

Костянтин Олександрович ТКАЧЕНКО^{1,2},

кандидат економічних наук, доцент, доцент кафедри інженерії програмного забезпечення,
доцент кафедри інформаційних технологій,
ORCID ID: <https://orcid.org/0000-0003-0543-3396>

Ольга Іванівна ТКАЧЕНКО²,

кандидат фізико-математичних наук, доцент, доцент, кафедра інформаційних технологій,
ORCID ID: <https://orcid.org/0000-0003-1800-618X>

Андрій Ярославович КАТРОШЕНКО²,

магістрант кафедри інформаційних технологій,
ORCID ID: <https://orcid.org/0009-0003-0124-7582>

¹ Національний університет «Київський авіаційний інститут»

² Національний транспортний університет

Прийняття: 15/06/2026

Рецензія: 18/06/2026

Публікація: 09/07/2026

DOI: <https://doi.org/10.53920/ITS-2026-1-7>

РОЗРОБЛЕННЯ СИСТЕМИ ПЕРСОНАЛІЗОВАНИХ РЕКОМЕНДАЦІЙ ЩОДО РЕКРУТИНГУ ІТ-ПЕРСОНАЛУ: ОНТОЛОГІЧНИЙ ПІДХІД

УДК

004.4; 004.6; 004.9



This is an Open Access
article distributed
under the terms
of the [Creative Commons
CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

© Ткаченко К.О.,
Ткаченко О.І.,
Катрошенко А.Я.,
2026

В наш час ринок фахівців в ІТ-галузі характеризується високою динамікою до змін та розвитком нових ІТ-технологій, мов програмування, фреймворків тощо. За таких умов ефективний процес рекрутингу ІТ-фахівців стає важливим фактором успішної роботи будь-якої компанії взагалі, та ІТ-компаній зокрема. Всі компанії висувають до ІТ-фахівців відповідні вимоги щодо рівня їхніх професійних компетенцій.

У роботі розглянута проблема підбору ІТ-персоналу за допомогою існуючих в наш час платформ для пошуку роботи в умовах постійного розвитку та змін ІТ-технологій. Ці платформи мають певні обмеження щодо способу пошуку ІТ-фахівців за ключовими словами, що обумовлено наявністю синонімії та відсутністю можливості врахування реальних зв'язків між компетенціями кандидатів на посади та відповідними вимогами роботодавців. Для усунення цих обмежень в роботі запропоновано систему персоналізованих рекомендацій для ІТ-фахівців, яка використовує онтологічний підхід для формалізації та розуміння семантики текстів резюме кандидатів з інформацією про їхні професійні компетенції.

ISSN 2786-7226 (print)
ISSN 2786-7234 (online)

В систему інтегровано інтелектуальний агент для більш глибокого лінгвістичного (з урахуванням контексту) аналізу резюме кандидатів та вимог щодо вакансій, текст яких є неструктурованим. Такий підхід сприяє підвищенню рівня точності метчингу текстів щодо компетенцій кандидатів з тими компетенціями, що зазначив роботодавець в тексті про наявну вакансію. Запропонований підхід дозволить пропонувати кандидатам обґрунтовані персоналізовані рекомендації про вакансії з порадами щодо розвитку кар'єри та спростить пошук працівників, які відповідають вимогам роботодавців щодо відповідної вакансії (посади).

Ключові слова: онтологія, онтологічне моделювання, великі мовні моделі, семантичний метчинг, ШІ-агент, рекрутинг, рекомендаційна система.

Kostiantyn TKACHENKO, Olha TKACHENKO, Andriy KATROSHENKO

DEVELOPMENT OF SYSTEM OF PERSONALIZED RECOMMENDATIONS FOR IT STAFF RECRUITMENT: AN ONTOLOGICAL APPROACH

Currently, the IT specialist market is characterized by a highly dynamic environment characterized by changes and the development of new IT technologies, programming languages, frameworks, etc. In this environment, an effective IT recruitment process is becoming a crucial factor in the success of any company in general, and IT companies in particular. All companies impose appropriate requirements on IT specialists in terms of their professional competencies.

This paper examines the problem of IT personnel recruitment using currently available job search platforms in the context of the constant development and change of IT technologies. These platforms have certain limitations in their ability to search for IT specialists using keywords, due to the presence of synonyms and the inability to account for real connections between candidate competencies and the corresponding employer requirements. To address these limitations, this paper proposes a personalized recommendation system for IT specialists. This system uses an ontological approach to formalize and understand the semantics of candidate resumes and information about their professional competencies.

The system integrates an intelligent agent for deeper linguistic (context-aware) analysis of candidate resumes and job postings, even though the text is unstructured. This approach improves the accuracy of matching candidate competency texts with the competencies noted by employers in the job posting. The proposed approach will offer candidates informed personalized job posting recommendations with career development advice and simplify the search for employees who meet employer requirements for the relevant job posting.

Keywords: *ontology, ontological modeling, large-scale language models, semantic matching, AI agent, recruiting, recommendation system.*

Постановка проблеми. Сучасний стан ринку ІТ-фахівців потребує нового підходу до пошуку та відбору серед багатьох кандидатів на вакантні посади саме тих, чії фахові компетенції відповідають тим вимогам, які висуваються компаніями-роботодавцями.

Але ті традиційні методи автоматизованого пошуку кандидатів/вакансій на більшості популярних платформ все частіше потребують свого розвитку (а інколи й повної заміни на інші) через недостатню гнучкість таких платформ при вирішенні наявних проблем підбору кандидатів та вакансій [6].

Серед основних проблем багатьох сучасних систем електронного рекрутингу (e-recruitment) слід виділити те, що вони ґрунтуються на лексичному аналізі текстів, здійснюючи повнотекстовий пошук за ключовими словами [12]. Недоліком такого підходу є, зокрема, проблема синонімії, коли однакові/споріднені знання та компетенції кандидатів (чи вимоги щодо відповідних вакансій) описуються різними термінами (наприклад, **React**, **React.js**, **ReactJS**) [6, 8].

Серед наявних проблем сучасних систем e-recruitment слід виділити також використання так званих «пласких» реляційних баз даних (БД), для яких характерною властивістю є відсутність ієрархічності компетенцій та їхньої ізоморфності, через що ці системи не працюють з логічними зв'язками між технологіями та компетенціями, що відповідають однаковим концепціям чи змісту/семантиці [17]. Все це обумовлює неточне розуміння контексту резюме кандидата та вимог щодо відповідної вакансії, бо так звана «класична» система не відслідковує деякі зв'язки, зокрема такі, як володіння певним інструментом чи фреймворком, наприклад, **FastAP**, автоматично гарантує знання його базової мови програмування **Python** [19].

Ця проблема може бути вирішена за допомогою переходу від звичайного синтаксичного аналізу текстів до більш складного семантичного аналізу [1, 16]. Суть семантичного пошуку у тому, що він базується на розумінні контексту змісту та певних концептів за допомогою онтологічного моделювання предметної області [5]. Онтологія дозволяє програмі формалізувати накопичені знання у вигляді взаємопов'язаних класів, властивостей і зв'язків [2, 5]. Використання такого підходу та поєднання з можливостями сучасних інтелектуальних агентів (ШІ-агентів) з обробки природної мови сприяє не тільки більш ефективному пошуку ІТ-фахівців, а й перетворює систему пошуку роботи/персоналу на інтелектуального помічника для тих, хто шукає роботу [11, 13].

Серед проблем сучасних систем е-recruitment є так званий «семантичний розрив» між вимогами роботодавців до компетенцій кандидатів, що сформульовані у об'явах про наявні вакансії в їхніх компаніях, та тим, які компетенції висвітлені у резюме кандидатів [12]. Більшість існуючих платформ та систем е-recruitment (наприклад, таких як Work.ua [21], Djinni.co [22]) використовують повнотекстове індексування чи жорстко фіксовані теги. Це є одним з факторів, який обумовлює відсутність повного охоплення компетенцій (в резюме кандидатів та об'явах щодо вакансій) при автоматизації процесу метчингу [6].

Серед факторів, які ускладнюють процес автоматизації процесу е-recruitment, слід виділити, зокрема, такі:

- Відсутність розуміння синонімії та полісемії, які використовуються IT-фахівцями; наприклад, програмісти часто використовують різні аббревіатури або варіанти написання одного й того ж поняття IT-сфери (наприклад, C#, CSharp та .NET є синонімічними за своєю семантикою); а алгоритми пошуку комп'ютерної лінгвістики сприймають ці поняття як різні та ніяк не пов'язані між собою, що обумовлює появу ситуації, коли не знаходяться кандидати, навіть якщо вони мають релевантний досвід [17].

- Відсутність ієрархічності та спадковості знань у реляційних БД, що використовуються в більшості сучасних систем е-recruitment (наприклад, такі компетенції, як знання та досвід практичного використання технологічних інструментів JavaScript, React та Vue.js, є незалежними записами). Це обумовлює нездатність системи «зрозуміти» та вивести імпліцитні знання (приховані або неявні знання, якими кандидат володіє на інтуїтивному рівні), наприклад, про те, що:

- володіння фреймворком **React** обов'язково передбачає наявність у IT-фахівця розуміння мови програмування **JavaScript** та вміння застосовувати її на практиці;

- **React** і **Vue.js** належать до однієї підмножини технологій) [19].

Через відсутність ієрархічності та спадковості знань у реляційних БД відбувається невідповідність понять: роботодавець в об'яві про вакансію вказує на потребу у IT-фахівця, що має досвід роботи з фреймворками (все рівно якими, тобто будь-якими) при розробленні frontend-частини системи (**Frontend Frameworks**), а кандидат може вказати лише конкретний технологічний інструментарій, наприклад **React**, через що вже заздалегіть можна сказати, що збіги (понятійні) в об'яві про вакансію і резюме кандидатів при звичайному пошуку (за ключовими словами) [17].

- Аналіз неструктурованого тексту, який було написано роботодавцем в об'яві про вакансію та/або в резюме IT-фахівців – кандидатів на

вакантну посаду.. Резюме та описи вакансій пишуться природною мовою, де контекст має важливе (іноді й вирішальне) значення. Сучасні стандартні машинні парсери не завжди правильно оцінюють та відрізняють фрази на кшталт «маю 4 роки практичного досвіту роботи з мовами програмування C# та Java» та «зараз вивчаю на сертифікованих курсах мови програмування C# та Java». Для визначення з такого тексту справжнього контексту потрібних сутностей потрібен підхід, що передбачає використання ШІ-агентів, робота яких ґрунтується на великих мовних моделях (**Large Language Model, LLM**) [13, 23]). Завдяки такому підходу системи e-recruitment здатні більш точно розуміти семантику речення природною мовою [13, 14].

Серед проблем сучасних систем e-recruitment слід виділити також виконання системами виключно функції пошуку, не надаючи при цьому зворотного зв'язку кандидатам на вакантну посаду. Це обумовлює, зокрема, нерозуміння IT-фахівцем того:

- чому його резюме має низький рівень сумісності з описом бажаної вакантної посади;
- яких конкретно компетенцій йому не вистачає, щоб роботодавці помітили та правильно оцінили його резюме (і відповідно його компетенції та їхній рівень) [11].

Таким чином, актуальність і необхідність розробки системи, яка б поєднувала можливості онтологічного моделювання відповідної предметної області (її об'єктів та процесів) для розрахунку семантичної відстані між відповідними технологіям/технологічним інструментарієм та наявність у системі ШІ-агентів для глибокого лінгвістичного аналізу (насамперед семантичного) контексту в опису вакансій та текстів резюме кандидатів для генерації персоналізованих рекомендацій для кандидатів, що шукають роботу відповідно до рівня власних компетенцій.

Аналіз останніх досліджень і публікацій. Дослідження платформ, що підтримують автоматизовані процеси e-recruitment, свідчать про те, що сучасні системи пошуку роботи мають певні обмеження (що притаманні більшості систем) при аналізі компетенцій кандидатів на вакантні посади в IT-сфері [6, 12].

Наприклад, популярна в Україні платформа Work.ua [21]:

- орієнтується на масовий ринок потреб/пропозицій персоналу, а не тільки на IT-сферу;
- покладається на:
 - реляційні БД;
 - системи повнотекстового індексування, такі як **Elasticsearch**

[24] або **Sphinx** [25]. Головним недоліком такого підходу є простий і негнучкий пошук лише за точним збігом текстових токенів (**keyword matching**) [12]. В ІТ-сфері такий підхід є неефективним через наявність технологій, які зазвичай є новими та/або схожими (спорідненими) та високу варіативність синонімів та аббревіатур. Зокрема, якщо вакансія потребує компетенцій щодо **MS SQL Server**, а кандидат вказав у резюме наявність компетенцій щодо **Microsoft SQL** або **MSSQL**, система повнотекстового пошуку без втручання словників синонімів просто проігнорує це резюме [6].

Інший підхід використовується в спеціалізованій платформі Djinni.co [22], яка базується на системі так званих тегів компетенцій. Кандидати на вакантні посади обирають з уже існуючого єдиного списку технологій, компетенціями з яких вони володіють, частково вирішуючи тим самим проблему синонімії понять (термінологічної синонімії). Але плоска структура БД, що використовується в Djinni.co не надає можливості [17]:

- відображення ієрархічності понять, яким відповідають компетенції кандидатів;
- розуміння семантичних зв'язків.

Теги **JavaScript**, **React** та **Vue.js** зберігаються в таблиці як незалежні поняття (які відповідають конкретним технологіям/технологічним інструментам). Використовуючи цей підхід Djinni.co просто не може зробити правильний логічний висновок (наприклад, такий «**React** і **Vue.js** є схожими клієнтськими фреймворками, а використання **React** обов'язково передбачає знання мови програмування **JavaScript**» [19]. В такій ситуації роботодавець чи його представник змушений перераховувати в об'яві про вакансію всі можливі фреймворки, інакше відповідних ІТ-фахівців, що мають потрібні компетенції, система просто не знайде.

В LinkedIn [26] профілі користувачів аналізуються за допомогою спеціальних алгоритмів ШІ та БД, структура яких відповідає ієрархічності понять, за якими відбувається пошук ІТ-фахівців, що мають потрібні фахові компетенції. Алгоритми, які використовуються в LinkedIn, є власністю компанії, а побудова моделі, що відображає зв'язки між ключовими словами дуже часто дає так званий «шум» у рекомендаціях вакансій, коли пріоритет набувають ті компетенції, що більш характерні для соціальної активності кандидата (користувача LinkedIn) замість того, щоб об'єктивно аналізувати та оцінювати його професійні компетенції [12].

Основними проблемами класичних систем e-recruitment при підборі ІТ-фахівців, зокрема, є [12, 17]:

- збереження інформації у вигляді пласких, не пов'язаних множин сутностей предметної області;
- невідповідність компетенцій, що вказуються в описах вимог щодо вакансій та резюме кандидатів (хтось пише узагальнено, а хтось вказує конкретні технології/фахові компетенції);
- рутинне навантаження на HR-фахівців.

Для усунення семантичного розриву між вимогами вакансій та компетенціями кандидатів можуть бути використані, наприклад, Semantic Web [1, 4] та технології, що ґрунтуються на моделюванні знань [1, 16]. Основним інструментом формалізації знань є онтологія, спеціально розроблена під пошук IT-фахівців. Ця онтологія є моделлю предметної області, що описує сутності, їхні властивості та зв'язки у форматі, придатному для комп'ютерної обробки [5].

Для формування онтологій було розроблено стандарт OWL (**Web Ontology Language**), що базується на дескрипційній логіці (Description Logics, DL) та дозволяє рушіям логічного виведення (reasoners) автоматично виявляти нові зв'язки між певними даними [2].

Мета статті. Метою статті є аналіз сучасного стану процесу підбору IT-фахівців, визначення проблем та перспектив підвищення ефективності та релевантності e-recruitment та розроблення програмного забезпечення системи персоналізованих рекомендацій для IT-фахівців на основі:

- онтологічного моделювання відповідної предметної області;
- використання ШІ-агентів для семантичного метчингу кандидатів і генерації персоналізованих рекомендацій щодо відповідних компетенцій.

Досягнення поставленої мети потребує вирішення наступних завдань:

- Проведення аналізу сучасних платформ e-recruitment.
- Обґрунтування використання онтологічного моделювання предметної області для предметної області e-recruitment.
- Розроблення онтології, яка б описувала:
 - компетенції IT-фахівців (кандидатів на відповідні вакантні посади) щодо IT-технологій;
 - ієрархію компетенцій в IT-сфері (щодо мов програмування, фреймворків, БД тощо), якими можуть/повинні володіти кандидати, які шукають роботу в IT-сфері;
 - семантичні зв'язки компетенцій та професійних ролей IT-фахівців.
- Розроблення алгоритму порівняння семантичної подібності компетенцій (Match Score), що ґрунтується на онтологічній відстані у відповідному графі знань.

- Інтегрування ШІ-агента для лінгвістичного аналізу текстів резюме та описів вакансій з подальшим виділенням основних понять щодо технологій (технологічних інструментів) ІТ-сфери та їхніх назв.
- Розроблення підсистеми генерації персональних індивідуалізованих рекомендацій для конкретного кандидата, які використовують знання та компетенції кандидата та виявлені в них так звані «про-галини».

Розроблення програмного забезпечення лексичного пошуку термінологічного співпадання на базі клієнт-серверної архітектури, використовуючи такі технології: **FastAPI, React, PostgreSQL, Owlready2** [7–9].

Виклад основного матеріалу дослідження. Для вирішення проблем дослідження доцільно здійснити порівняння класичного реляційного та онтологічного підходів. Реляційні БД, що орієнтовані на збереження даних і транзакцій, мають жорстку схему, що відображена у вигляді таблиць, стовпців, рядків. При такому підході дані, які відсутні в БД, вважаються хибними [9]. Крім того, в цих БД відсутнє логічне виведення нових знань на рівні систем керування БД.

Бази знань, що використовують онтології (описані за допомогою OWL [2]), орієнтовані на контекст знання, а не на термін, який відноситься до нього. Це сприяє логічному виведенню нових знань [17]. Бази знань використовують динамічну схему (на основі RDF/OWL) та спираються на підхід, згідно з яким відсутні дані вважаються невідомими [2].

Це вбудоване логічне виведення (reasoning) дозволяє системі адаптуватись, відкриваючи нові факти. Зв'язки (**Relations**) в онтологіях [1, 2]:

- є самостійними сутностями, що мають власні ієрархії (на відміну від жорстких зовнішніх ключів у межах реляційних БД);
- забезпечують високу семантичну сумісність єдиних ідентифікаторів (IRI/URI) та стандартів W3C.

Для розроблення системи персоналізованих рекомендацій щодо рекрутингу ІТ-персоналу, здатної розмірковувати подібно до досвідченого HR-фахівця, було створено онтологію, структура якої є ієрархічною, забезпечуючи її легку інтеграцію з існуючими стандартами.

В розробленій системі поєднано: класичні реляційні БД (для транзакційного збереження), бази знань на основі онтологій (для семантичного виведення нових знань) та великі мовні моделі (LLM-агенти) для аналізу природної мови [13, 19]. Це поєднання сприяє роботі системи не тільки для збереження резюме кандидатів на вакантні посади, а й як ШІ-радник щодо подальшої кар'єри (професійної діяльності) кандидатів [12].

Система має дві частини:

- серверну (backend), що побудована з використанням вебфреймворку FastAPI, що базується на мові програмування Python [7], вибір якої обґрунтовано, зокрема, наявністю спеціалізованих бібліотек для роботи з Semantic Web [1, 4].
- клієнтську (frontend), що реалізована з використанням бібліотеки React [8].

Дані в системі розділені на такі складові:

• **Транзакційну**, що реалізується за допомогою PostgreSQL [9] і відповідає за збереження таких даних:

- облікові записи користувачів системи — роботодавців, кандидатів;
- профілі компаній-роботодавців;
- текстові описи вакансій (з описом потрібних/необхідних компетенцій);
- тексти резюме кандидатів.

Сервер взаємодіє з БД через ORM-бібліотеку SQLAlchemy [10].

• **Семантичну**, яка представлена OWL-файлом бази знань (онтологією) [2]. Для забезпечення алгоритму метчингу онтологічний граф (**OntoGraf**) передається серверу методами бібліотеки Owlready2 [4]. Це сприяє швидкому пошуку суміжних класів об'єктів предметної області та розраховувати семантичну відстань між ними у режимі real-time [17].

ШІ-агент (на базі LLM) [13], отримуючи текст опису вакансії або резюме кандидата, написаний природною мовою, здійснює розпізнавання сутностей (**Named Entity Recognition, NER**) з урахуванням їхнього лінгвістичного контексту у тексті [14]. ШІ-агент виконує, зокрема, наступне:

- розпізнає реальні фахові компетенції кандидата (знання технології та практичний досвід використання технологічного інструментарію);
- нормалізує їхні назви до тих, що є в онтології;
- зіставляє їх зі словником системи, який синхронізований з онтологією.
- виступає мостом між природною мовою користувача та онтологією (нормалізованою логікою онтографа) [13].

В клієнтській частині системи:

- для збереження JWT-токенів авторизації, кешування списків вакансій та результатів метчингу використано Redux Toolkit [20];
- візуальне оформлення створено за допомогою фреймворку TailwindCSS;
- інтерфейс користувачів підтримує їхні зручні особисті кабінети, де рекомендації, згенеровані на основі онтології предметної області, візуалізуються у вигляді індикаторів збігу (**Match Score**) [11, 15] та додаткових панелей з так званими «кар'єрними» рекомендаціями.

Проектування онтології та опис її сутностей. Для побудови онтології використовується платформа **Protégé** [3]. Онтологія (рис. 1) поділена на дві основні категорії знань предметної області: ієрархічну структуру технологічного стека та професійні ролі (посади), що існують на сучасному ринку IT. Важливою основою онтології є її таксономія – ієрархія класів (**Classes**), які об'єднують сутності за їхніми спільними властивостями [5].

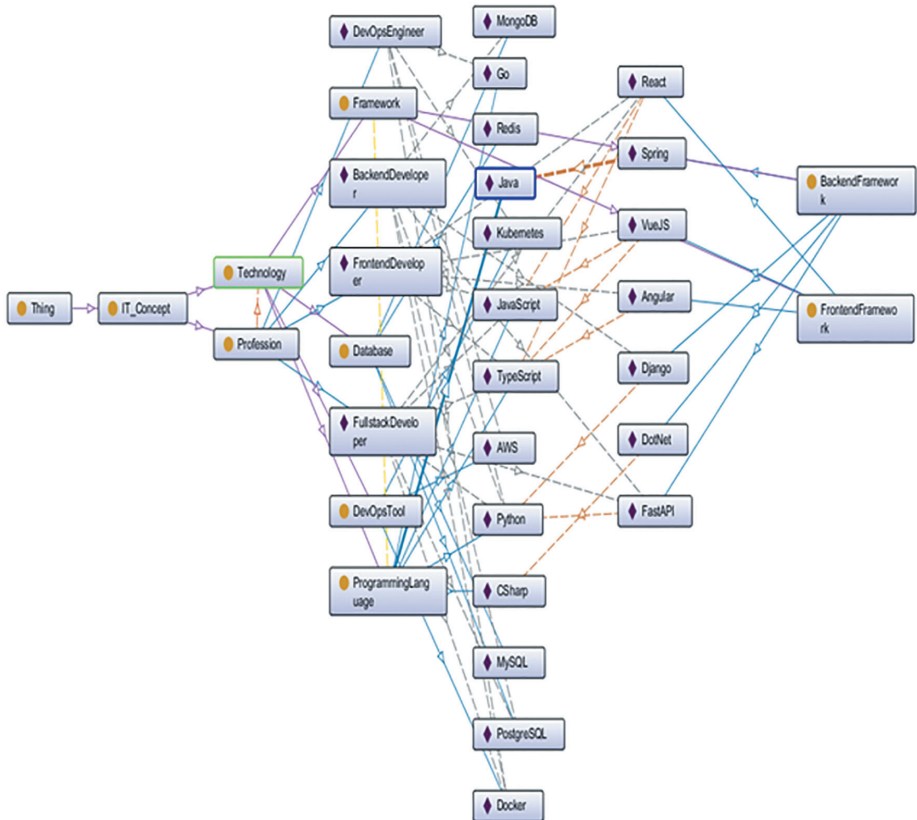


Рис. 1. Ієрархічна структура класів IT-сфери у середовищі Protégé

Джерело: [авторська розробка]

Основним класом онтологічної моделі виступає **IT_Concept**, який є підкласом метакласу **owl:Thing**. Від **IT_Concept** будується розгалуження на два базові класи, які не перетинаються між собою [2]:

- **Technology** – це клас, що відповідає за будь-яку технологічну сутність, концепцію або інструмент, який буде використовуватися в робочих IT-процесах.
- **Profession** – це клас, що характеризує професійні посади та технологічні напрямки на ринку праці в IT-сфері (наприклад, **FrontendDeveloper**, **DevOpsEngineer**).

Клас **Technology** має такі підкласи:

- **ProgrammingLanguage** – тут прописані мови програмування (наприклад, такі як **JavaScript**, **Python**, **C#** тощо);
- **Framework** – програмні основи, які розділені на підкласи **FrontendFramework** (**React**, **Angular**, **VueJS** та інші) та **BackendFramework** (**FastAPI**, **Django**, **Spring** та інші);
- **Database** – системи керування БД реляційного та нереляційного типу (**PostgreSQL**, **MongoDB** та інші);
- **DevOpsTool** – інструментарій для автоматизації розгортання (CI-, CD-процесів) та хмарні інфраструктури (**Docker**, **Kubernetes**, **AWS** та інші).

Множина семантичних властивостей об'єктів (**Object Properties**) [2] використовується для створення повноцінної бази знань. Ці властивості об'єктів будують правила взаємодії між різними відгалуженнями ієрархії класів.

Властивістю, що вирішує проблему прихованих залежностей між компетенціями кандидатів, які їх не вказали, але натякнули на їхнє існування в інших компетенціях, є властивість **requires_language** (вимагає мову).

Ця властивість обумовлює створення логічного правила, яке поєднує:

- клас **Framework**, що виступає джерелом зв'язку;
- клас **ProgrammingLanguage** є його ціллю [5, 17].

Покладаючись на ці правила, система зможе розуміти приховані знання, практичні навички та компетенції кандидата. Наприклад, якщо кандидат вказав, що володіє фреймворком **React**, система (через транзитивний зв'язок, що **React** потребує знання мови **JavaScript**) автоматично зробить висновок про володіння кандидатом базовою мовою програмування **JavaScript** [19].

Важливою властивістю є **uses_technology**, яка встановлює зв'язок між класом **Profession** та класом **Technology**. Це сприяє створенню певних вимог до конкретних посад. Наприклад, сутність **BackendDeveloper** пов'язується цією властивістю з сутностями **Python**, **FastAPI** та **PostgreSQL**. Ці зв'язки сприяють не просто порівнюванню резюме з конкретною вакансією, а й автоматичному генеруванню профілю так званого «ідеального кандидата» для певної посади [17].

Після формування структури класів та їхніх властивостей онтологію було наповнено індивідами (**Individuals**) – конкретними екземплярами технологій, які сформували первинний словник системи. Верифікація розробленої моделі на логічну несуперечливість була виконана за допомогою логічного рушія (**reasoner**) **HermiT** [3].

Парсинг тексту та роль ШІ-агента. Процес розуміння системою неструктурованого тексту (описів вакансій та резюме) та перетворення у спеціалізовані вузли онтографа виконується спеціальною підсистемою (модулем) парсингу [14]. В розробленій системі персоналізованих рекомендацій для ІТ-фахівців модуль побудований за такою схемою [13]:

- виконання лексико-синтаксичного аналізу на основі динамічних регулярних виразів;
- глибокий контекстний аналіз із залученням ШІ-агента.

Проблемою класичного повнотекстового пошуку є відсутність гнучкості, через що стандартні алгоритми щодо межі слова працюють некоректно [12]. При класичному пошуку символ «+» буде відкинуто і система розпізнає **C++** як **C**, що призведе до семантичної помилки, для усунення якої система використовує алгоритм динамічної побудови регулярних виразів. Функція парсера (**compile_skill_pattern**) виконує аналіз символічного складу кожної назви компетенції з БД. Використання лише регулярних виразів має суттєвий недолік так звану «контекстну сліпоту». Цей лексичний парсер успішно вилучає сутності, але не зрозуміє контексту, в якому вони вжиті [12].

Отримавши виділені токени, ШІ-агент виконує завдання NER [13, 14]. Аналізуючи синтаксичну структуру речень за допомогою механізмів уваги (**attention mechanisms**), які використовують LLM, ШІ-агент спроможний розрізняти, зокрема [14]:

- реальний практичний досвід і теоретичні знання;
- плани щодо навчання та наявні компетенції.

Далі контекстний аналіз ШІ-агента виконує операцію нормалізації, проводячи фільтрацію хибних збігів, уніфікуючи синоніми та аббревіатури, що сформує множину нормалізованих фахових (технологічних) компетенцій. Цей підхід гарантує, що до онтологічного розрахунку Match Score потрапляють лише компетенції, якими кандидат дійсно володіє, підвищуючи тим самим якість фінальних рекомендацій.

Алгоритм семантичного метчингу. Для якісної оцінки відповідності кандидата (точніше його компетенцій) вимогам певної вакансії у системі e-recruitment використовується алгоритм семантичного метчингу, який базується на метриках семантичної близькості (**Semantic Similarity**) у таксономіях та методах онтологічного порівняння компетенцій [17-19].

На відміну від алгоритмів в традиційних системах e-recruitment, які використовують булеву логіку (збіг є/немає), алгоритм семантичного метчингу оперує нечіткими множинами та обчислює для компетенцій коефіцієнт подібності $Sim \in [0, 1.0]$.

Введемо такі позначення: D – множина вимог до певної вакансії; $S_i^{req} \in D$ – конкретна технологія/технологічний інструмент, що є необхідним для задоволення i -ої вимоги; C – множина компетенцій кандидата, які виділяються з тексту його резюме, де $S_i^{cand} \in C$. Функція семантичної подібності $Sim(S^{req}, S^{cand})$ розрахує рівень збігу між цими двома концептами, використовуючи онтологію та один з наведених нижче методів обчислення показників збігів [17-19]:

- **Лексико-семантичний збіг:** Якщо ідентифікатори компетенцій і вимог в онтології є тотожними, то система зазначить повну відповідність між ними (наприклад, вакансія вимагає React, кандидат володіє React).
- **Спадкова семантика:** Якщо вакансія потребує знання базової мови програмування, а кандидат зазначив, що володіє фреймворком, який будується на цій мові, то використовується властивість `requires_language` для логічного виявлення зв'язку між вимогами і компетенціями, коли кандидат знає, що більш складний похідний інструмент автоматично «закриває» питання щодо базової мови програмування (наприклад, вакансія вимагає знання JavaScript, а кандидат володіє React). Система в цій ситуації оцінює подібність між вимогою і компетенцією як максимальну.
- **Часткова сумісність:** Якщо вакансія вимагає від кандидата знання, наприклад, конкретного фреймворку, а той володіє лише знаннями мови програмування, на якій фреймворк базується, то система нараховує коефіцієнт подібності – 0,5, бо вивчення фреймворку за умови знання його базової мови потребує від кандидата менших зусиль, ніж вивчення з «нуля».
- **Однорівнева горизонтальна подібність:** Якщо технології належать до одного класу (наприклад, належать класу `FrontendFramework`), то система буде вважати їх спорідненими технологіями (наприклад: вакансія вимагає React кандидат володіє `Vue.js`). Подібність в цій ситуації буде 0,6.
- **Опосередкована категоріальна подібність:** Якщо технології не є прямо спорідненими, але мають спільного предка на більш високого рівня в ієрархії (наприклад, обидві належать до узагальненого класу `Framework`), то система фіксує далеку категоріальну подібність. Подібність становить 0,2.
- **Повна відсутність зв'язку:** Якщо технології належать до різних, неперетинних гілок у онтології (наприклад, БД та інструменти `DevOps`) які не мають жодних спільних зв'язків у онтології, то така семантична відстань вважається мінімальною. Подібність становить 0,0.

Кандидат має багато компетенцій, тому для кожної вимоги у вакансії алгоритм знаходить максимальне значення подібності серед цих компетенцій з резюме кандидата.

Відповідність кандидата певній вакансії обчислюється як середнє арифметичне максимальних подібностей для всіх вимог вакансії.

Запропонований алгоритм забезпечує гнучкість повнотекстового пошуку. Наприклад, якщо вакансія буде вимагати знання **FastAPI**, **PostgreSQL** та **Docker**, а кандидат володіє **Django** (споріднений з **FastAPI** фреймворк, 60%), **PostgreSQL** (прямий збіг, 100%) та **Docker** (прямий збіг, 100%), запропонований алгоритм розрахує відповідність кандидата вакансії дорівнює **86.7%**. У цій ситуації ж класична система e-recruitment не змогла б так точно оцінити відповідність кандидата вимогам вакансії.

Генерація персоналізованих рекомендацій. Однією з проблем, з якою стикаються ті, хто шукає роботу на платформах e-recruitment, є їхня пасивність. Вони виконують виключно функцію фільтрації вакансій для кандидатів на вакантні посади, не надаючи їм зворотного зв'язку. Кандидати потрапляють у ситуації коли їм не зрозуміло, що слід вивчити/знати, щоб роботодавець їх помітив. Для вирішення цієї проблеми у розроблену систему інтегровано підсистему генерації персоналізованих рекомендацій користувачеві, яка спроможна надати пораду/рекомендацію, запропонувавши кандидатові персональну траєкторію професійного розвитку.

Система здійснює аналіз результатів метчингу для кожної окремої вимоги у вакансії. Генерація рекомендацій ініціюється для тих навичок S_i^{req} , для яких максимальна семантична подібність із профілем/резюме кандидата є меншою за 1.0 (тобто коли виявлено «прогалину» в знаннях кандидата). Формування рекомендації не є шаблонним, вона генерується з урахуванням, зокрема:

- досвіду кожного кандидата та вимог ринку;
- того, який саме логічний зв'язок в онтографі забезпечив частковий збіг.

Генерація рекомендацій передбачає:

• **Рекомендації на основі базової мови програмування** (Language-to-Framework). Якщо подібність в цій ситуації буде 0,5 (вакансія може вимагати знання фреймворку, а кандидат вже знає його базову мову), то система генерує підказку: «Ви вже володієте даною мовою S_i^{cand} . Рекомендуємо вам розпочати вивчення даного фреймворку S_i^{req} ». Це визначить напрямки і пришвидшить процес навчання для кандидата на вакантну посаду.

• **Рекомендації на основі спорідненості.** Якщо подібність буде 0,6 і технології споріднені та мають спільного безпосереднього батька в онтології

(наприклад, React та Vue.js), то буде надана рекомендація, що ґрунтується на схожості компетенцій: «Оскільки Ви маєте досвід роботи з певною технологією тієї ж категорії S_i^{cand} , що і технологія S_i^{req} , то її вивчення буде для Вас нескладним. Зверніть увагу на те, що вказана технологія може збільшити множину вакансій, на які Ви можете претендувати».

• **Рекомендація вивчення чогось важливого з «нуля».** Якщо подібність буде $< 0,4$ (що свідчить про відсутність близьких семантичних зв'язків або повну відсутність компетенції кандидата щодо вакансії), то система вкаже йому на цю критичну «прогалину»: «Для проходження відбору на відповідну вакансію Вам необхідно вивчити технологію S_i^{req} , з «нуля», бо вона є критично важливою для претендентів на дану посаду».

Крім аналізу конкретної пари «резюме-вакансія» система, використовуює властивість **uses_technology**, яка пов'язує професії (клас **Profession**) з набором технологій (клас **Technology**). Це дозволить кандидату, обравши бажану посаду (наприклад, **.NET FullstackDeveloper**), побачити свої критичні «прогалини» у знаннях, завдяки тому що система використовуючи онтологію (онтограф):

- сформує стеки-еталони технологій для більшості вакансій;
- порівняє їх з наявними у кандидата компетенціями;
- побудує глобальну «дорожню карту» навчання (що насамперед слід зараз вивчити).

Такий підхід доповнює роботу системи e-recruitment, роблячи її персональним помічником для тих, хто шукає роботу. Підсистема генерації рекомендацій на основі онтологічної моделі відповідної предметної області, надаючи рекомендації претендентам//кандидатам, формує чіткий план їхнього розвитку. Це суттєво підвищує зручність та цінність розробленої системи для користувачів, стимулюючи їх до подальшого професійного розвитку.

Висновки. В роботі було проведено аналіз сучасних систем e-recruitment, які використовуються в сфері підбору IT-фахівців, визначено їх переваги та недоліки. В роботі було також:

- досліджено процеси й сучасні технології лінгвістичного аналізу природомовних текстів об'яв про вакансії та резюме кандидатів;
- визначено основні проблеми e-recruitment;
- визначено основні проблеми лінгвістичного аналізу природомовних текстів;
- розроблено онтологічну модель компетенцій (визначених в множини вакансій щодо IT-сфери та резюме кандидатів);
- розроблено авторський прототип системи e-recruitment, в якому реалізовано механізми автоматизованого підбору IT-персоналу.

Було продемонстровано, що використання онтологічного підходу сприяє ефективному подоланню «семантичного розриву» між вимогами, що висунуті роботодавцями в об'явах про вакансії, та компетенціями, які вписали у резюме кандидати. Алгоритм розрахунку семантичної подібності сприяє виявленню прихованих знань.

Інтегрування в систему ШІ-агента (на основі LLM) обумовило високий рівень контекстного аналізу неструктурованих текстів резюме та вакансій.

Розроблено підсистему генерації персоналізованих рекомендацій для тих, хто шукає роботу. Завдяки графу знань система надає кандидатам персоналізовані поради щодо вивчення та закриття «прогалин» щодо відсутніх у кандидата компетенцій.

Подальший розвиток системи e-recruitment передбачає, зокрема:

- масштабування бази знань новими ІТ-посадами, навичками та іншими сутностями онтології відповідної предметної області ІТ-сфери;
- можливе застосування методів машинного навчання для автоматичного адаптивного коригування коефіцієнтів в алгоритмі метчингу з урахуванням ситуації, що складається на ринку праці в ІТ-сфері.

ЛІТЕРАТУРА

1. Berners-Lee T., Hendler J., Lassila O. The Semantic Web. *Scientific American*, 2001. Vol. 284(5). P. 34–43.
2. OWL Web Ontology Language Overview. W3C Recommendation, 2004. URL: <https://www.w3.org/TR/owl-features/> (дата звернення: 11.06.2026).
3. Musen M. A. The Protégé project: a look back and a look forward. *AI Matters*, 2015. Vol. 1(4). P. 4–12.
4. Lamy J. B. Owlready2: A Python module for ontology-oriented programming, with or without Class/Individual confusion. *Artificial Intelligence in Medicine*, 2017. Vol. 80. P. 11–14.
5. Gruber T. R. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 1993. Vol. 5(2). P. 199–220.
6. Al-Qurishi M. et al. A semantic matching engine for job recruitment in cloud environment. *IEEE Access*, 2015. Vol. 3. P. 2217–2228.
7. FastAPI framework documentation. URL: <https://fastapi.tiangolo.com/> (дата звернення: 11.06.2026).
8. React: A JavaScript library for building user interfaces. URL: <https://react.dev/> (дата звернення: 14.06.2026).
9. PostgreSQL 16 Documentation. PostgreSQL Global Development Group, 2023. URL: <https://www.postgresql.org/docs/16/> (дата звернення: 09.06.2026).

10. SQLAlchemy 2.0 Documentation. URL: <https://docs.sqlalchemy.org/en/20/> (дата звернення: 09.06.2026).
11. Zhang Y., Chen X. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends in Information Retrieval*, 2020. Vol. 14(1). P. 1–101.
12. Brek A., Boufaïda M. Explainable person–job recommendations: challenges, approaches, and comparative analysis. *Frontiers in Artificial Intelligence*, 2025. Vol. 8.
13. Salakar E. et al. Resume Screening Using Large Language Models. 2023 6th International Conference on Advances in Science and Technology (ICAST), 2023. P. 494–499.
14. Almasbekuly A. Z. Benchmarking Large Language Models for Information Extraction from Job Vacancy Descriptions. *Research Reviews*, 2026. № 12.
15. Arrieta A. B. et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 2020. Vol. 58. P. 82–115.
16. Bizer C., Heath T., Berners-Lee T. Linked Data: The Story So Far. *International Journal on Semantic Web and Information Systems*, 2009. Vol. 5(3). P. 1–22.
17. García-Sánchez F. et al. Ontology-based skill matchmaking framework. *Information Systems*, 2006. Vol. 31(6). P. 551–574.
18. Wu Z., Palmer M. Verbs semantics and lexical selection. Proceedings of the 32nd annual meeting on Association for Computational Linguistics, 1994. P. 133–138.
19. Fenza G., Loia V., Senatore S. A hybrid approach to semantic skill matchmaking. *International Journal of Intelligent Systems*, 2012. Vol. 27(4). P. 345–364.
20. Redux Toolkit. URL: <https://redux-toolkit.js.org>. (дата звернення: 09.06.2026)
21. Work.ua. Сервіс пошуку роботи №1 в Україні. URL: <https://www.work.ua>. (дата звернення: 08.06.2026).
22. Djinni. Hire talent or find a job:remotely & on your own. URL: <https://djinni.co/hire>. (дата звернення: 08.06.2026).
23. Остаточний гід: Що таке LLM і чому це важливо у 2025 році. URL: <https://www.pippit.ai/uk-ua/resource/what-is-llm> (дата звернення: 09.06.2026).
24. Elasticsearch. URL: <https://www.elastic.co/elasticsearch> (дата звернення: 09.06.2026).
25. Sphinx. URL: <https://sphinxsearch.com> (дата звернення: 09.06.2026).
26. LinkedIn. Ласкаво просимо до професійної спільноти! URL: <https://ua.linkedin.com> (дата звернення: 09.06.2026).

REFERENCES

1. Berners-Lee T., Hendler J., Lassila O. The Semantic Web. *Scientific American*, 2001. Vol. 284(5). P. 34–43.
2. OWL Web Ontology Language Overview. W3C Recommendation, 2004. URL: <https://www.w3.org/TR/owl-features/> (Accessed 11.06.2026).

3. Musen M. A. The Protégé project: a look back and a look forward. *AI Matters*, 2015. Vol. 1(4). P. 4–12.
4. Lamy J. B. Owlready2: A Python module for ontology-oriented programming, with or without Class/Individual confusion. *Artificial Intelligence in Medicine*, 2017. Vol. 80. P. 11–14.
5. Gruber T. R. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 1993. Vol. 5(2). P. 199–220.
6. Al-Qurishi M. et al. A semantic matching engine for job recruitment in cloud environment. *IEEE Access*, 2015. Vol. 3. P. 2217–2228.
7. FastAPI framework documentation. URL: <https://fastapi.tiangolo.com/> (Accessed 11.06.2026).
8. React: A JavaScript library for building user interfaces. URL: <https://react.dev/> (Accessed 14.06.2026).
9. PostgreSQL 16 Documentation. PostgreSQL Global Development Group, 2023. URL: <https://www.postgresql.org/docs/16/> (Accessed 09.06.2026).
10. SQLAlchemy 2.0 Documentation. URL: <https://docs.sqlalchemy.org/en/20/> (Accessed 09.06.2026).
11. Zhang Y., Chen X. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends in Information Retrieval*, 2020. Vol. 14(1). P. 1–101.
12. Brek A., Boufaïda M. Explainable person–job recommendations: challenges, approaches, and comparative analysis. *Frontiers in Artificial Intelligence*, 2025. Vol. 8.
13. Salakar E. et al. Resume Screening Using Large Language Models. 2023 6th International Conference on Advances in Science and Technology (ICAST), 2023. P. 494–499.
14. Almasbekuly A. Z. Benchmarking Large Language Models for Information Extraction from Job Vacancy Descriptions. *Research Reviews*, 2026. № 12.
15. Arrieta A. B. et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 2020. Vol. 58. P. 82–115.
16. Bizer C., Heath T., Berners-Lee T. Linked Data: The Story So Far. *International Journal on Semantic Web and Information Systems*, 2009. Vol. 5(3). P. 1–22.
17. García-Sánchez F. et al. Ontology-based skill matchmaking framework. *Information Systems*, 2006. Vol. 31(6). P. 551–574.
18. Wu Z., Palmer M. Verbs semantics and lexical selection. *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, 1994. P. 133–138.
19. Fenza G., Loia V., Senatore S. A hybrid approach to semantic skill matchmaking. *International Journal of Intelligent Systems*, 2012. Vol. 27(4). P. 345–364.
20. Redux Toolkit. URL: <https://redux-toolkit.js.org> (Accessed 09.06.2026).
21. Work.ua. Ukraine's #1 job service. URL: <https://www.work.ua/en> (Accessed 08.06.2026).

22. Djinni. Hire talent or find a job:remotely & on your own. URL: <https://djinni.co/hire> (Accessed 08.06.2026).
23. The Ultimate Guide: What is an LLM and Why is it Important in 2025. URL: <https://www.pippit.ai/uk-ua/resource/what-is-llm> (Accessed 09.06.2026).
24. Elasticsearch. URL: <https://www.elastic.co/elasticsearch> (Accessed 09.06.2026).
25. Sphinx. URL: <https://sphinxsearch.com> (Accessed 09.06.2026).
26. LinkedIn. Welcome to the professional community! URL: <https://ua.linkedin.com> (Accessed 09.06.2026).