповіді лише на запитання з найвищими ступенями порівняння. Проте завдання нашої системи – знайти відповіді саме на конкретний тип запитань, а не на будь-які типи запитання.

Нашу задачу також можна розглядати як QA-задачу. Для такого типу задач часто використовують ML-моделі на основі архітектури Transformer [23]. Така система використовує модель BERT. Завдяки моделі BERT досягнуто найвищого рівня продуктивності в низці задач розуміння природної мови. Причини цього досі ще не є достатньо зрозумілими. Сучасні дослідження зосереджено на аналізі взаємозв'язку у виході BERT як результату ретельно підібраних послідовностей входу, внутрішніх векторних подань за допомогою зондувальних класифікаторів, та взаємозв'язків, поданих вагами уваги. Наведемо приклади роботи цієї системи.

Текст переданий на вхід: "*Like keyphrase extraction, document summarization aims to identify the essence of a text. The only real difference is that now we are dealing with larger text units—whole sentences instead of words and phrases. Before getting into the details of some summarization methods, we will mention how summarization systems are typically evaluated. The most common way is using the so-called ROUGE (Recall-Oriented Understudy for Gisting Evaluation) measure. This is a recall-based measure that determines how well a system-generated summary covers the content present in one or more human-generated model summaries known as references. It is recall-based to encourage systems to include all the important topics in the text. Recall can be computed with respect to unigram, bigram, trigram, or 4-gram matching. For example, ROUGE-1 is computed as division of count of unigrams in reference that appear in system and count of unigrams in reference summary*."

Запитання 1: "*What is the text summarization goal?*".
Відповідь 1: "*Identify the essence of a text*".

Запитання 2: "*What is the most used evaluation metric for text summarization?*".
Відповідь 2: "*Recall-Oriented Understudy for Gisting Evaluation*".

Запитання 3: "*Why it uses recall?*".
Відповідь 4: "*To encourage systems to include all the important topics in the text*".

Система правильно відповіла на кожне із запитань. Перевагами цієї системи є точність і широкий спектр запитань, на які вона може відповідати, бо немає ніяких обмежень на те, яке запитання задавати. Єдиною вимогою є те, щоб відповідь на нього була у текстовому контенті, що передається на вхід.

Попри високу точність ця система не має можливості виводити рейтинговий список знайдених

результатів на підставі багатьох джерел. Тобто вона дає одну відповідь на запитання, і ця відповідь має бути певним загальновідомим фактом, описаним у тексті, що передається на вхід. Наша система працює із інформацією, яка є суб'єктивною, на основі великої кількості суб'єктивних думок, знаходить найімовірнішу відповідь. Тобто ця система вирішує ширше завдання, а мета цієї роботи – досягти максимальної точності на вузькому спектрі запитань, в чому і полягає *наукове обґрунтування нашого наукового дослідження*.

Методи та моделі для розв'язання задач під час реалізації проєкту CQA-системи

NLP – підрозділ інформатики/IT та штучного інтелекту (англ. artificial intelligence, AI), який вивчає, як інформаційні системи (IC) аналізують природні (людські) мови. NLP дає змогу застосовувати ML-алгоритми текстового контенту і природної мови [24–30].

N-грама – послідовність з *n* елементів. Із семантичного погляду це може бути послідовність звуків, складів, слів або літер. На практиці частіше трапляються N-грами як послідовність слів, стійких словосполучень (колокація). У NLP-галузі N-грами використовують переважно для передбачення на основі імовірнісних моделей. N-грамна модель розраховує ймовірність останнього слова N-грами, якщо відомі всі попередні. Використовуючи цей підхід для моделювання мови, передбачають, що поява кожного слова залежить тільки від попередніх слів. Інше застосування N-грам – виявлення плагіату. Якщо розділити текст на декілька невеликих фрагментів, поданих N-грамами, їх легко порівняти один з одним, і визначити ступінь подібності контрольованих/аналізованих текстів. N-грами часто успішно використовують для категоризації текстового контенту та мови. Крім того, їх можна застосовувати для створення функцій, які дають змогу отримувати знання із текстового контенту [31–33]. Використовуючи N-грами, можна ефективно знайти пропозиції, щоб замінити слова із помилками правопису/синтаксису/перекладу.

Під час експериментування з англомовними текстами використано рекурентну нейронну мережу для генерування тексту та виділено N-грами. Текст поділено на біграми та триграми.

Основна частина коду, використаного для виділення біграм та триграм:

```
import nltk
from nltk import bigrams
from nltk import trigrams

text="""Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam ornare
tempor lacus, quis pellentesque diam tempus vitae. Morbi justo mauris,
congue sit amet imperdiet ipsum dolor sit amet, consectetur adipiscing elit. Nullam ornare
tempor lacus, quis pellentesque diam"""

tokens = nltk.word_tokenize(text)
tokens = [token.lower() for token in tokens if len(token) > 1]  # same as unigrams
bi_tokens = bigrams(tokens)
tri_tokens = trigrams(tokens)
tri_tokens = list(tri_tokens)
print([(item, tri_tokens.count(item)) for item in sorted(set(tri_tokens))])
```

Результат виконання програми:

```
[(('adipiscing', 'elit', 'nullam'), 2), (('amet', 'consectetur', 'adipiscing'), 2), (('amet', 'imperdiet', 'ipsum'), 1), (('congue', 'sit', 'amet'), 1), (('consectetur', 'adipiscing', 'elit'), 2), (('diam', 'tempus', 'vitae'), 1), (('dolor', 'sit', 'amet'), 2), (('elit', 'nullam', 'ornare'), 2), (('imperdiet', 'ipsum', 'dolor'), 1), (('ipsum', 'dolor', 'sit'), 2), (('justo', 'mauris', 'congue'), 1), (('lacus', 'quis', 'pellentesque'), 2), (('lorem', 'ipsum', 'dolor'), 1), (('mauris', 'congue', 'sit'), 1), (('morbi', 'justo', 'mauris'), 1), (('nullam', 'ornare', 'tempor'), 2), (('ornare', 'tempor', 'lacus'), 2), (('pellentesque', 'diam', 'tempus'), 1), (('quis', 'pellentesque', 'diam'), 2), (('sit', 'amet', 'consectetur'), 2), (('sit', 'amet', 'imperdiet'), 1), (('tempor', 'lacus', 'quis'), 2), (('tempus', 'vitae', 'morbi'), 1), (('vitae', 'morbi', 'justo'), 1)]
```

Під час експериментування натреновано LSTM (один із видів рекунентної нейронної мережі), яка б давала можливість продовжувати речення. Тренування відбувалось і на українському, і на англійському тексті протягом 60 епох. Для тренування використано твір Джорджа Орвела "1984"

українською мовою; з англомовних текстів “1984”, а також твори Фрідріха Ніцше та Вільяма Шекспіра. Для реалізації нейронної мережі використано бібліотеку Keras. Проаналізувавши результати, які повертала нейронна мережа, можна помітити, що незважаючи на мову, на якій тренувалась нейронна мережа, вона повертала набір букв, який містив слова та словосполучення реальної мови. Цікаво, що залежно від тексту, на якому тренували нейронну мережу, вона повертала текст у стилі висловлювання того чи іншого автора. Ще одне цікаве спостереження: натренована на віршах Вільяма Шекспіра нейронна мережа повертала віршований текст.

Для ранжування результатів також вирішено застосувати TF-IDF. Це статистичний показник, який використовують для оцінювання важливості слів у контексті тексту, що є частиною колекції потоку контенту чи корпусу текстів [34]. Вага (значущість) слова пропорційна до кількості вживань цього слова у тексті й обернено пропорційна до частоти появи слова у інших текстах колекції. Показник TF-IDF використовують у задачах аналізу текстового контенту та ІІП. Його можна застосовувати як один із критеріїв релевантності контенту до пошукового запиту, а також під час розрахунку міри спорідненості текстового контенту для кластеризації. Найпростішу функцію ранжування можна визначити як суму TF-IDF кожного терміну в запиті. Більшість функцій ранжування ґрунтуються на цій простій моделі. TF (англ. term frequency – частота слова) – відношення кількості появи вибраного слова до загальної кількості слів текстового контенту. IDF (англ. inverse document frequency – обернена частота текстового контенту) – інверсія частоти, з якою слово трапляється у текстах колекції. Використання IDF зменшує вагу широковживаних слів [35].

$$TF = \frac{n_i}{\sum_k n_k}, IDF = log\frac{|D|}{|(d_i \supset t_i)|}, TF\text{-}IDF = TF \cdot IDF,$$

де n і n_i – кількість входжень слова в контент, а в знаменнику – загальна кількість слів у контенті; $|D|$ – кількість текстового контенту колекції;$|(d_i \supset t_i)|$ – кількість текстового контенту, в якому виявлено слово t_i (коли $n_i \neq 0$) [35].

Інший підхід – розмічування частин мови (англ. part-of-speech tagging) [36–79]. Для цього можна використати готові рішення, наприклад, популярну бібліотеку NLTK. Результатами є слова, позначені мітками як POS tags. Мітка, що відповідає власним назвам (англ. proper nouns), – це NNP.

TextRank – популярний алгоритм знаходження ключових слів на основі графів [40]. Алгоритми ранжування на основі графів є способом визначення важливості вершини в графі на основі глобальної інформації, рекурсивно отриманої із усього графа [41–46]. Основна ідея, реалізована в моделі рейтингування на основі графів, – це “голосування” або “рекомендація”. Коли одна вершина зв’язується з іншою, вона ніби віддає голос за цю іншу вершину. Що більше голосів віддано за вершину, то вища її важливість. І навіть більше, важливість вершини, яка подає голос, визначає, наскільки важливе саме голосування, і цю інформацію також враховує модель рейтингу. Отже, оцінка, пов’язана із вершиною, визначається на основі голосів, поданих за неї, і оцінки вершин, які віддають ці голоси, тобто для обчислення ваги вершини:

$$S(V_i) = (1 - d) + d\sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j),$$

де коефіцієнт демпфування (d), який можна встановити від 0 до 1, виконує роль інтегрування в моделі ймовірності стрибка з цієї вершини до іншої випадкової вершини в графі; $Out()$ – список вихідних вершин; $In()$ – список вхідних вершин. Очікуваним кінцевим результатом для задачі знаходження ключових слів є набір слів або фраз, репрезентативних для тексту природною мовою. Отже, одиниці, які підлягають ранжуванню, є послідовностями однієї або кількох лексичних одиниць, виділених із тексту, і вони представляють вершини, які додаються до текстового графа. Будь-яке відношення, яке можна визначити між двома лексичними одиницями, є потенційно корисним зв’язком (ребром), що можна додати між двома такими вершинами. Алгоритм використовує відношення спільної зустрічальності, яке контролюється відстанню між зустрічами слів: дві вершини з’єднуються, якщо відповідні лексичні одиниці зустрічаються разом у вікні з максимальною кількістю слів, де можна встановити від 2 до 10 слів.

Формально задача ідентифікації семантично подібних речень або частин речень відповідно до запиту користувача у подібному реченні (як набір множини ключових слів) може розглядатись як завдання “Recognising Textual Entailment”. The Recognising Textual Entailment (RTE) задача розглядається як розпізнавання двох фрагментів тексту: чи можна значення одного вивести з іншого. Ця задача не прив’язана до домену і запропонована для розпізнавання мінливості змістових виразів, які зазвичай потрібні у багатьох задачах [1–7]. Фундаментальний феномен природної мови – це різноманітність семантичних виразів, у яких те саме значення може бути виражено або логічно виведено з різних текстів. Цей феномен можна розглядати як проблему мовної двозначності, в якій формуються зв’язки багато до багатьох між мовними виразами і значеннями [7]. Текстове відношення логічного висновку між двома текстами: T (текст) та H (гіпотеза) представляє фундаментальний феномен природної мови. Це позначається як T → H й означає, що значення H можна логічно вивести із T [7]. Це відношення напрямлене, бо значення одного виразу (e.g. “buy”) можна зазвичай логічно вивести з іншого (e.g. “own”). Проте логічне виведення в іншу сторону менш очевидне [1–12]. Текстовий логічний висновок залежить від контексту, нетранзитивний і немонотонний [7]. Задачу генерування/розпізнавання семантично подібних речень можна звести до задачі генерування тексту та перевірки того, чи згенерований текст семантично подібний до зразка.

Розпізнавання текстового логічного виведення – це одне із найскладніших завдань опрацювання природної мови і прогрес у цьому завданні є ключем до розв’язання інших задач, таких як знаходження відповідей на запитання (Question Answering), видобування інформації (Information Extraction), пошук інформації (Information Retrieval), сумаризація тексту (Text Summarization) тощо. Наприклад, система відповідей на питання повинна ідентифікувати текст, що логічно виводиться як очікувана відповідь. Задано питання, текст логічно виводиться з очікуваної відповіді. На основі подібності фрагментів тексту у пошуку інформації запит повинен логічно виводитись із отриманих документів. У сумаризації надлишкові речення можна упускати, якщо їх можна логічно вивести з інших речень. У завданні видобування інформації логічне виведення є між різними варіантами т

ексту, що виражають однакове відношення до цільового тексту. У перевірці машинного перекладу правильний переклад повинен бути семантично подібним до зразкового перекладу і тому вони повинні логічно виводитись один із одного. Тому так само у задачі визначення значення слів (Word Sense Disambiguation), що розглядається як загальна задача, розв’язання задачі логічного виведення може консолідувати дослідження в прикладній задачі семантичного виведення [1–9]. Механізм автоматичного генерування різних перефразувань одного речення істотно практично впливатиме на системи генерування тексту, які приймають текст на вхід і видають текст. Прикладні задачі передбачають сумаризацію і переписування тексту. Інакше цікаве застосування – це використання генерування семантично подібних речень для розширення наборів даних, із додаванням декількох версій їхніх речень. Це корисно як для машинного перекладу, так і для так званих аргументацій даних під час пошуку відповідей в QA-системах, які використовують для тренування моделей машинного навчання [7].

Нещодавній швидкий прогрес дослідження нейромережевої природної мови, особливо на вивченні семантичних текстових зображень, може дати змогу створювати справді нові продукти. Це також може допомогти підвищити продуктивність на різних завданнях, пов’язаних із природною мовою, із обмеженою кількістю навчальних даних, таких як побудова сильних класифікаторів тексту із лише 100 наведених прикладів. Побудова онтології конкретного домену сьогодні спирається на інтуїцію інженера знань, а типовий вихід – це тезаурус термінів, кожен з яких, як очікується, позначає поняття. Онтологічні інженери, як правило, розробляють тезаурус на спеціальній основі й на порівняно невеликому рівні. Працівники в конкретному домені створюють власну спеціальну мову і один пристрій для цього створення є повторенням вибраних ключових слів для консолідації, або відхилення одного чи більше понять. Масштабованіший, систематичний і автоматичний підхід до

побудови онтологіїьможливий завдяки автоматичній ідентифікації цих ключових слів. Використовують підходи до вивчення та видобування ключових слів, аналізуючи корпус випадково зібраних неструктурованих, тобто таких, що не містять будь-якого типу націнки, тексти в конкретній області, із посиланням на лексичні уподобання працівників домену. Аналіз часто вживаних слів у словосполученнях приводить до створення семантичної мережі. Мережу можна ввести в термінологічну базу даних або забезпечити формалізм подання знань, а взаємозв'язок між вузлами мережі допомагає у візуалізації та автоматичному висновку над часто використовуваними словами, що позначають важливі поняття в області [7–8; 18; 25; 34; 41–42].

Підхід ідентифікації семантичної подібності рівня речення на основі неконтрольованого вивчення розмовних даних сьогодні доволі актуальний. Зауважимо, що семантично подібні вхідні речення мають подібний розподіл речень відповіді, а модель, навчена для прогнозування взаємозв'язків між входами, повинна неявно навчитися корисним семантичним поданням. "Скільки вам років?" та "Який ваш вік?" – обидва питання про вік, на які можна дати схожі відповіді, наприклад, "Мені 20 років". Натомість запитання "Як ти?" та "Скільки вам років?" використовують подібні слова, але мають різні значення і приводять до різних відповідей (рис. 3) [7, 47, 48].

Рис. 3. Питання семантично подібні, якщо на них можна відповісти тими самими відповідями. Інакше вони семантично різні

Методи генерування векторного подання слів охоплюють нейронні мережі, зменшення розмірності матриці суміжної поширеності слів, імовірнісні моделі, метод побудови бази знань та явне подання у термінах контексту, в якому з'являються слова [7]. Репрезентації слів, обчислені з використанням нейронних мереж, дуже цікаві, оскільки отримані вектори кодують багато мовних закономірностей і шаблонів. Багато з цих структур можна подати як лінійні перетворення. Для багатьох завдань опрацювання природньої мови доступні обмежені обсяги навчальних даних. Це виклик для глибоких методів навчання даних. Через високу вартість анотування даних під наглядом дуже великі навчальні набори зазвичай недоступні для більшості дослідницьких та галузевих задач. Universal Sentence Encoder – це модель, що розширює багатозадачне навчання, додаючи більше задач. Модель намагається уточнити/зрозуміти текст, отримавши на вхід частину тексту. Однак замість архітектури кодера-декодера в оригінальній моделі використано архітектуру тільки для кодування за допомогою спільного кодування для керування завданнями прогнозування. Отже, час навчання істотно зменшується, зберігаючи продуктивність на різних завданнях, ураховуючи класифікацію настроїв і семантичну подібність (рис. 4) [7, 47, 48].

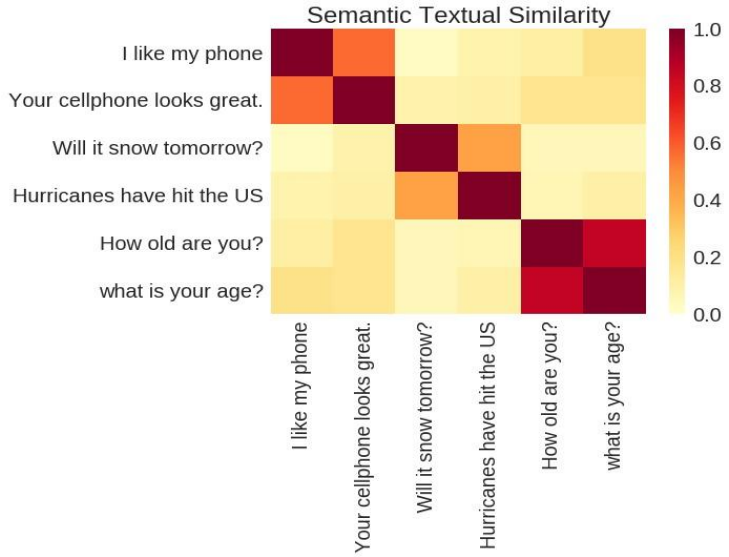

Рис. 4. Порівняння семантичної подібності по парах речень через Universal Sentence Encoder [48]

Мета полягає у тому, щоб забезпечити єдину модель кодування, яка може підтримувати якомога більше різноманітних додатків, ураховуючи виявлення перефразування, спорідненість, кластеризацію і класифікацію тексту (рис 5) [7, 47, 48]. Дві версії Universal Sentence Encoder використовують різні архітектури. Простіша версія, яка працює швидше, але з дещо меншою точністю, застосовує Deep Average Network (DAN), складніша версія використовує архітектуру Transformer [49].

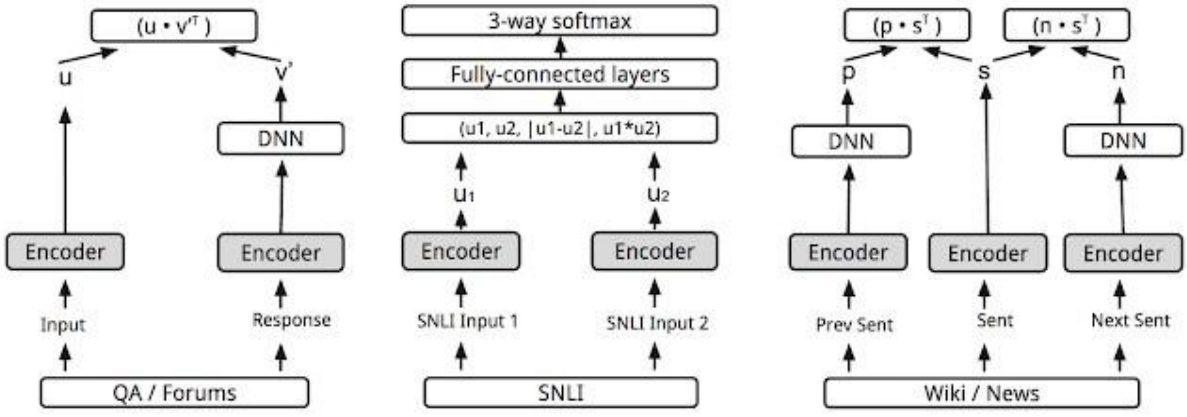

Рис. 5. Використання Universal Sentence Encoder для різнотипних задач [49]

Згідно з фразовою структурою граматика речення складається із іменної фрази та дієслівної фрази (рис. 6).

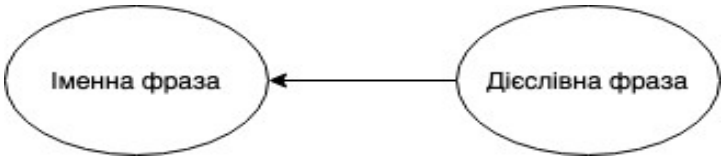

Рис. 6. Схема взаємозв'язків згідно з фразовою структурою речення у англійській мові

Набір даних Microsoft Research Paraphrase Corpus (MSRP) складається із 5801 пари речень,

кожна супроводжується бінарним судженням, яке вказує, чи люди, які оцінювали, вважають цю пару речень достатньо подібною, щоб вони вважались перефразуванням. Ці дані опубліковані з метою заохочувати до дослідження у сферах, пов'язаних із перефразуванням і логічним виведенням, та допомогти створити дискурс щодо належної побудови корпусів перефразувань для навчання та оцінювання. У деяких випадках два речення було оцінено як "семантично однакові", хоча насправді вони семантично розходились, принаймні якоюсь мірою. Наприклад, два судді вважали два речення перефразованими [50]:

- Charles O. Prince, 53, was named as Mr. Weill's successor.
- Mr. Weill's longtime confidant, Charles O. Prince, 53, was named as his successor.

Перефразування може розглядатись як логічне виведення в обидві сторони If a full paraphrase relationship can be described as "bidirectional entailment", then the majority of the "equivalent" pairs in this dataset exhibit "mostly bidirectional entailments". У цьому наборі даних одне речення має інформацію, якої немає в іншому реченні, тобто у ньому немає строгого перефразування, а дозволений певний ступінь свободи.

Набір даних Quora Question Pairs містить 400,000 рядків потенційний питань-дублікатів. Наприклад, Each line contains IDs for each question in the pair, the full text for each question, and a binary value that indicates whether the line truly contains a duplicate pair. Кілька зразків із набору даних [51]:

id	qid1	qid2	question1	question2	is_duplicate
447	895	896	What are natural numbers?	What is a least natural number?	0
1518	3037	3038	Which pizzas are the most popularly ordered pizzas on Domino's menu?	How many calories does a Dominos pizza have?	0
3272	6542	6543	How do you start a bakery?	How can one start a bakery business?	1
3362	6722	6723	Should I learn python or Java first?	If I had to choose between learning Java and Python, what should I choose to learn first?	1

Рис. 7. Зразок даних Quora Question Pairs

Важливі твердження про набір даних:

- Початковий процес відбору дав незбалансований набір даних, в якому більше пар речень, що є дублікатами. Тому його доповнили також негативними прикладами.
- Розподіл питань не повинен розглядатись як реальний розподіл питань на Quora.
- Мітки класів мають певний шум, тобто не всі правильно позначені.

Набір даних Multi-Genre Natural Language Inference (MultiNLI) – це набір даних, який складається із 433,000 прикладів і є найбільшим для розпізнавання логічного виведення [52]. MultiNLI охоплює десять різних жанртів письмової та розмовної англійської, що дає змогу перевіряти системи на близькій до реальної складності мові (рис. 8).

Met my first girlfriend that way.	FACE-TO-FACE **contradiction** C C N C	I didn't meet my first girlfriend until later.
8 million in relief in the form of emergency housing.	GOVERNMENT **neutral** N N N N	The 8 million dollars for emergency housing was still not enough to solve the problem.
Now, as children tend their gardens, they have a new appreciation of their relationship to the land, their cultural heritage, and their community.	LETTERS **neutral** N N N N	All of the children love working in their gardens.
At 8:34, the Boston Center controller received a third transmission from American 11	9/11 **entailment** E E E E	The Boston Center controller got a third transmission from American 11.
I am a lacto-vegetarian.	SLATE **neutral** N N E N	I enjoy eating cheese too much to abstain from dairy.
someone else noticed it and i said well i guess that's true and it was somewhat melodious in other words it wasn't just you know it was really funny	TELEPHONE **contradiction** C C C C	No one noticed and it wasn't funny at all.

Рис. 8. Довільно вибрані приклади з MultiNLI

Речення природної мови потрібно класифікувати на тр класи: "entailment", "neutral",

“contradiction”. “Entailment” – значення другого речення можна логічно вивести із першого. “Neutral” – значення другого речення неможливо логічно вивести з першого через недостатню кількість даних у першому реченні чи через різний зміст речень (наприклад, речення із різних доменів). “Contradiction” – значення другого речення суперечить значенню першого.

Речення зразок перетворюється на речення з інакшим формулюванням, але з таким самим змістом, як перше. Між двома реченнями повинна виконуватись умова семантичної подібності. Тобто під час класифікації цієї пари речень їх необхідно класифікувати як “entailment”.

Векторне подання слів набуло поширення в опрацюванні природних мов. Вони дають змогу легко обчислити семантичну подібність між двома словами або знайти слова, найподібніші на цільове слово. Однак часто нас цікавить подібність двох речень або коротких текстів. Багато додатків повинні обчислити подібність між двома короткими текстами. Пошукові системи, наприклад, повинні моделювати релевантність документа до запиту, за винятком перекриття у словах між ними. Аналогічно, сайти запитань і відповідей, такі як Quora, повинні визначити, чи було запитання вже поставлено раніше. Цей тип подібності тексту часто обчислюють, спочатку створюючи векторне подання двох коротких текстів, а потім визначаючи подібність косинуса між ними. Хоча вбудовування слів, наприклад, word2vec і GloVe, стало стандартними підходами для знаходження семантичної подібності між двома словами, існує менша згода щодо того, як обчислювати векторне подання речення. Нижче розглянемо деякі з найпоширеніших методів і порівняємо їх ефективність за двома встановленими тестами. Найпростішим способом оцінювання семантичної подібності між парою пропозицій є прийняття середнього значення векторних подань слів у двох реченнях і обчислення косинуса між отриманими векторами. Очевидно, що цей простий базовий рівень залишає значний простір для варіацій. Досліджуватимемо наслідки ігнорування стоп-слів і обчислення середньозваженого TF-IDF. Однією із цікавих альтернатив базового підходу є Word Mover Distance. Цей підхід використовує векторні подання слів у двох текстах для вимірювання мінімальної відстані, на яку слова в одному тексті повинні “подорожувати” у семантичному просторі, щоб досягти слів у іншому тексті.

Прийняття середнього значення векторних подань слів у реченні (Smooth Inverse Frequency) означає тенденцію надавати занадто велику вагу словам, які є абсолютно незначними, семантично кажучи. Цей підхід намагається вирішити цю проблему двома способами:

· Зважування: як і у нашого базового алгоритму TF-IDF вище, SIF приймає середньозважене значення векторного подання слова у реченні. Кожне подання слова зважується на $a/(a + p(w))$, де a є параметром, який зазвичай встановлюють на 0,001, а $p(w)$ – оцінювана частота слова в корпусі посилання.

· Загальне видалення компонентів: надалі SIF обчислює головний компонент результівних подань для набору речень. Потім він віднімає з цих подань їх проєкції на їх перший головний компонент. Це має усунути варіації, пов’язані з частотою та синтаксисом, що є менш актуальним семантично.

Всі вищенаведені методи мають дві важливі характеристики. Прості способи мішків слів не враховують порядок слів. А векторне подання слова краще застосовувати для навчання без учителя. Обидві ці риси потенційно небезпечні. Оскільки відмінності в порядку слів часто супроводжуються відмінностями в сенсі, ми хотіли б, щоб вбудовування речення було чутливим до цієї варіації. Крім того, навчання з учителем може допомогти вкладенню речень безпосередньо дізнатися значення речення. Попередньо підготовлені моделі для кодування речення прагнуть відігравати ту саму роль, що і word2vec і GloVe, але для векторного подання речення: подання, яке вони здійснюють, можна використовувати в різних додатках, таких як класифікація тексту, переказ тощо. Як правило, вони використовували навчання з вчителем та без, щоб захопити якомога більше універсальної семантичної інформації.

Модель Roberta показує state-of-the-art результати на GLUE, RACE і SQuAD. Це модель, яка має таку саму архітектуру, як модель BERT, але із невеликими модифікаціями. Точність моделі

істотно покращується у разі довшого тренування моделі із більшими батчами на більшій кількості даних; вилучивши мету передбачення наступного речення; тренувавши на довших послідовностях; і послідовно змінюючи шаблон маски під час тренування.

Завдання опрацювання природної мови, такі як відповіді на запитання, сумаризація, машинний переклад і розуміння тексту, часто вирішуються навчанням з учителем на спеціалізованих наборах даних. Модель генерування тексту GPT-2 демонструє, що модель починає вивчати ці завдання без явного спеціалізованого тренування, коли вона натренована на наборі даних мільйонів вебсторінок WebText. Цікавою ознакою моделі є те, що вона натренована на задачі моделювання мови, хоча демонструє певний успіх і у задачах, на яких не була явно натренована. Найбільша модель GPT-2 має 1,5 мільярда параметрів і архітектуру Transformer. Вона досягає state of the art результатів на семи із восьми протестованих наборів даних моделювання мови із zero-shot налаштуванням, але все ще має недонавчання (underfit) на WebText.

Будь-який пошук семантичної подібності передбачає аналіз на відношення еквівалентності (~) на множині *X*, для якого виконуються такі умови: рефлексивність, симетричність та транзитивність. Запис вигляду "$a \sim b$" читають як "a еквівалентно b". Наслідком властивостей рефлексивності, симетричності та транзитивності є те, що будь-яке відношення еквівалентності забезпечує розподіл будь-якої базової множини на непересічні класи еквівалентності. Два елементи цієї множини еквівалентні між собою тоді й тільки тоді, коли вони належать до одного класу еквівалентності. Наведемо приклади відношень еквівалентності:

· Найнаочніший приклад відношення еквівалентності – поділ учнів школи на класи.

· Відношення рівності – тривіальне відношення еквівалентності на довільній множині, зокрема на множині дійсних чисел.

· Порівняння за модулем.

· В евклідовій геометрії відношення конгруентності, подібності та паралельності прямих.

· Відношення рівнопотужності множин є відношенням еквівалентності.

Якщо для транзитивності задачі Recognising Textual Entailment виконується відношення транзитивності між набором виразів, то їх можна подати в ієрархічній графовій структурі [7].

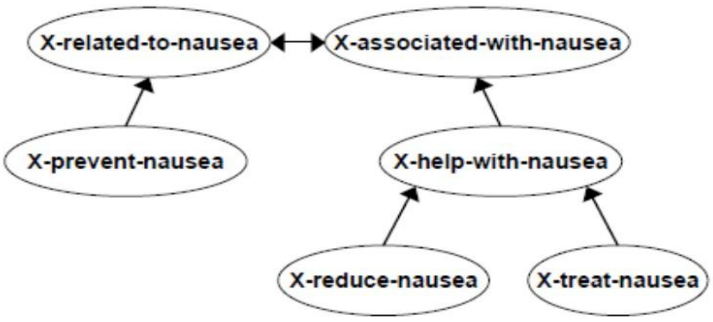

Рис. 9. Ієрархічна графова структура транзитивних залежностей між словами

Подані у вигляді такої структури дані можна використати для тренування моделі. Модель, натренована на даних, у яких є транзитивні відношення, навчиться реалізовувати це відношення.

Виклад основного матеріалу

UML-діаграма послідовності відображає взаємодію об'єктів системи, впорядкованих за часом, та послідовність надісланих повідомлень. Є три основні об'єкти системи, між якими здійснюється обмін повідомленнями: Computer, Google search та Reddit API (рис. 10–12).

Computer – це комп'ютер, на якому запущено систему і через який користувач користується цією системою. Для встановлення системи йому треба встановити інтерпретатор Python.

Google search – це вебсторінка Google-пошуку. Система вводить запит користувача у Google пошук, і після цього результати пошуку зберігає для наступного кроку. Результатами попереднього кроку є набір посилань, після цього система переходить за кожним і здійснює вебскрапінг сторінок.

Reddit API – це прикладний програмний інтерфейс (API), який є набором чітко визначених методів для взаємодії різних компонентів сервісу Reddit. Система робить запити до API, щоб отримати коментарі до кожного поста у Reddit, який зібраний на попередньому кроці.

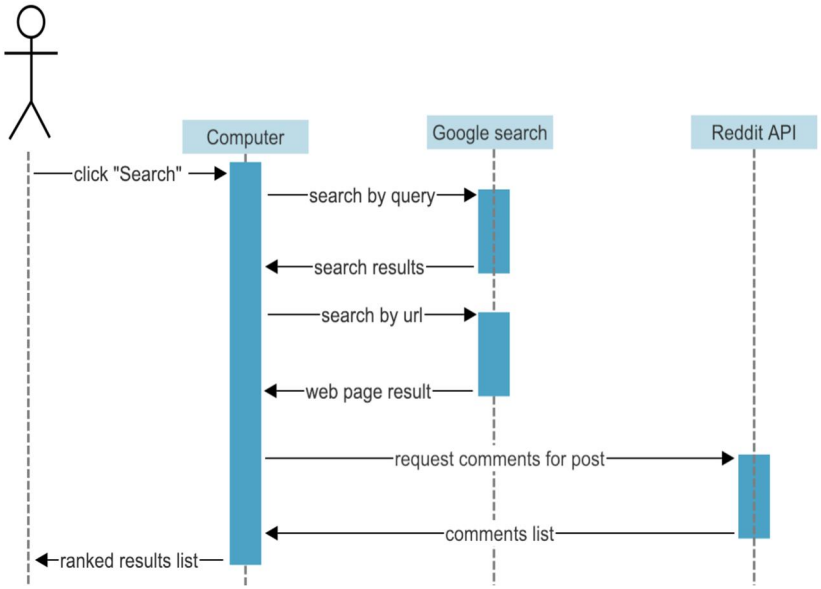

Рис. 10. UML діаграма послідовності

На UML-діаграмі засобів використання подано користувача, який взаємодіє із системою, задаючи запитання (рис. 11). Запитання повинно бути англійською мовою і мати найвищий ступінь порівняння. Після цього система видає проранжований список відповідей на запитання. Відповідями завжди є власні назви об’єктів, таких як книги, програмні продукти тощо. Результат програми також охоплює подання ймовірностей до кожного елемента списку.

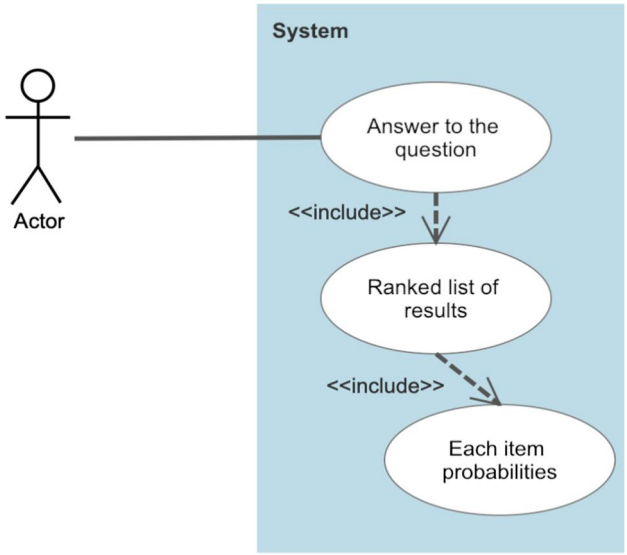

Рис. 11. UML діаграма варіантів використання

Ключовим класом є KeywordsEngine, оскільки він відповідає за основну логіку системи (рис. 12). Він має метод run(), що викликається після того, як користувач ввів запит. Цей клас має

відношення композиції із класами KeywordsSearch і KeywordsRank. KeywordsSearch відповідає за пошук ключових слів у джерелах, а KeywordRank за ранжування цих ключових слів у фінальний список. Результат програми описується класом RankedItems, який містить список елементів, кожен з яких описаний класом RankedItem. RankedItem складається із ідентифікатора (id), власне текстового контенту ключового слова (text) та оцінки (score), на основі якої результуючий список проранжовано. З класом KeywordsEngine асоціативним зв'язком з'єднаний клас Source, він описує кожне з джерел, у якому шукають ключові слова. Основне поле цього класу – це content, в якому зберігається заскраплений текст з цього джерела, та має такі поля, як назва (title), мова (language) тощо.

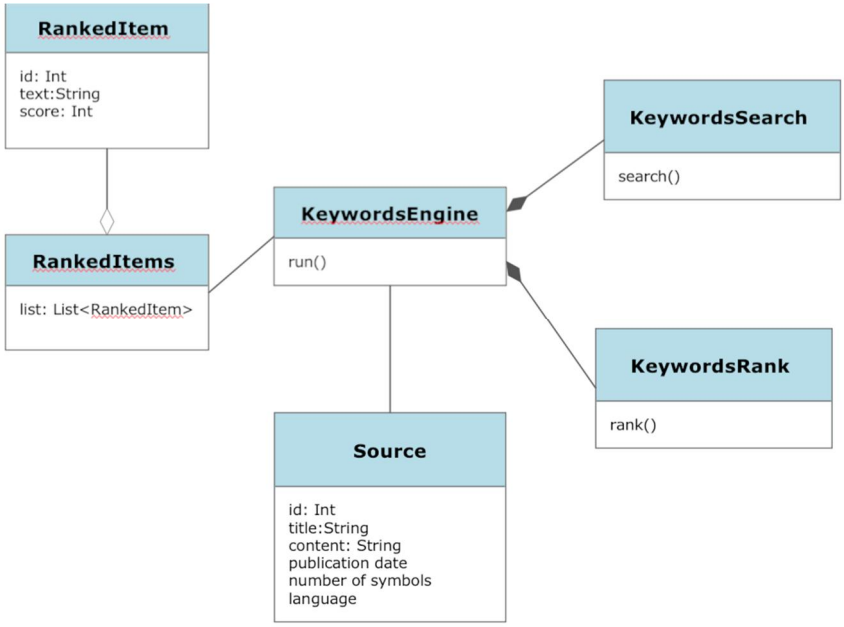

Рис. 12. UML діаграма класів

Діаграма станів відображає три стани, у яких може перебувати система: System is ready to get a question, Sources processing та Presenting results with ranked list of answers (рис. 13). У першому стані система перебуває перед тим, як користувач ввів запитання. Після надання запитання система переходить до його опрацювання, тобто у другий стан. У другому стані вона перебуває, поки опрацьовує кожне джерело. Після опрацювання всіх джерел вона переходить у фінальний стан, який означає подання результатів користувачу. Користувач може виконати дію для задавання іншого запитання, у такому випадку система перейде у початковий стан.

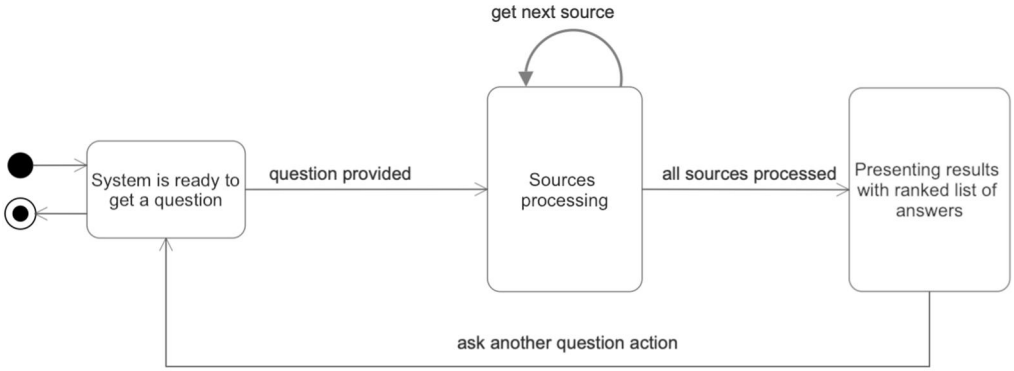

Рис. 13. UML діаграма станів

Задачу можна розв'язувати двома способами. Перший – розмічування частин мови (part-of-speech tagging) і пошук перетину слів між різними джерелами. Тобто результати будуть ранжуватись на основі відношення кількості проаналізованих джерел, до кількості джерел, у яких трапляється ця

власна назва. Другий – використання TF-IDF, у якому TF обчислюватиметься на всіх джерелах разом, а IDF на певному корпусі тексту на загальну тематику. Пошуковий запит: "*The best IDE for Python*". Результати, що видала система: "*pycharm", "atom", "pydev", "code", "spyder", "idle", "anaconda", "Microsoft", "vim", "sublime text", "eclipse", "linux", "github", "git", "how", "java", "howev", "thonny", "studio code", "please", "django", "emacs", "windows", "vs code", "pydev pydev", "top", "visual studio code"*.

З цього прикладу роботи системи можемо зробити висновок, що найпопулярніші варіанти відповідей є найкращими. Тобто у першій відповіді "*pycharm*" справді визначено найпопулярніше середовище розроблення для Python. Четвертим варіантом відповіді є "code", це можна пояснити тим, що одне із найкращих середовищ розроблення – Visual Studio Code, і через проблему із об'єднанням частин власної назви виникли проблеми.

Специфікація вимог до модуля генерування відповіді на запит користувача має вигляд змістовного речення на основі аналізу семантично подільних речень відносно запиту.

Можна виділити такі основні характеристики:

- Ідентифікація перефразованих речень, у яких немає або майже немає однакових слів.
- Гнучкість розпізнавання речень з урахуванням зміни структури речення, використання синонімів чи антонімів.
- Гранування семантично подібних речень.

У модулі буде лише один клас: користувачі додатка. Додаток призначений для людей, які мають потребу автоматично ідентифікувати чи генерувати семантично подібні речення.

Характеристики модуля:

1. Збереження даних.

1.1. Опис і пріоритет. Пріоритет – середній. Можливість зберегти результати генерування та ідентифікації семантично подібних речень.

1.2. Послідовність дія – відгук. Користувач відкриває додаток, вказує відповідний параметр у консолі та назву файла, в який будуть збережені результати.

1.3. Функціональні вимоги:

REQ 1. Інформативне повідомлення про те, що починається процес збереження.

REQ 2. Надання можливості скасувати збереження.

2. Ідентифікація семантично подібних речень.

2.1. Опис і пріоритет. Пріоритет – високий. Можливість автоматичної ідентифікації семантично подібних речень.

2.2. Послідовність дія – відгук. Користувач відкриває додаток та вказує речення, яке потрібно ідентифікувати.

2.3. Функціональні вимоги:

REQ 1. Точність ідентифікації повинна бути достатньо високою, щоб це було ефективніше від неавтоматичного способу.

REQ 2. Надати можливість скасувати процес ідентифікації.

3. Генерування семантично подібних речень.

3.1. Опис і пріоритет. Пріоритет – високий. Можливість автоматичного генерування семантично подібних речень.

3.2. Послідовність дія – відгук. Користувач відкриває додаток.

Користувач вказує речення, яке потрібно перефразувати.

3.3. Функціональні вимоги:

REQ 1. Точність перефразування повинна бути достатньо високою, достатньою, щоб це було ефективніше від генерування перефразувань неавтоматичним способом.

REQ 2. Надання можливості скасувати процес генерування.

Вимоги зовнішніх інтерфейсів:

1. Користувацькі інтерфейси. Користувач може взаємодіями із системою за допомогою персонального комп'ютера, у якому є достатньо обчислювальних ресурсів, щоб працювати із системою.

2. Апаратні інтерфейси. Поточна система не використовуватиме жодних апаратних інтерфейсів.

3. Програмні інтерфейси: NLTK; PyTorch; Keras.

Інші нефункціональні вимоги:

1. Вимоги продуктивності. Система повинна швидко ідентифікувати та генерувати речення без перебору великих баз даних готових зразків.

2. Вимоги безпеки. Персональні дані є конфіденційними і не передаються третім особам. Забезпечити це можна, зробивши цю систему із відкритим вихідним кодом.

3. Атрибути якості програмного продукту: зручність використання; надійність; зручність супроводу.

Нижче наведено основі діаграми модуля генерування відповіді на запит користувача у вигляді змістовного речення на основі аналізу семантично подільних речень щодо запиту (рис. 14–18).

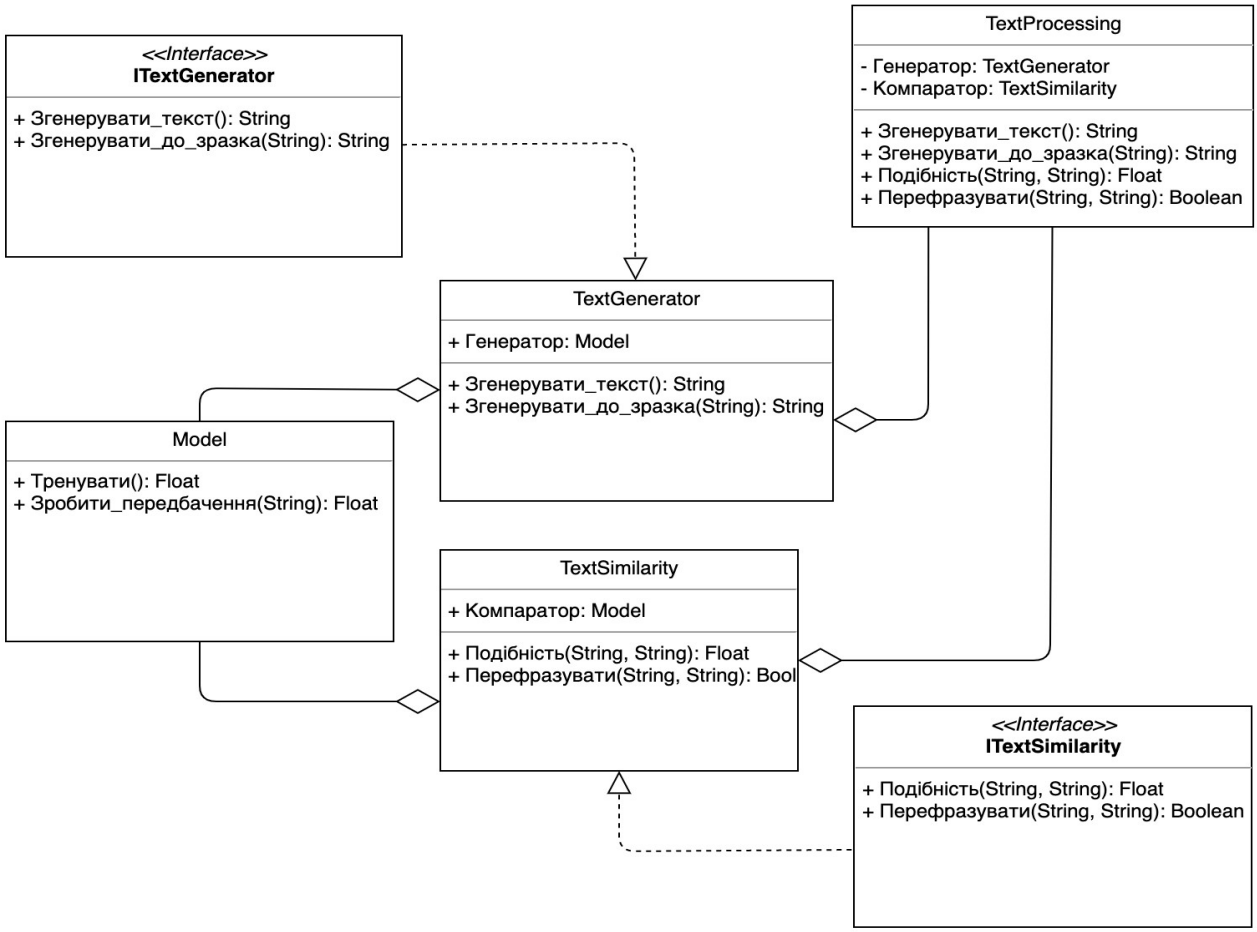

Рис. 14. Діаграма класів

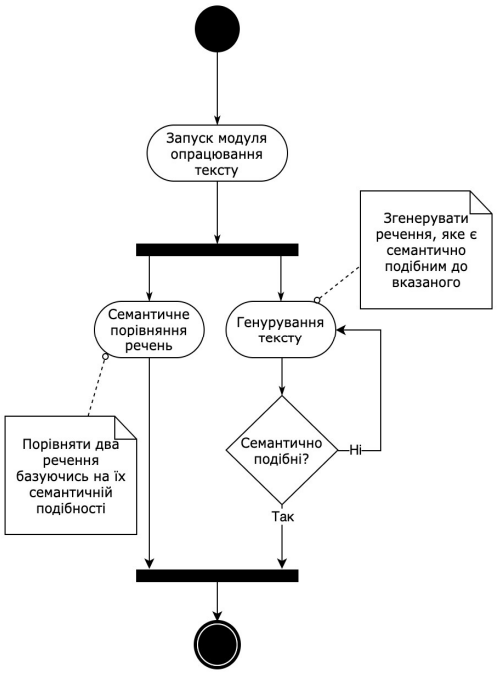

Рис. 16. Діаграма діяльності

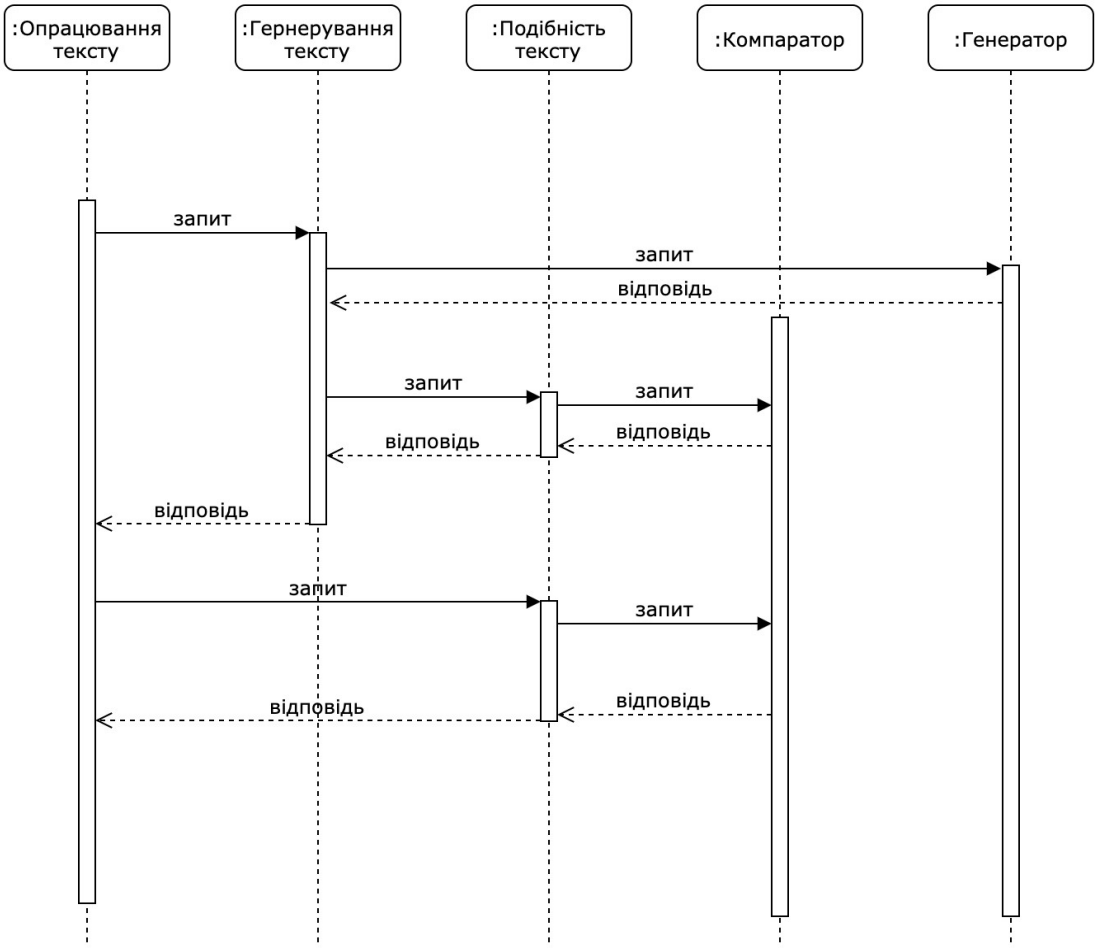

Рис. 15. Діаграма послідовності

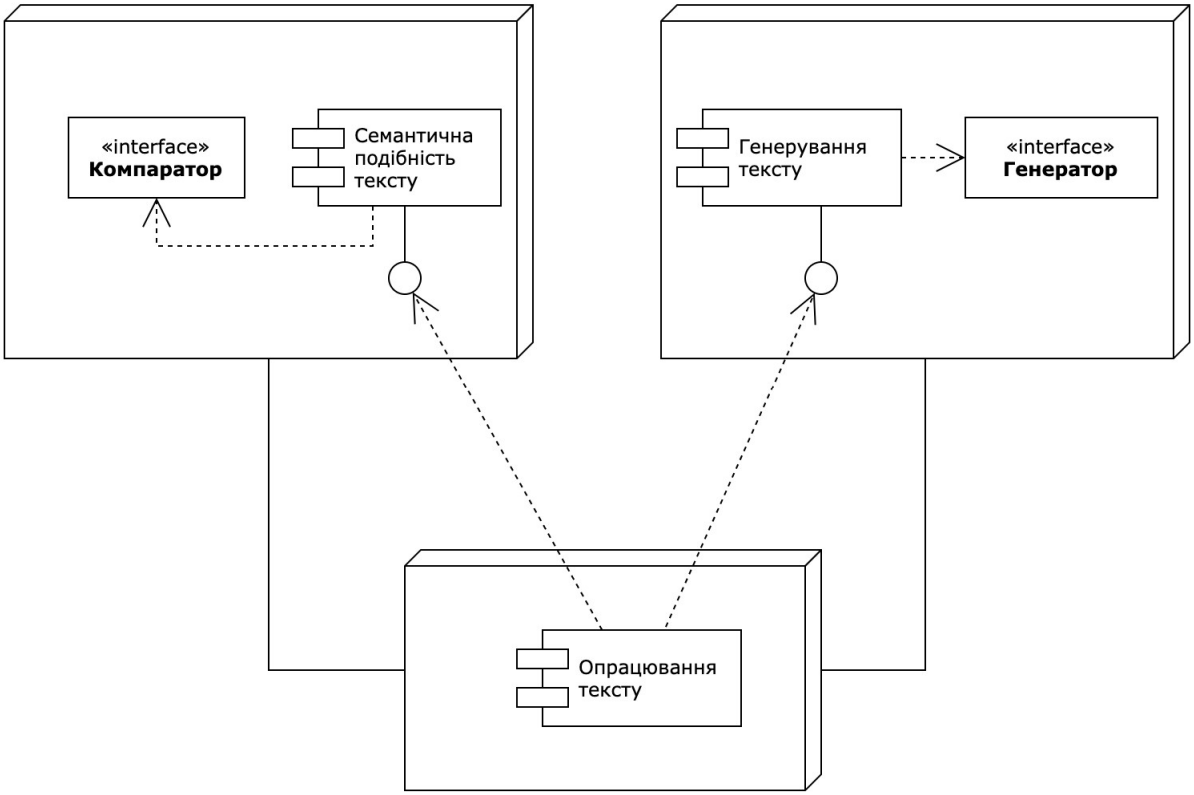

Рис. 17. Діаграма розгортання

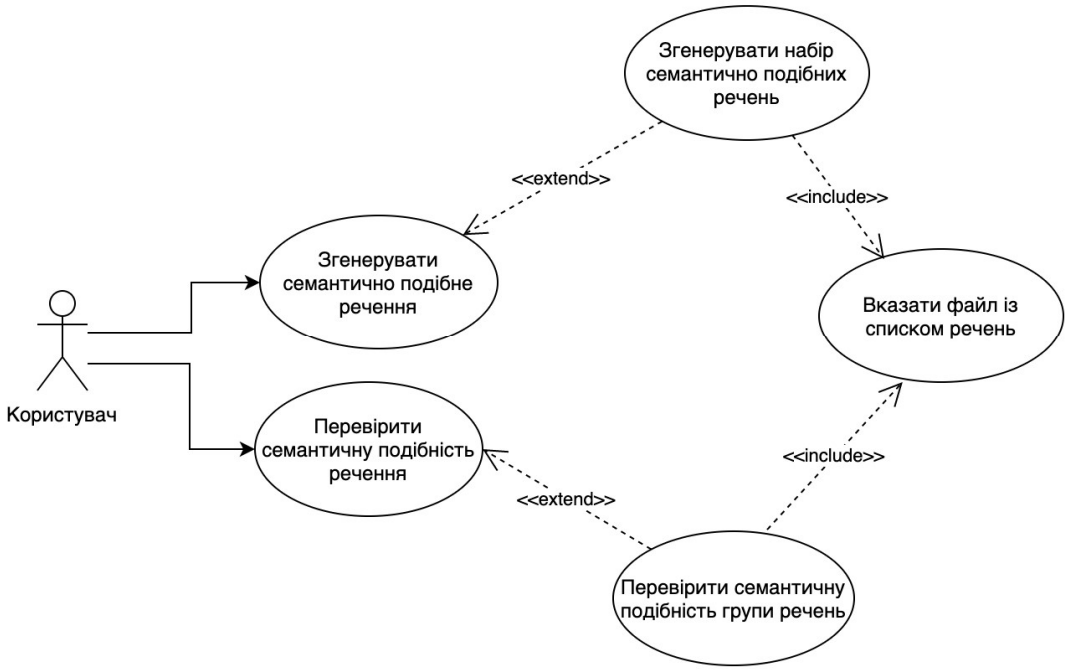

Рис. 18. Діаграма варіантів використання

Інноваційною складовою цієї роботи є дослідження транзитивності у задачі RTE, адже наявні моделі явним способом не реалізують цього відношення. Тобто якщо з одного речення логічно

випливає друге, а з другого третє, то також повинно виконуватись логічне виведення третього речення із першого. Тобто, якщо A → B і B → C, то повинно виконуватись і A → C. Також в обсяг завдань цієї роботи входить дослідження обмежень наявних наборів даних для розв'язання задачі RTE, і обмежень стосовно розв'язання цієї задачі як такої з погляду філософії.

Для тренування і тестування системи використано набір даних MultiNLI. З нього вибрано підмножину набору даних розміром сто тисяч прикладів. Двадцять п'ять відсотків даних виділено на тестування системи, решту – на тренування. Косинус подібності:

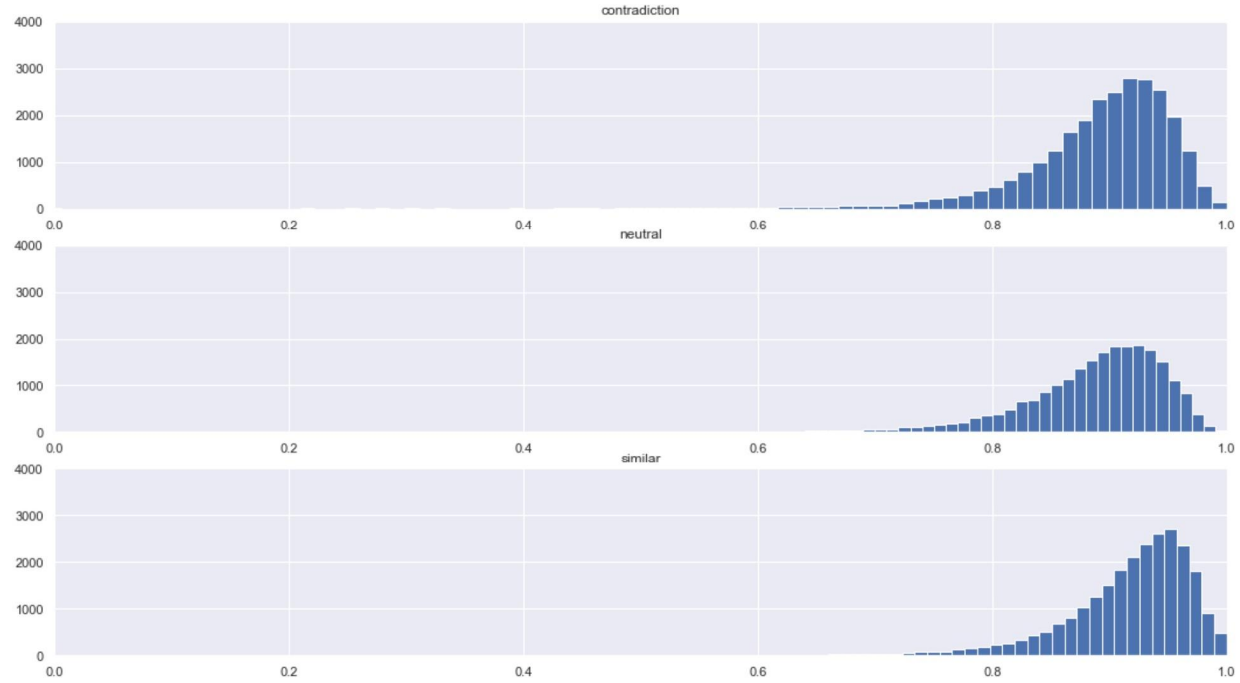

Рис. 19. Гістограми для косинусів подібності для кожного із класів

Середнє арифметичне для класів "similar", "neutral" та "contradiction" дорівнює 0,914, 0,885 та 0,89 відповідно. Медіана для "similar", "neutral" та "contradiction" дорівнює 0.928, 0.898 та 0.904 відповідно. Середнє квадратичне відхилення – 0,062, 0,068 та 0,07 відповідно. Як бачимо, хоча найбільше середнє арифметичне і медіана у класу "similar", проте різниця між значеннями різних класів недостатньо велика для того, щоб їх можна було легко класифікувати, використовуючи цю ознаку. Враховуючи розподіли різних класів, можемо зробити висновок, що ці класи неможливо чітко розділити із використанням лише значень косинуса подібності між векторами. Використаємо косинус кута між векторами для тренування лінійної моделі класифікації. Результати перевірки моделі подано на рис. 20–22.

Metric and class name	The value of the metric
Accuracy	0.40497333333333335
Precision "similar"	0.42551798203106583
Precision "neutral"	0.37994034302759133
Precision "contradiction"	0.3839664919012331
Recall "similar"	0.6424503677924543
Recall "neutral"	0.043975487657517694
Recall "contradiction"	0.4938964659784493

Рис. 20. Значення метрик точності логістичної регресії для кожного із класів

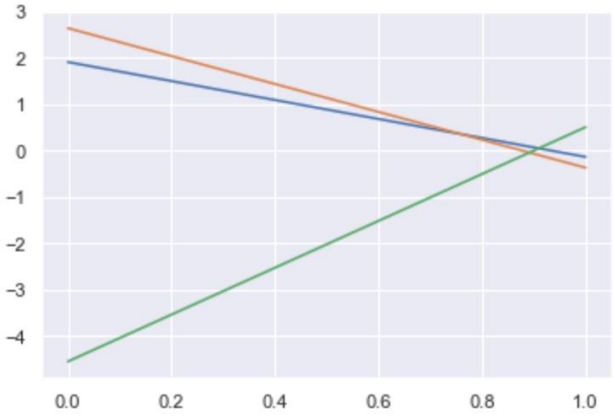

Рис. 21. Прямі, побудовані логістичною регресією для відділення кожного із класів

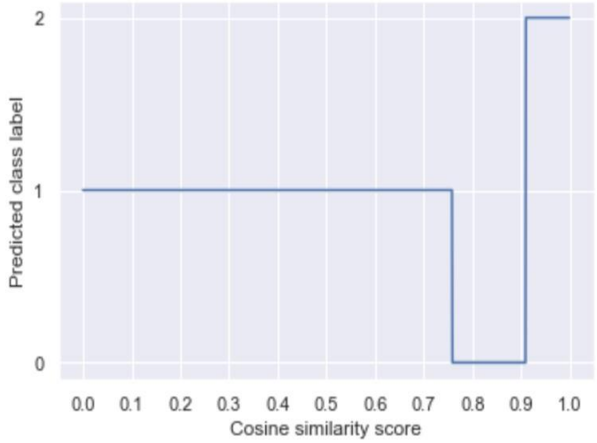

Рис. 22. Графік залежності значення косинуса подібності та результату передбачення логістичної регресії

З графіка на рис. 22 бачимо, що логістична регресія класифікує пару речень як “neutral”, якщо косинус подібності між векторами цієї пари речень міститься у проміжку від 0 до 0,75. За значень косинуса подібності від 0,75 до 0,9 модель видає результат “contradiction”, а за значень від 0,9 до 1 – “similar”. Те, що модель для класу “contradiction” використовує більші значення косинуса подібності, ніж для класу “neutral”, свідчить про те, що формулювання “contradiction” більш схоже на “similar”, ніж на “neutral”, що не є хорошим результатом, бо “contradiction” повинен би був мати менші значення косинуса подіності, ніж “neutral”.

Metric and class name	The value of the metric	
	SGD classifier metric accuracy metrics for each class	the accuracy of the reference vector method for each class
Accuracy	0.3522533333333333	0.40784
Precision “similar”	0.3486562736315514	0.43468647238915975
Precision “neutral”	0.22058823529411764	0
Precision “contradiction”	0.39380674448767833	0.38399959722082366
Recall “similar”	0.9481531282132405	0.6064225262991378
Recall “neutral”	0.000647332988089073	0
Recall “contradiction”	0.09151533418732574	0.5747117775600934

Рис. 23. Значення метрик точності SGD класифікатора для кожного із класів

На рис. 24 подано аналіз середнього між ембедінгами пари речень, на рис. 25 – аналіз середнього ембедінгами пари речень та косинус подібності між векторами пари речень

Metric and class name	The value of the metric
Accuracy	0.50164
Precision “similar”	0.48400272294077606
Precision “neutral”	0.4658757850662945
Precision “contradiction”	0.5472785722203747
Recall “similar”	0.5031847133757962
Recall “neutral”	0.42971163748712665
Recall “contradiction”	0.5639707562257253

Рис. 24. Значення метрик точності логістичної регресії для кожного з класів у разі використання середнього між ембедінгами пари речень

Metric and class name	The value of the metric	
	logistic regression accuracy metrics for each class using the mean between embodies and the cosine of similarity between pairs of sentences	Random Forest accuracy metric values for each class using medium between embedding and cosine similarity between pairs of sentences
Accuracy	0.54196	0.47148
Precision “similar”	0.5365727310401989	0.4736957474791758
Precision “neutral”	0.5303206997084549	0.48207101626727306
Precision “contradiction”	0.557492931196984	0.46352987498769566
Recall “similar”	0.6108752064166076	0.5097900448218919
Recall “neutral”	0.46833161688980435	0.354788877445932
Recall “contradiction”	0.5405528901073795	0.5379255197623943

Рис. 25. Значення метрик точності логістичної регресії для кожного з класів у разі використання середнього між ембедінгами та косинуса подібності між парами речень

Також на рис. 25 подано значення метрик точності класифікатора Random Forest для кожного з класів у разі використання середнього між ембедінгами та косинуса подібності між парами речень. На рис. 26 подано результати визначення символьної відстані між парами речень.

Metric and class name	The value of the metric
Accuracy	0.3724
Precision “similar”	0.7134606317774634
Precision “neutral”	0.002957121734844751
Precision “contradiction”	0.3848809523809524
Recall “similar”	0.356835465424748
Recall “neutral”	0.26666666666666666
Recall “contradiction”	0.40682018371712597

Рис. 26. Значення метрик точності логістичної регресії для кожного з класів у разі використання символьної відстані між парами речень

На рис. 27 подано результати знаходження перетину по словах між парами речень.

Metric and class name	The value of the metric
Accuracy	0.40008
Precision “similar”	0.7511786892975012
Precision “neutral”	0
Precision “contradiction”	0.43202380952380953
Recall “similar”	0.367680147695148
Recall “neutral”	0
Recall “contradiction”	0.47332724664145037

Рис. 27. Значення метрик точності логістичної регресії для кожного з класів у разі використання перетину по словах між парами речень

На рис. 28 подані результати знаходження довжин речення, як ознака для класифікації.

Metric and class name	The value of the metric
Accuracy	0.37364
Precision “similar”	0.3248443689869836
Precision “neutral”	0.40391943385955364
Precision “contradiction”	0.37398934503290504
Recall “similar”	0.13666666666666666
Recall “neutral”	0.2742730409068507
Recall “contradiction”	0.7033239038189534

Рис. 28. Значення метрик точності логістичної регресії для кожного з класів у разі використання довжин речень

На рис. 29 подано результати знаходження кількості слів як ознаки для класифікації.

Metric and class name	The value of the metric
Accuracy	0.37472
Precision “similar”	0.32454212454212455
Precision “neutral”	0.40823844608171467
Precision “contradiction”	0.3708430482267763
Recall “similar”	0.10547619047619047
Recall “neutral”	0.3003942828979793
Recall “contradiction”	0.7123998114097124

Рис. 29. Значення метрик точності логістичної регресії для кожного із класів у разі використання кількості слів

На рис. 30 наведено результати знаходження простих класифікаторів.

The name of the classifier	The accuracy value
Most common class classifier	0.33936
Stratified classifier	0.33712

Рис. 30. Значення метрик точності логістичної регресії у разі використання простих класифікаторів

На рис. 31 подано результати об'єднання разом усіх ознак метрик точності логістичної регресії та класифікатора Random Forest для кожного з класів у разі об'єднання усіх ознак разом.

Metric and class name	The value of the metric	
	The value of logistic regression accuracy metrics for each class when all traits are combined	Random Forest accuracy metric values for each class when combining all features together
Accuracy	0.56468	0.49868
Precision "similar"	0.5648089508127507	0.499311075781664
Precision "neutral"	0.5616968357054027	0.5236065573770492
Precision "contradiction"	0.567137169743033	0.481986265187533
Recall "similar"	0.6311630101439019	0.5556735079028072
Recall "neutral"	0.5233007209062822	0.4111740473738414
Recall "contradiction"	0.5370116518163125	0.5211331962531415

Рис. 31. Значення метрик точності логістичної регресії та класифікатора Random Forest для кожного з класів у разі об'єднання всіх ознак разом

У подальших дослідженнях вдосконалюватимемо алгоритм для досягнення коректного розпізнавання частин мови, щоб отримати кращі результати. Одним із обмежень є розпізнавання частин мови. Другим обмеженням можна вважати дещо простий алгоритм ранжування. Стосовно розпізнавання частин мови, як бачимо із результатів, деякі слова, які насправді не є власними назвами, розпізнаються як власні назви. Для розпізнавання частин мови використано бібліотеку NLTK, яка доволі базовою, тому підвищити точність можна, використавши складніші підходи до розв'язання цієї задачі, наприклад, натреновані нейронні мережі. Що стосується алгоритму ранжування, то його також побудовано на евристиці, проте, якщо власна назва трапляється у багатьох ресурсах, то вона важлива. Покращення у цій площині можна досягти, враховуючи інші характеристики власних назв, а не лише використані на вебсторінці тощо. Наприклад, можна враховувати, де саме на сторінці вжито ці ключові слова, скільки раз їх використано тощо. Під час подальших досліджень заплановано перевірити, яку точність покаже алгоритм TF-IDF для розв'язання цієї задачі. Оскільки розроблена система має обмеження ще на першому етапі збирання даних, покращуючи скрепінг, намагатимемось досягти вищої якості результатів, що, ймовірно, підвищить точність.

Висновки

Перевірка роботи запропонованої інформаційно-довідкової системи засвідчує, що завдання знаходження відповіді на запитання на основі найвищого ступеня порівняння за допомогою текстового контенту з відкритих англомовних вебресурсів вона виконує із достатньою точністю. До кожного елемента списку додається числова характеристика ймовірності переваги конкретної

відповіді над іншими. Ця метрика забезпечує ранжування отриманих результатів. Інформаційно-довідкова система забезпечує відповіді на запитання, на які немає однозначної відповіді, що вирізняє її серед класичних інформаційних систем пошуку відповідей на запитання типу QA-систем. Останні ґрунтуються на гіпотезі, що існує єдина істинна відповідь на запитання, такі системи працюють із загальновідомими фактами. Прикладними питаннями, на які вони відповідають, можуть бути, наприклад дата народження відомої людини або кількість населення певної країни. Натомість запропонована інформаційно-довідкова система відповідає на суб'єктивні запитання, наприклад, "Яка найкраща книга у жанрі фентезі?" або "Яка найкраща мова програмування?". Під час апробації системи виявлено, що деякі вебсайти блокують автоматичний скрепінг даних та вимагають увімкнути файли кукі. Покращення даних здійснюється через відфільтрування вебсторінок, щоб залишити лише сторінки із контентом англійською мовою, щоб уникнути рекомендування вебсайтів іншими мовами, якщо запит зроблено англійською або якщо у запиті є лише власні назви, які не перекладаються іншими мовами.

Список літератури

1. Aksonov D., Gozhyj A., Kalinina I., Vysotska V. (2021). Question-Answering Systems Development Based on Big Data Analysis. *Computer Sciences and Information Technologies* (CSIT): proceedings of the IEEE 16th International Conference, 22–25 Sept., Lviv, Ukraine.. P. 113–118. DOI: 10.1109/CSIT52700.2021.9648631.

2. Breja M., Jain S. (2020). Causality for Question Answering. *CEUR workshop proceedings*, Vol. 2604, 884–893.

3. Kubinska S., Holoshchuk R., Holoshchuk S., Chyrun L. (2022). Ukrainian Language Chatbot for Sentiment Analysis and User Interests Recognition based on Data Mining. *CEUR Workshop Proceedings*, Vol. 3171, 315–327.

4. Husak V., Lozynska O., Karpov I., Peleshchak I., Chyrun S., Vysotskyi A. (2020). Information System for Recommendation List Formation of Clothes Style Image Selection According to User's Needs Based on NLP and Chatbots. *CEUR workshop proceedings*, Vol. 2604, 788–818.

5. Romanovskyi O., Pidbutska N., Knysh A. (2021). Elomia Chatbot: The Effectiveness of Artificial Intelligence in the Fight for Mental Health. *CEUR Workshop Proceedings*, Vol. 2870, 1215–1224.

6. Yarovyi A., Kudriavtsev D. (2021). Method of Multi-Purpose Text Analysis Based on a Combination of Knowledge Bases for Intelligent Chatbot. *CEUR Workshop Proceedings*, Vol. 2870, 1238–1248.

7. Zdebskyi P., Lytvyn V., Burov Y., Rybchak Z., Kravets P., Lozynska O., Holoshchuk R., Kubinska S., Dmytriv A. (2020). Intelligent System for Semantically Similar Sentences Identification and Generation Based on Machine Learning Methods. *CEUR workshop proceedings*, Vol. 2604, 317–346.

8. Lytvyn V., Burov Y., Kravets P., Vysotska V., Demchuk A., Berko A., Ryshkovets Y., Shcherbak S., Naum O. (2019). Methods and Models of Intellectual Processing of Texts for Building Ontologies of Software for Medical Terms Identification in Content Classification. *CEUR Workshop Proceedings*, Vol. 2362, 354–368.

9. Vysotska V., Berko A., Lytvyn V., Kravets P., Dzyubyk L., Bardachov Y., Vyshemyrska S. (2020). Information Resource Management Technology Based on Fuzzy Logic. *Advances in Intelligent Systems and Computing*, Vol. 1246, 164–182. DOI: 10.1007/978-3-030-54215-3_11.

10. Berko A., Matseliukh Y., Ivaniv Y., Chyrun L., Schuchmann V. (2021). The text classification based on Big Data analysis for keyword definition using stemming. *Computer science and information technologies* : proceedings of IEEE 16th International conference on computer science and information technologies. Lviv, Ukraine, 22–25 September, 2021, 184–188. DOI: 10.1109/CSIT52700.2021.9648764.

11. Hladun O., Berko A., Bublyk M., Chyrun L., Schuchmann V. (2021). Intelligent system for film script formation based on artbook text and Big Data analysis. *Computer science and information technologies* : proceedings of IEEE 16th International conference on computer science and information technologies. Lviv, Ukraine, 22–25 September, 2021, 138–146. DOI: 10.1109/CSIT52700.2021.9648682.

12. Dyriv A., Andrunyk V., Burov Y., Karpov I., Chyrun L. (2021). The user's psychological state identification based on Big Data analysis for person's electronic diary. *Computer science and information technologies*: proceedings of IEEE 16th International conference on computer science and information technologies. Lviv, Ukraine, 22–25 September, 2021, 101–112. DOI: 10.1109/CSIT52700.2021.9648810.

13. Burov Y., Horodetska A., Bublyk M., Nashkerska M., Vysotska V. (2021). Tourist Service with the Situation Context Processing. *International Conference on New Trends in Languages, Literature and Social*

Communications (ICNTLLSC 2021), 2021/5/27, 233–243, DOI: 10.2991/assehr.k.210525.028.

14. Lytvyn V., Vysotska V., Peleshchak I., Basyuk T., Kovalchuk V., Kubinska S., Chyrun L., Rusyn B., Pohreliuk L., Salo T. (2019). Identifying Textual Content Based on Thematic Analysis of Similar Texts in Big Data. *Proceedings of the International Conference on Computer Sciences and Information Technologies*, CSIT, 84–91. DOI: 10.1109/STC-CSIT.2019.8929808.

15. Vysotska V., Lytvyn V., Kovalchuk V., Kubinska S., Dilai M., Rusyn B., Pohreliuk L., Chyrun L., Chyrun S., Brodyak O. (2019). Method of Similar Textual Content Selection Based on Thematic Information Retrieval. *Proceedings of the International Conference on Computer Sciences and Information Technologies*, CSIT, 2019, 1–6. DOI: 10.1109/STC-CSIT.2019.8929752.

16. Savytska L., M. Sübay T., Vnukova N., Bezugla I., Pyvovarov V. (2022). Word2Vec Model Analysis for Semantic and Morphologic Similarities in Turkish Words. *CEUR Workshop Proceedings*, Vol. 3171, 161–176.

17. Savytska L., Vnukova N., Bezugla I., Pyvovarov V., Turgut Sübay M. (2021). Using Word2vec Technique to Determine Semantic and Morphologic Similarity in Embedded Words of the Ukrainian Language. *CEUR Workshop Proceedings*, Vol. 2870, 235–248.

18. Lytvyn V. The similarity metric of scientific papers summaries on the basis of adaptive ontologies (2011). *Proceedings of 7th International Conference on Perspective Technologies and Methods in MEMS Design*, 162.

19. Dupuch M., Trinquar, L., Colombet I., Jaulent M.-C., Grabar N. (2010). Exploitation of semantic similarity for adaptation of existing terminologies within biomedical area. *CEUR Workshop Proceedings*, 673.

20. Cardon R., Grabar N. (2020). A French corpus for semantic similarity. *LREC 2020 – 12th International Conference on Language Resources and Evaluation, Conference Proceedings*, 6889–6894.

21. Elalfy D., Gad W., Ismail R. (2018). A hybrid model to predict best answers in question answering communities. *Egyptian Informatics Journal*, Vol. 19(1), 21–31. DOI: 10.1016/j.eij.2017.06.002.

22. Sahu T. P., Nagwani N. K., Verma S. (2016). Selecting Best Answer: An Empirical Analysis on Community Question Answering Sites. *IEEE Access*, Vol. 4, 4797–4808, DOI: 10.1109/ACCESS.2016.2600622.

23. Question And Answer Demo Using BERT. URL: https://www.pragnakalp.com/demos/BERT-NLP-QnA-Demo

24. Lytvyn V., Vysotska V., Rzheuskyi A. (2019). Technology for the Psychological Portraits Formation of Social Networks Users for the IT Specialists Recruitment Based on Big Five, NLP and Big Data Analysis. *CEUR Workshop Proceedings*, Vol. 2392, 147–171.

25. Shu C., Dosyn D., Lytvyn V., Vysotska V., Sachenko A., Jun S. (2019). Building of the Predicate Recognition System for the NLP Ontology Learning Module. *Proceedings of the International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, IDAACS, 2, 802–808. DOI: 10.1109/IDAACS.2019.8924410.

26. Oliinyk V.-A., Vysotska V., Burov Y., Mykich K., Basto-Fernandes V. (2020). Propaganda Detection in Text Data Based on NLP and Machine Learning. *CEUR workshop proceedings*, Vol. 2631, 132–144.

27. Balush I., Vysotska V., Albota S. (2021). Recommendation System Development Based on Intelligent Search, NLP and Machine Learning Methods. *CEUR Workshop Proceedings*, Vol. 2917, 584–617.

28. Batiuk T., Vysotska V., Holoshchuk R., Holoshchuk S. (2022). Intelligent System for Socialization of Individual's with Shared Interests based on NLP, Machine Learning and SEO Technologies. *CEUR Workshop Proceedings*, Vol. 3171, 572–631.

29. Deriviere J., Hamon T., Nazarenko A. (2006). A scalable and distributed NLP architecture for web document annotation. *Lecture Notes in Computer Science*, Vol. 4139, 56–67. DOI: 10.1007/11816508_8.

30. Boyè M., Tran T. M., Grabar N. (2014). NLP-oriented contrastive study of linguistic productions of alzheimer's and control people. *Lecture Notes in Computer Science*, Vol. 8686, 412–424. DOI: 10.1007/978-3-319-10888-9_41.

31. Lytvyn V., Vysotska V., Budz I., Pelekh Y., Sokulska N., Kovalchuk R., Dzyubyk L., Tereshchuk O., Komar M. (2019). Development of the quantitative method for automated text content authorship attribution based on the statistical analysis of N-grams distribution. *Eastern-European Journal of Enterprise Technologies*, Vol. 6(2-102), 28–51. DOI: 10.15587/1729-4061.2019.186834.

32. Vysotska V., Markiv O., Teslia S., Romanova Y., Pihulechko I. (2022). Correlation Analysis of Text Author Identification Results Based on N-Grams Frequency Distribution in Ukrainian Scientific and Technical Articles. *CEUR Workshop Proceedings*, Vol. 3171, 277–314.

33. Boyer C., Dolamic L., Grabar N. (2015). Automated Detection of Health Websites' HONcode Conformity: Can N-gram Tokenization Replace Stemming? *Studies in Health Technology and Informatics*, Vol. 216, 1064.

34. Lytvyn V., Burov Y., Vysotska V., Pukach Y., Tereshchuk O., Shakleina I. (2021). Abstracting Text Content Based on Weighing the TF-IDF Measure by the Subject Area Ontology. *International Conference on Smart Information Systems and Technologies* (SIST), Nur-Sultan, Kazakhstan. DOI: 10.1109/SIST50301.2021.9465978.

35. Das M., Kamalanathan S., Alphonse P. J. A. (2021). A Comparative Study on TF-IDF Feature Weighting Method and Its Analysis Using Unstructured Dataset. *CEUR Workshop Proceedings*, Vol. 2870, 98–107.

36. Lande D., Dmytrenko O. (2021). Using Part-of-Speech Tagging for Building Networks of Terms in Legal Sphere. *CEUR Workshop Proceedings*, Vol. 2870, 87–97.

37. Hrytsiv N., Bekhta I., Tkachivska M., Byalyk V. (2022). Sylvia Plath's I felt-Narrative Label of The Bell Jar in Ukrainian Translation: Tagging Textness Features. *CEUR Workshop Proceedings*, Vol. 3171, 240–255.

38. Mukalov P., Zelinskyi O., Levkovych R., Tarnavskyi P., Pylyp A., Shakhovska N. (2019). Development of System for Auto-Tagging Articles, Based on Neural Network. *CEUR Workshop Proceedings*, Vol. 2362, 106–115.

39. Shakhovska N., Basystiuk O., Shakhovska K. (2019). Development of the Speech-to-Text Chatbot Interface Based on Google API. *CEUR Workshop Proceedings*, Vol. 2386, 212–221.

40. Hlavcheva Y., Kanishcheva O., Vovk M., Glavchev M. (2021). Identification of the Author's Idea Based on the Modified TextRank Method. *CEUR Workshop Proceedings*, Vol. 2870, 118–128.

41. Lytvyn V., Vysotska V., Dosyn D., Burov Y. (2018). Method for ontology content and structure optimization, provided by a weighted conceptual graph. *Webology*, Vol. 15(2), 66–85.

42. Batiuk T., Chyrun L., Oborska O. (2022). Ontology Model and Ontological Graph for Development of Decision Support System of Personal Socialization by Common Relevant Interests. *CEUR Workshop Proceedings*, Vol. 3171, 877–903.

43. Petrenjuk V., Petrenjuk D. (2022). Application Trend through Planar 3-minimal & Projective Planar 2-minimal Graphs. *CEUR Workshop Proceedings*, Vol. 3171, 1737–1747.

44. Petrenjuk V. (2020). About φ-Transformation Graphs as a Tool for Investigations. *CEUR workshop proceedings*, Vol. 2604, 1309–1319.

45. Lytvyn V., Uhryn D., Fityo A. (2016). Modeling of territorial community formation as a graph partitioning problem. *Eastern-European Journal of Enterprise Technologies*, Vol. 1(4), 47–52. DOI: 10.15587/1729-4061.2016.60848.

46. Meleshko Y., Yakymenko M., Semenov S. (2021). A Method of Detecting Bot Networks Based on Graph Clustering in the Recommendation System of Social Network. *CEUR Workshop Proceedings*, Vol. 2870, 1249–1261.

47. Learning Semantic Textual Similarity from Conversations (2022). URL: https://uk.wikipedia.org/wiki/.

48. TensorFlow. Universal Sentence Encoder, 2022. URL: https://www.tensorflow.org/hub/tutorials/semantic_similarity_with_tf_hub_universal_encoder

49. Huilgol P. (2022). Top 4 Sentence Embedding Techniques using Python! URL: https://www.analyticsvidhya.com/blog/2020/08/top-4-sentence-embedding-techniques-using-python/

50. Neubig G. (2022). Pre-trained Sentence and Contextualized Word Representations. URL: http://www.phontron.com/class/nn4nlp2021/assets/slides/nn4nlp-09-sentrep.pdf

51. Add Quora Question Triplets Dataset (2022). URL: https://github.com/huggingface/datasets/issues/4654

52. The Multi-Genre NLI Corpus (2022). URL: https://cims.nyu.edu/~sbowman/multinli/

References

1. Aksonov D., Gozhyj A., Kalinina I., Vysotska V. (2021). Question-Answering Systems Development Based on Big Data Analysis. *Computer Sciences and Information Technologies* (CSIT): proceedings of the IEEE 16th International Conference, 22–25 Sept., Lviv, Ukraine, 113–118. DOI: 10.1109/CSIT52700.2021.9648631.

2. Breja M., Jain S. (2020). Causality for Question Answering. *CEUR workshop proceedings*, Vol. 2604, 884–893.

3. Kubinska S., Holoshchuk R., Holoshchuk S., Chyrun L. (2022). Ukrainian Language Chatbot for Sentiment Analysis and User Interests Recognition based on Data Mining. *CEUR Workshop Proceedings*, Vol. 3171, 315–327.

4. Husak V., Lozynska O., Karpov I., Peleshchak I., Chyrun S., Vysotskyi A. (2020). Information System for Recommendation List Formation of Clothes Style Image Selection According to User's Needs Based on NLP and Chatbots. *CEUR workshop proceedings*, Vol. 2604, 788–818.

5. Romanovskyi O., Pidbutska N., Knysh A. (2021). Elomia Chatbot: The Effectiveness of Artificial Intelligence in the Fight for Mental Health. *CEUR Workshop Proceedings*, Vol. 2870, 1215–1224.

6. Yarovyi A., Kudriavtsev D. (2021). Method of Multi-Purpose Text Analysis Based on a Combination of

Knowledge Bases for Intelligent Chatbot. *CEUR Workshop Proceedings*, Vol. 2870, 1238–1248.

7. Zdebskyi P., Lytvyn V., Burov Y., Rybchak Z., Kravets P., Lozynska O., Holoshchuk R., Kubinska S., Dmytriv A. (2020). Intelligent System for Semantically Similar Sentences Identification and Generation Based on Machine Learning Methods. *CEUR workshop proceedings*, Vol. 2604, 317–346.

8. Lytvyn V., Burov Y., Kravets P., Vysotska V., Demchuk A., Berko A., Ryshkovets Y., Shcherbak S., Naum O. (2019). Methods and Models of Intellectual Processing of Texts for Building Ontologies of Software for Medical Terms Identification in Content Classification. *CEUR Workshop Proceedings*, Vol. 2362, 354–368.

9. Vysotska V., Berko A., Lytvyn V., Kravets P., Dzyubyk L., Bardachov Y., Vyshemyrska S. (2020). Information Resource Management Technology Based on Fuzzy Logic. *Advances in Intelligent Systems and Computing*, Vol. 1246, 164–182. DOI: 10.1007/978-3-030-54215-3_11.

10. Berko A., Matseliukh Y., Ivaniv Y., Chyrun L., Schuchmann V. (2021). The text classification based on Big Data analysis for keyword definition using stemming. *Computer science and information technologies*: proceedings of IEEE 16th International conference on computer science and information technologies. Lviv, Ukraine, 22–25 September, 2021, 184–188. DOI: 10.1109/CSIT52700.2021.9648764.

11. Hladun O., Berko A., Bublyk M., Chyrun L., Schuchmann V. (2021). Intelligent system for film script formation based on artbook text and Big Data analysis. *Computer science and information technologies*: proceedings of IEEE 16th International conference on computer science and information technologies. Lviv, Ukraine, 22–25 September, 2021, 138–146. DOI: 10.1109/CSIT52700.2021.9648682.

12. Dyriv A., Andrunyk V., Burov Y., Karpov I., Chyrun L. (2021). The user's psychological state identification based on Big Data analysis for person's electronic diary. *Computer science and information technologies*: proceedings of IEEE 16th International conference on computer science and information technologies. Lviv, Ukraine, 22–25 September, 2021, 101–112. DOI: 10.1109/CSIT52700.2021.9648810.

13. Burov Y., Horodetska A., Bublyk M., Nashkerska M., Vysotska V. (2021). Tourist Service with the Situation Context Processing. *International Conference on New Trends in Languages, Literature and Social Communications* (ICNTLLSC 2021), 2021/5/27, 233–243. DOI: 10.2991/assehr.k.210525.028.

14. Lytvyn V., Vysotska V., Peleshchak I., Basyuk T., Kovalchuk V., Kubinska S., Chyrun L., Rusyn B., Pohreliuk L., Salo T. (2019). Identifying Textual Content Based on Thematic Analysis of Similar Texts in Big Data. *Proceedings of the International Conference on Computer Sciences and Information Technologies*, CSIT, 84–91. DOI: 10.1109/STC-CSIT.2019.8929808.

15. Vysotska V., Lytvyn V., Kovalchuk V., Kubinska S., Dilai M., Rusyn B., Pohreliuk L., Chyrun L., Chyrun S., Brodyak O. (2019). Method of Similar Textual Content Selection Based on Thematic Information Retrieval. *Proceedings of the International Conference on Computer Sciences and Information Technologies*, CSIT, 2019, 1–6. DOI: 10.1109/STC-CSIT.2019.8929752.

16. Savytska L., M. Sübay T., Vnukova N., Bezugla I., Pyvovarov V. (2022). Word2Vec Model Analysis for Semantic and Morphologic Similarities in Turkish Words. *CEUR Workshop Proceedings*, Vol. 3171, 161–176.

17. Savytska L., Vnukova N., Bezugla I., Pyvovarov V., Turgut Sübay M. (2021). Using Word2vec Technique to Determine Semantic and Morphologic Similarity in Embedded Words of the Ukrainian Language. *CEUR Workshop Proceedings*, Vol. 2870, 235–248.

18. Lytvyn V. The similarity metric of scientific papers summaries on the basis of adaptive ontologies (2011). *Proceedings of 7th International Conference on Perspective Technologies and Methods in MEMS Design*, 162.

19. Dupuch M., Trinquar, L., Colombet I., Jaulent M.-C., Grabar N. (2010). Exploitation of semantic similarity for adaptation of existing terminologies within biomedical area. *CEUR Workshop Proceedings*, 673.

20. Cardon R., Grabar N. (2020). A French corpus for semantic similarity. *LREC 2020 – 12th International Conference on Language Resources and Evaluation, Conference Proceedings*, 6889–6894.

21. Elalfy D., Gad W., Ismail R. (2018). A hybrid model to predict best answers in question answering communities. *Egyptian Informatics Journal*, Vol. 19(1), 21–31. DOI: 10.1016/j.eij.2017.06.002.

22. Sahu T. P., Nagwani N. K., Verma S. (2016). Selecting Best Answer: An Empirical Analysis on Community Question Answering Sites. *IEEE Access*, Vol. 4, 4797-4808. DOI: 10.1109/ACCESS.2016.2600622.

23. Question And Answer Demo Using BERT. URL: https://www.pragnakalp.com/demos/BERT-NLP-QnA-Demo.

24. Lytvyn V., Vysotska V., Rzheuskyi A. (2019). Technology for the Psychological Portraits Formation of Social Networks Users for the IT Specialists Recruitment Based on Big Five, NLP and Big Data Analysis. *CEUR Workshop Proceedings*, Vol. 2392, 147–171.

25. Shu C., Dosyn D., Lytvyn V., Vysotska V., Sachenko A., Jun S. (2019). Building of the Predicate

Recognition System for the NLP Ontology Learning Module. *Proceedings of the International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, IDAACS, 2, 802–808. DOI: 10.1109/IDAACS.2019.8924410.

26. Oliinyk V.-A., Vysotska V., Burov Y., Mykich K., Basto-Fernandes V. (2020). Propaganda Detection in Text Data Based on NLP and Machine Learning. *CEUR workshop proceedings*, Vol. 2631, 132–144.

27. Balush I., Vysotska V., Albota S. (2021). Recommendation System Development Based on Intelligent Search, NLP and Machine Learning Methods. *CEUR Workshop Proceedings*, Vol. 2917, 584–617.

28. Batiuk T., Vysotska V., Holoshchuk R., Holoshchuk S. (2022). Intelligent System for Socialization of Individual's with Shared Interests based on NLP, Machine Learning and SEO Technologies. *CEUR Workshop Proceedings*, Vol. 3171, 572–631.

29. Deriviere J., Hamon T., Nazarenko A. (2006). A scalable and distributed NLP architecture for web document annotation. *Lecture Notes in Computer Science*, Vol. 4139, 56–67. DOI: 10.1007/11816508_8.

30. Boyè M., Tran T.M., Grabar N. (2014). NLP-oriented contrastive study of linguistic productions of alzheimer's and control people. *Lecture Notes in Computer Science*, Vol. 8686, 412–424. DOI: 10.1007/978-3-319-10888-9_41.

31. Lytvyn V., Vysotska V., Budz I., Pelekh Y., Sokulska N., Kovalchuk R., Dzyubyk L., Tereshchuk O., Komar M. (2019). Development of the quantitative method for automated text content authorship attribution based on the statistical analysis of N-grams distribution. *Eastern-European Journal of Enterprise Technologies*, Vol. 6(2-102), 28–51. DOI: 10.15587/1729-4061.2019.186834.

32. Vysotska V., Markiv O., Teslia S., Romanova Y., Pihulechko I. (2022). Correlation Analysis of Text Author Identification Results Based on N-Grams Frequency Distribution in Ukrainian Scientific and Technical Articles. *CEUR Workshop Proceedings*, Vol. 3171, 277–314.

33. Boyer C., Dolamic L., Grabar N. (2015). Automated Detection of Health Websites' HONcode Conformity: Can N-gram Tokenization Replace Stemming? *Studies in Health Technology and Informatics*, Vol. 216, 1064.

34. Lytvyn V., Burov Y., Vysotska V., Pukach Y., Tereshchuk O., Shakleina I. (2021). Abstracting Text Content Based on Weighing the TF-IDF Measure by the Subject Area Ontology. *International Conference on Smart Information Systems and Technologies* (SIST), Nur-Sultan, Kazakhstan. DOI: 10.1109/SIST50301.2021.9465978.

35. Das M., Kamalanathan S., Alphonse P.J.A. (2021). A Comparative Study on TF-IDF Feature Weighting Method and Its Analysis Using Unstructured Dataset. *CEUR Workshop Proceedings*, Vol. 2870, 98–107.

36. Lande D., Dmytrenko O. (2021). Using Part-of-Speech Tagging for Building Networks of Terms in Legal Sphere. *CEUR Workshop Proceedings*, Vol. 2870, 87–97.

37. Hrytsiv N., Bekhta I., Tkachivska M., Byalyk V. (2022). Sylvia Plath's I felt-Narrative Label of The Bell Jar in Ukrainian Translation: Tagging Textness Features. *CEUR Workshop Proceedings*, Vol. 3171, 240–255.

38. Mukalov P., Zelinskyi O., Levkovych R., Tarnavskyi P., Pylyp A., Shakhovska N. (2019). Development of System for Auto-Tagging Articles, Based on Neural Network. *CEUR Workshop Proceedings*, Vol. 2362, 106–115.

39. Shakhovska N., Basystiuk O., Shakhovska K. (2019). Development of the Speech-to-Text Chatbot Interface Based on Google API. *CEUR Workshop Proceedings*, Vol. 2386, 212–221.

40. Hlavcheva Y., Kanishcheva O., Vovk M., Glavchev M. (2021). Identification of the Author's Idea Based on the Modified TextRank Method. *CEUR Workshop Proceedings*, Vol. 2870, 118–128.

41. Lytvyn V., Vysotska V., Dosyn D., Burov Y. (2018). Method for ontology content and structure optimization, provided by a weighted conceptual graph. *Webology*, Vol. 15(2), 66–85.

42. Batiuk T., Chyrun L., Oborska O. (2022). Ontology Model and Ontological Graph for Development of Decision Support System of Personal Socialization by Common Relevant Interests. *CEUR Workshop Proceedings*, Vol. 3171, 877–903.

43. Petrenjuk V., Petrenjuk D. (2022). Application Trend through Planar 3-minimal & Projective Planar 2-minimal Graphs. *CEUR Workshop Proceedings*, Vol. 3171, 1737–1747.

44. Petrenjuk V. (2020). About φ-Transformation Graphs as a Tool for Investigations. *CEUR workshop proceedings*, Vol. 2604, 1309–1319.

45. Lytvyn V., Uhryn D., Fityo A. (2016). Modeling of territorial community formation as a graph partitioning problem. *Eastern-European Journal of Enterprise Technologies*, Vol. 1(4), 47–52. DOI: 10.15587/1729-4061.2016.60848.

46. Meleshko Y., Yakymenko M., Semenov S. (2021). A Method of Detecting Bot Networks Based on Graph Clustering in the Recommendation System of Social Network. *CEUR Workshop Proceedings*, Vol. 2870, 1249–1261.

47. Learning Semantic Textual Similarity from Conversations (2022). URL: https://uk.wikipedia.org/wiki/.
48. TensorFlow. Universal Sentence Encoder (2022). URL: https://www.tensorflow.org/hub/tutorials/semantic_similarity_with_tf_hub_universal_encoder
49. Huilgol P. (2022). Top 4 Sentence Embedding Techniques using Python! URL: https://www.analyticsvidhya.com/blog/2020/08/top-4-sentence-embedding-techniques-using-python/
50. Neubig G. (2022). Pre-trained Sentence and Contextualized Word Representations. URL: http://www.phontron.com/class/nn4nlp2021/assets/slides/nn4nlp-09-sentrep.pdf
51. Add Quora Question Triplets Dataset (2022). URL: https://github.com/huggingface/datasets/issues/4654
52. The Multi-Genre NLI Corpus (2022). URL: https://cims.nyu.edu/~sbowman/multinli/

INFORMATION SYSTEM FOR EXTRACTION OF INFORMATION FROM OPEN WEB RESOURCES

Petro Zdebskyi[1], Andrii Berko[1], Lyubomyr Chyrun[2]

[1] Lviv Polytechnic National University, Information Systems and Networks Department,
12, S. Bandera str., Lviv, Ukraine
[2] Ivan Franko National University of Lviv, Applied Mathematics Department,
1, University str., Lviv, Ukraine
E-mail: petro.v.zdebskyi@lpnu.ua, ORCID: 0000-0002-0478-2308
E-mail: Andrii.Y.Berko@lpnu.ua, ORCID: 0000-0003-2892-9519
E-mail: Lyubomyr.Chyrun@lnu.edu.ua, ORCID: 0000-0002- 9448-1751

The purpose of the work is to develop a project of an information and reference system for finding answers to questions based on the highest degree of comparison using text content from open English-language web resources. Examples of such questions can be: "What is the best book ever?", "What is the most popular IDE for Python". The result of the functioning of the information and reference system is a ranked list of answers based on the frequency of appearance of each of the answer options. Also, a numerical characteristic of the probability of the preference of a particular answer over others is added to each element of the list. Based on this metric, the obtained results are ranked. This information and reference system works with questions to which there is no unequivocal answer, what differs it from classic information systems for finding answers to questions of the QA-system type. The latter have a hypothesis that there is only one true answer to the question, often such systems work with well-known facts. Examples of questions they answer can be, for example, the date of birth of a famous person, or the population of a certain country. Instead, the proposed information and reference system answers subjective questions, for example, "What is the best book in the fantasy genre?" or "What is the best programming language?". The system is based on the popularity of one or another answer. Proper names based on the analysis of N-grams are also keywords for forming the answer to the question.

Key words: information system; project; QA system; web application; content search; similarity of text fragments; Part-of-speech tagging; N-gram; TF-IDF; TextRank.