

Костянтин Коломієць, Мар'ян Опришко*, Андрій Нечипуренко*, Юрій Прусак*

магістрант кафедри КН НЛТУ України, Львів, Україна.
k.kolomiets@gmail.com.

*асистент кафедри ІПЗ, Національний лісотехнічний університет України,
Львів, Україна. <https://orcid.org/0009-0009-7981-9249>
m.opryshko@nltu.edu.ua.

*старший викладач кафедри ІПЗ, НЛТУ України, Львів, Україна.
<https://orcid.org/0000-0003-3955-1849> andriy.nechepurenko@gmail.com.

*доцент кафедри КН, НЛТУ України, Львів, Україна. <https://orcid.org/0000-0003-3349-4468> y.prusak@nltu.edu.ua.

Інтелектуальна система пошуку фільмів засобами Angular

Анотація – У цьому дослідженні розглянуто інтелектуальний алгоритм пошуку фільмів з використанням Angular, який дозволяє користувачам швидко та ефективно знаходити фільми за їх коротким описом використовуючи алгоритми обробки природної мови. Система використовує дані з The Movie DB і представляє собою важливий крок у розробці інтелектуальних веб-застосунків. У роботі детально описані засоби програмного забезпечення, використані під час розробки, включаючи Visual Studio Code, Node.js, Angular CLI, NPM і Git. Розроблена система демонструє ефективність Angular у створенні високоефективних веб-застосунків з інтелектуальними функціями пошуку.

Ключові слова – Angular, інтелектуальна система, пошук фільмів, The Movie DB, Natural Language Processing.

Angular є однією з найпопулярніших платформ для розробки веб-застосунків, і вона може бути успішно використана для створення інтелектуальних систем [1]. У цій роботі розглянуто процес створення інтелектуальної системи пошуку фільмів з використанням Angular. Система надає можливість користувачам здійснювати пошук фільмів за їх приблизним описом і відображати результати на веб-сторінці.

Angular є основною технологією, використаною для створення інтерфейсу користувача цієї системи, що надає широкі можливості для розробки високоефективних інтерактивних веб-застосунків і забезпечує простоту управління структурою проекту та компонентами [1]. Для реалізації серверної частини сис-

теми використано `node.js` та `express` [2, 4]. `Node.js` дозволяє виконувати JavaScript на стороні сервера, а `express` надає можливість створювати RESTful API для обробки запитів. The Movie DB виступає як джерело даних для системи. Він містить велику кількість інформації про фільми, включаючи назви, описи, рейтинги та багато іншого [3]. Під час розробки системи також використано такі засоби, як Visual Studio Code для редагування коду, Iterm2 для роботи з терміналом, NPM для управління пакетами, Angular CLI для швидкого створення компонентів та Git для контролю версій коду.

Досліджуваний алгоритм розроблений для ефективного пошуку фільмів за їх описом, враховуючи особливості формулювань користувача. Він використовує Angular на клієнтській стороні та `node.js` на серверній стороні для взаємодії з базою даних The Movie DB. Робота алгоритму передбачає виконання таких кроків:

Крок 1: Введення інтересів користувача. Користувач вводить свої інтереси, ключові слова або фрази, які описують те, що його цікавить у фільмах. Це можуть бути слова, пов'язані з жанрами, темами, атмосферою фільмів тощо.

Крок 2: Відправка запиту на сервер. Після введення інтересів, користувач натискає кнопку пошуку, і на клієнтській стороні генерується HTTP-запит до сервера. Цей запит містить введені користувачем інтереси для пошуку.

Крок 3: Обробка запиту на сервері. Сервер, який працює на базі `node.js` і використовує `express`, отримує HTTP-запит від клієнта. Сервер розпочинає обробку запиту та взаємодіє з базою даних The Movie DB для отримання фільмів, які мають описи, що відповідають інтересам користувача.

Крок 4: Аналіз описів фільмів. Отримавши фільми з бази даних, сервер аналізує описи кожного фільму, враховуючи інтереси користувача. Для цього використовуються алгоритми обробки природної мови, які визначають співпадіння між інтересами користувача і текстом опису фільму.

Крок 5: Повернення результатів на клієнтську сторону. Після аналізу описів фільмів, сервер відправляє список фільмів, які найкраще відповідають

інтересам користувача, на клієнтську сторону. Angular відповідає за відображення цих результатів на веб-сторінці у зручному форматі для користувача.

Крок 6: Перегляд деталей фільму. Користувач може обрати конкретний фільм і переглянути деталі, такі як опис, рейтинг, актори тощо. Для цього використовується запит до сервера за допомогою ідентифікатора фільму, отриманого з результатів пошуку.

У системі інтелектуального пошуку фільмів використовується алгоритм обробки природної мови (Natural Language Processing, NLP), який допомагає визначити співпадіння між інтересами користувача і текстом опису фільму [5]. Цей алгоритм дозволяє більш точно і персоналізовано знаходити фільми, які відповідають вподобанням користувача.

Опис алгоритму NLP:

Крок 1: Токенізація. Перший крок алгоритму – це токенизація, в якій текст опису фільму розбивається на окремі слова або токени. Це може включати в себе розділення тексту на окремі слова, видалення пунктуації та інших символів.

Крок 2: Векторизація. Після токенизації кожне слово перетворюється у вектор, що представляє його значення. Цей вектор створений за допомогою векторної моделі слів Word2Vec. Це дозволяє представити текст опису фільму у вигляді числового вектора.

Крок 3: Визначення інтересів користувача. Інтереси користувача, введені на попередньому кроці, також перетворюються у вектор, що представляє їх значення. Для цього використовуються та ж сама векторна модель слів.

Крок 4: Обчислення схожості. За допомогою обчислення косинусної схожості між векторами інтересів користувача та тексту опису фільму визначається ступінь відповідності. Косинусна схожість дає відомості про те, наскільки близькі є два вектори в просторі слів.

Крок 5: Ранжування фільмів. Фільми ранжуються в порядку спадання схожості з інтересами користувача. Тобто фільми, які мають найбільшу схожість з інтересами користувача, розміщуються на перших позиціях у списку результатів пошуку.

Висновки. Розроблений інтелектуальний алгоритм пошуку фільмів за описом дозволяє користувачам знаходити фільми, назва яких є невідома. Використання Angular на клієнтській стороні та node.js на серверній стороні дозволили реалізувати цей алгоритм. Розглянутий підхід робить пошук фільмів більш інтелектуальним та персоналізованим для кожного користувача. Алгоритм обробки природної мови (NLP) в системі пошуку фільмів за описом допомагає робити пошук більш точним та зручним. Використовуючи цей алгоритм, система враховує інтереси користувача та знаходить фільми, які найкраще відповідають їхнім вподобанням, враховуючи зміст введеного тексту опису фільмів.

Список використаних літературних джерел

- [1] Документація Angular: <https://angular.io/docs>
- [2] Документація Express.js: <https://expressjs.com/>
- [3] Документація The Movie Database API: <https://developers.themoviedb.org/3/>
- [4] Документація Node.js: <https://nodejs.org/en/docs/>
- [5] Cambria E, White B (2014) Jumping NLP curves: a review of natural language processing research. IEEE Computat Intell Mag 5:48–57

Юрій Паславський, Ігор Крошний*

аспірант кафедри КН НЛТУ України, Львів, Україна.

yuriy.paslavsky@nltu.edu.ua

*доцент кафедри КН НЛТУ України, Львів, Україна.

kroshny.igor@nltu.edu.ua

Методи використання алгоритмів неінтерактивних аргументів знання для ідентифікації користувача

Анотація - З розвитком мережових технологій надзвичайно актуально постало завдання підвищення ефективності і захищеності схем ідентифікації з одночасним дотриманням правил безпеки, анонімності і конфіденційності. Стратегічним напрямком дій у цій сфері є запровадження алгоритмів, що мають підвищену криптографічну складність та потребують суттєвих часових витрат задля їх зламу. Для розв'язання цих завдань використовуються основні