УДК 004.8 DOI: 10.30837/0135-1710.2024.180.055

В.М. ЛЕВИКІН, Д.О. ДІДЕНКО, О.А. АЛЬОШКІН

МЕТОД ФОРМУВАННЯ ЗАЯВОК ПРИРОДНОЮ МОВОЮ НА ОСНОВІ ВДОСКОНАЛЕНОЇ МОДЕЛІ BERT

Розглянуто основні особливості формування заявки на отримання послуги як однієї з важливих задач сучасних систем управління послугами. Проведено аналіз сучасних інформаційних систем і технологій, визначено переваги і недоліки існуючих моделей і методів, які використовуються для вирішення цієї задачі. Розроблено комбінований метод формування заявок на отримання послуги, в якому поєднуються особливості моделі BERT, простіших методів машинного навчання та байєсівської класифікації. Розглянуто особливості програмної реалізації та апробації розробленого методу на прикладі вирішення задачі формування і обробки заявок на оренду автомобілів.

1. Вступ

Переважна більшість сучасних інформаційних систем (ІС) управління бізнес-діяльністю підприємств та організацій побудована на основі концепції надання послуг і управління цими послугами. Головна ідея цієї концепції полягає не в просуванні нових продуктів на ринок, а в зосередженні зусиль підприємства чи організації на задоволенні потреб клієнтів [1]. З 2005 року почалися роботи зі створення типової системи управління послугами, базовані на цій концепції. Основним поточним результатом цих робіт слід визнати зафіксовані у стандарті ISO 20000:2018 основні вимоги до процесів та функцій такої системи [2]. За цими вимогами, надання будь-якої послуги можливе або за результатом визначення потреби кожного окремого користувача (заявки на отримання послуги), або за результатом аналізу множини аналогічних, раніше сформульованих потреб значної кількості користувачів. Тому проблеми, пов’язані з вирішенням задачі формування заявок на отримання послуги, слід визнати такими, що вимагають особливої уваги.

На жаль, зазначені у [2] вимоги не можуть надати ніякої інформації щодо моделей та методів, які можна застосовувати для вирішення задач управління послугами. Тому під час створення окремих ІС та інформаційних технологій (ІТ) автоматизованого управління послугами у різних предметних галузях постійно існує необхідність пошуку, розробки або вдосконалення моделей, методів і алгоритмів, які могли б вирішити відповідні задачі найкращим чином.

Ця необхідність значно ускладнює проведення робіт зі створення ІТ вирішення задачі формування заявок на отримання послуги. Зазначене ускладнення, зокрема, пов’язане з необхідністю застосовувати для цієї ІТ моделі і методи, які відповідали б таким умовам:

а) можливість сприйняття потреб користувача, сформульованих його природною мовою;

б) можливість поступової формалізації зафіксованих описів потреб користувача;

в) можливість попередньої класифікації сформованих заявок з метою покращення надання відповідної послуги.

Одним з напрямів розробки та вдосконалення згадуваних вище моделей і методів є постійне збільшення позитивного досвіду користувачів таких систем. Досвід користувача буде позитивним, якщо при оформленні заявки на отримання послуги йому необхідно буде лише надати свої дані у систему через інтерфейс та не контактувати із менеджерами для їх

підтвердження. Це скорочує час на оформлення заявки та полегшує цей процес. Крім того, коли користувач не витрачає час на розмови з менеджером та на уточнення інформації, підвищується його лояльність до компанії та зацікавленість у користуванні її послугами. З іншого боку, скорочення витрат часу менеджера за рахунок неучасті в оформленні заявок також позитивно сприяє бізнесові.

Таке збільшення позитивного досвіду користувачів і бізнесу стає особливо важливим під час створення та експлуатації ІС та ІТ автоматизованого управління послугами, заявки на отримання яких формуються користувачами природною мовою у вигляді коротких неструктурованих текстів. Такі заявки є характерними для систем малого і середнього бізнесу, а також для платформ дистанційного надання державних послуг громадянам країни чи окремого регіону. Тому існує постійна необхідність проведення науково-дослідних робіт та реалізації стартапів, спрямованих на постійне поліпшення моделей і методів розпізнавання коротких текстів природною мовою та прикладних засобів їх реалізації. Найбільшу зацікавленість під час проведення досліджень в межах цього напрямку з урахуванням умов а) та в) викликають моделі і методи штучного інтелекту, які дозволяють формалізувати обробку текстів природною мовою та отримувати з цих текстів дані і знання, необхідні для оформлення заявок. Тому проведення досліджень з вдосконалення та розвитку існуючих методів штучного інтелекту, спрямованих на обробку природної мови користувача, є актуальними для поліпшення вирішення задачі формування заявок на отримання послуги.

2. Аналіз літературних даних і постановка проблеми дослідження

2.1. Аналіз сучасних інформаційних технологій, використовуваних для автоматизації формування заявок на отримання послуги

Процес формування заявок на отримання послуги є невід'ємною частиною систем управління підприємствами та організаціями у різних галузях економічної діяльності суспільства. Тому не дивно, що значна кількість сучасних розробок в галузі ІТ присвячена вирішенню проблеми автоматизації цього процесу. Такі розробки, зазвичай, є ІС або ІТ, що поєднують в собі різноманітні функції з метою підвищення ефективності, зручності та точності виконання окремих процесів і функцій управління послугами [3].

Важливою частиною таких ІС або ІТ є онлайн-платформа, яка дозволяє клієнтам здійснювати реалізацію заявки на отримання послуги через веб-сайт або мобільний додаток [3-5]. Користувачі, наприклад, можуть під час створення заявки визначати особливості опису послуги, виходячи з предметної галузі, вказувати терміни та місце отримання обраної послуги, а також додавати додаткові опції. Такі системи включають також модулі для внутрішнього управління компанією, включаючи облік послуг, які може отримати користувач, розподіл цих послуг між різними локаціями, ведення історії обслуговування та станів окремих послуг, а також моніторинг ресурсів, які витрачаються під час надання послуги за заявкою [3, 5]. Крім того, такі ІС або ІТ можуть включати модулі для фінансового обліку, виставлення рахунків та обробки даних про оплату [3]. Функціональні можливості таких ІС або ІТ можуть розширюватися в результаті інтеграції додаткових аналітичних сервісів чи технологій [3-5]. Такі інструменти дозволяють керівництву підприємств або організацій оцінювати ефективність надання послуг, виявляти тенденції попиту на окремі послуги та групи послуг, формувати і приймати управлінські рішення тощо. В цілому подібні ІС або ІТ поліпшують управління тими галузями економічної діяльності, які базуються на процесах надання послуг,

що дозволяє зосередитися на покращенні обслуговування клієнтів та оптимізації внутрішніх процесів.

Актуальним базовим підходом до автоматизації процесів формування та обробки заявок на отримання послуги залишається підхід, який базується на використанні рекомендаційних систем. Цей підхід застосовується в IT-продуктах основних компаній, що є лідерами цього сегменту ринку [6]. Такі IT-продукти дозволяють розробляти різні рекомендації, легко інтегруються до існуючих систем та мають зручний кабінет адміністратора як інструмент управління функціями рекомендаційної системи. Механізм рекомендацій таких IT-продуктів може бути застосований до будь-якого домену предметної галузі, котрий характеризується наявністю каталогу послуг і можливістю взаємодії з великою кількістю користувачів [6, 7]. Застосована у веб- та мобільних додатках рекомендаційна система сприяє поліпшенню користувацького досвіду за рахунок надання кожному окремому користувачу послуги, яка характеризується найбільшою релевантністю окремій заявці цього користувача [7, 8]. При цьому в таких IT-продуктах є можливість визначати різні типи властивостей послуги, які піддаються аналізу з використанням рекомендаційних моделей на основі вмісту заявки користувача [7]. Додатковою властивістю цих продуктів є можливість використання фільтрів або бустерів під час розробки правил рекомендаційної системи, що дозволяє підвищувати якість цих правил [6].

Окремою особливістю рекомендаційних систем та IT на їх основі є механізм взаємодії між цими IT та IC або IT, які використовують рекомендаційні системи у своїй діяльності. Сучасні рекомендаційні IT як основні варіанти реалізації такої взаємодії розглядають: віджети HTML, API на стороні клієнта, API на стороні сервера або інтеграції сегмента [6].

Для ініціації сценаріїв роботи рекомендаційної системи необхідна наявність заявки користувача на отримання послуги з публікацією цієї заявки на веб-сайті або у мобільному додатку IC або IT, в електронній пошті користувача тощо. Кожен сценарій орієнтований на певний тип рекомендацій, наприклад: рекомендація для конкретного користувача, рекомендація елементів, пов'язаних із заявкою чи послугою або персоналізований повнотекстовий пошук. Крім цього, кожен сценарій має можливість додаткових налаштувань.

Бажану реалізацію рекомендаційної моделі визначає рекомендаційна логіка, для різних випадків використання рекомендаційних систем можуть використовуватися різні рекомендаційні логіки. Результати функціонування рекомендаційних систем використовуються для фільтрації або вибору окремих послуг на основі їхніх властивостей. Правила, які часто використовуються, в багатьох випадках можуть бути реалізовані у вигляді окремих бібліотек, доступних для негайного використання. Як інструменти формування унікальних правил в багатьох рекомендаційних системах застосовуються інтуїтивно зрозуміла мова запитів (наприклад, ReQL) або механізми інтеграції рекомендацій за допомогою віджета HTML або API & SDK обраної мови програмування [6].

Окремі розробки у цьому напрямку надають рекомендаційним системам додаткові можливості. Як приклад таких розробок можна згадати систему персоналізації Dynamic yield [9], в якій на основі засобів штучного інтелекту реалізовано можливості персоналізації та оптимізації рекомендацій. На думку авторів цієї системи, зазначені можливості допомагають провідним брендам створювати особистий цифровий досвід, який сприяє залученню нових користувачів, конверсії та збільшенню доходу, поліпшенню лояльності та задоволеності клієнтів, надаючи індивідуальний контент, рекомендації та рекламні акції кожному окремому

користувачеві [9]. Іншим прикладом таких розробок є система Amazon Personalize [10], у якій застосовані технології машинного навчання (ML). Дане рішення спрощує інтеграцію персоналізованих рекомендацій у існуючі веб-сайти, додатки, системи електронного маркетингу тощо. Система Amazon Personalize дозволяє розробникам швидко створювати та розгортати ефективні рішення, а також сегментувати клієнтів у будь-якому масштабі [10].

Таким чином, основними напрямами сучасного розвитку рекомендаційних систем як основного способу автоматизованої реалізації процесів формування та обробки заявок на отримання послуги є застосування моделей і методів штучного інтелекту для набуття таких можливостей:

– формалізація заявок, сформованих природною мовою користувача;

– підвищення рівня релевантності послуги індивідуальній заявці користувача;

– персоналізація результатів обробки заявки з урахуванням особливостей кожного окремого користувача;

– оптимізація процесів пошуку та попередньої обробки заявок.

2.2. Аналіз формальних основ автоматизації формування і обробки заявок на отримання послуги

Основу сучасних ІТ, які використовуються для автоматизованого вирішення задачі формування та обробки заявок на отримання послуги зазвичай складають методи ML, обробки природної мови (NLP) та алгоритми автоматичного оформлення заявок [11]. Ці методи інтегруються в систему автоматичного формування та обробки заявок для автоматизації та поліпшення ефективності взаємодії з клієнтами. Зокрема, методи ML та NLP допомагають розпізнавати різноманітні сценарії та вимоги користувачів і реагувати на них, підвищуючи рівень персоналізації та ефективність взаємодії з користувачами. В цілому ж застосування методів ML та NLP обумовлюється такими етапами загальної послідовності дій з формування та обробки заявок на отримання послуги [1, 11].

Етап 1. Розпізнавання та валідація даних при подачі користувачами заявки з використанням різних засобів (онлайн-форми веб-сайтів, мобільні додатки, чат-боти, голосові асистенти тощо).

Етап 2. Розпізнавання та валідація даних, введених користувачем, з використанням моделей нейронних мереж для визначення правильності введення дат, часу, імен, номерів телефонів тощо.

Етап 3. Розуміння текстового введення, яке може здійснюватися з використанням методів NLP для визначення синтаксичних та семантичних елементів мови, що сприяє правильній інтерпретації заявок.

Етап 4. Класифікація типу заявки за допомогою методів ML для визначення особливостей поданої заявки (різновид послуги, питання про ціни тощо).

Етап 5. Аналіз синтаксису та семантики текстового введення, екстракція інформації з текстового контенту з використанням методів NLP.

Етап 6. Екстракція ключової інформації з виділенням з тексту заявок ключових елементів (дати, часи, локації, основні вимоги тощо) за допомогою методів ML.

Етап 7. Аналіз ключової інформації та автоматична генерація відповідей з використанням методів NLP.

Аналіз цих етапів вказує на те, що під час дослідження можливостей вдосконалення сучасних ІТ формування та обробки заявок основну увагу бажано зосередити на подальшому

розвитку методів NLP. Застосування саме цих методів дозволяє таким IT перетворити описи заявок, зроблені природною мовою користувачів, на формальні структуровані представлення даних, які можна обробляти методами ML.

Слід зазначити, що популярність застосування методів NLP значною мірою обумовлена впровадженням у цих методах передавального навчання та навчених мовних моделей. Ці заходи розсунули межі розуміння та генерації заявок природною мовою користувачів. В цілому, останні вдосконалення мовних моделей, які застосовуються в методах NLP, схоже, зумовлені не тільки значним збільшенням обчислювальних можливостей, але також відкриттям інноваційних способів полегшення моделей при збереженні високої продуктивності. Тому пропонується зосередити увагу на такому напрямі досліджень, як вдосконалення і подальший розвиток найпопулярніших мовних моделей, які застосовуються в методах NLP.

Однією з найрозповсюдженіших таких моделей є модель двоспрямованих кодувальних представлень з трансформерів (Bidirectional Encoder Representations from Transformers, BERT). На відміну від існуючого підходу до тренування моделей розпізнавання природної мови зліва направо, цю модель розроблено для попереднього формування на основі немаркованого тексту природною мовою багатошарових представлень лівого і правого контекстів. Заздалегідь навчені представлення BERT можуть бути відрегульовані лише за допомогою одного додаткового рівня виведення [12]. Це дає змогу застосовувати модель BERT для вирішення широкого кола задач (відповідь на запитання, мовний висновок тощо) без істотних модифікацій архітектури IT, які орієнтовано на вирішення специфічних задач предметних галузей.

Архітектура моделі BERT майже ідентична архітектурі моделі OpenAI GPT. Але за результатами перевірки двох варіантів моделі BERT на завданнях бенчмарку GLUE середнє підвищення точності розпізнавання текстів природною мовою для цих варіантів перевищує значення цього показника для моделі OpenAI GPT на 4, 5% і 7,0 % відповідно. Під час перевірки на завданні MNLI з бенчмарку GLUE модель BERT у порівнянні з моделлю OpenAI GPT отримала підвищення точності розпізнавання на 4,6 % [12]. Загалом, застосування моделі BERT може значно поліпшувати методи NLP, які використовуються на різних етапах формування та обробки заявок (чат-боти для кращої взаємодії з клієнтами; аналіз відгуків клієнтів; пошук відповідної інформації тощо).

Іншою популярною мовною моделлю слід визнати модель GPT-2. Це модель-трансформер, яка досягає найкращих результатів в 7 з 8 перевірених наборів даних моделювання мови. Застосування моделі GPT-2 розглядається сучасними IT-компаніями як перспективний шлях до побудови систем обробки мови, які вчаться виконувати завдання на основі своїх природних демонстрацій [13, 14]. Зокрема, команда OpenAI демонструє, що заздалегідь навчені мовні моделі GPT-2 можуть використовуватися для вирішення подальших задач без будь-яких змін параметрів або архітектури. Загалом, модель GPT-2 формує послідовні абзаци тексту та досягає перспективних, конкурентних і найкращих результатів у найрізноманітніших задачах [13].

Навчається дана модель на великих наборах даних, які формуються з таких джерел [13]:

– веб-сторінки, які фільтруються людьми;

– очищені тексти з видаленням всіх документів Вікіпедії для мінімізації накладання навчальних та тестових наборів;

– набори даних, отримані з WebText (загальна кількість документів перевищує вісім мільйонів документів при загальному обсязі тексту цих документів 40 Гб).

GPT2 демонструє досить перспективні результати у таких напрямах вдосконалення методів NLP, як підвищення ефективності міркувань, відповіді на запитання, розуміння читання та перекладу. Але що стосується практичних застосувань, модель GPT-2 без точного налаштування далеко не придатна для використання [13, 14].

Основною перевагою моделі BERT є можливість досягання кращих показників застосування, аніж аналогічні показники у підходів до попередньої підготовки на основі авторегресивного моделювання мови. Ця перевага виникає завдяки можливості моделювання двонаправлених контекстів та попередження підготовки на основі автоматичного кодування [12]. Однак, внаслідок псування вхідних даних масками BERT нехтує залежністю між маскованими позиціями. Для подолання цього недоліку було запропоновано модель XLNet. Цю модель розробили дослідники з університету Карнегі Меллона та Google для вирішення задач обробки природної мови (розуміння читання, класифікація тексту, аналіз сентиментальності тощо). На основі моделі XLNet створено узагальнений метод авторегресивного преднавчання, який використовує найкращі результати як авторегресивного моделювання мови (наприклад, Transformer-XL), так і автокодування (наприклад, BERT), уникаючи їхніх обмежень. Експерименти демонструють, що модель XLNet перевершує BERT і Transformer-XL і досягає найкращих результатів у 18 задачах NLP [15].

XLNet може надавати допомогу компаніям із широким спектром проблем NLP, зокрема [15]:

– створення чат-ботів для підтримки клієнтів першої лінії або відповіді на запити щодо продуктів;

– аналіз сентиментальності для оцінки проінформованості та сприйняття бренду на основі відгуків клієнтів та соціальних мереж;

– пошук відповідної інформації в базах документів або в Інтернеті тощо.

Розглянуті моделі мають один загальний недолік – ускладнення параметрів реплікації та точного налаштування внаслідок значних обчислювальних витрат на навчання цих моделей. Facebook AI та дослідники університету Вашингтона проаналізували навчання моделі BERT та виявили кілька змін у навчальній процедурі, що покращують її ефективність. Зокрема, дослідники використали новий, більший набір даних для навчання, застосували для навчання моделі набагато більшу кількість ітерацій та видалили прогнозування навчальної мети. Отримана в результаті оптимізована модель RoBERTa (надійно оптимізований підхід BERT) відповідає оцінкам нещодавно представленої моделі XLNet в бенчмарку GLUE. Дана модель може використовуватися в бізнес-середовищі для широкого кола подальших задач, включаючи системи діалогу, відповіді на запитання, класифікацію документів тощо [16].

Команда Google Research займається проблемою постійно зростаючого розміру попередньо навчених мовних моделей, що призводить до обмеження пам’яті, збільшення часу навчання та інколи несподівано погіршення продуктивності. Зокрема, ця команда запропонувала модель Lite BERT (ALBERT). Дана модель базується на таких двох методах зменшення параметрів [17]:

– факторизована параметризація вбудовування, де розмір прихованих шарів відокремлюється від розміру вкладених словникових запасів шляхом розкладання великої матриці, що містить словниковий запас, на дві малі матриці;

– спільне використання параметрів між шарами, щоб запобігти зростанню кількості параметрів із глибиною мережі.

Ефективність застосування ALBERT додатково покращується шляхом введення самоконтрольованої втрати для прогнозування порядку речень для усунення обмежень BERT щодо узгодженості між реченнями. ALBERT може в подальшому покращити свою продуктивність за допомогою майнінгу, ефективнішого навчання моделі та інших підходів. Ця модель може бути використана в бізнес-налаштуваннях для підвищення продуктивності широкого кола подальших задач, включаючи підвищення продуктивності чат-ботів, аналіз сентиментальності, аналіз документів та класифікацію тексту [17].

Дослідницька група Alibaba запропонувала розширити BERT до нової мовної моделі StructBERT. Експерименти демонструють, що введена модель суттєво покращує сучасні результати вирішення різних задач із розуміння природної мови, включаючи аналіз сентиментальності та відповіді на запитання. Модель базується на архітектурі BERT з багатошаровою двонаправленою мережею трансформерів [18]. Як і інші попередньо навчені мовні моделі, StructBERT може допомагати компаніям у виконанні різноманітних задач NLP, включаючи відповіді на запитання, аналіз сентиментальності, узагальнення документів тощо.

Дослідницька група Google запропонувала уніфікований підхід до передачі навчання в NLP з метою встановлення нового рівня технології в цій галузі. З цією метою вони запропонували розглядати кожну проблему NLP як проблему «перетворення тексту в текст». Така структура дозволяє використовувати одну і ту саму модель, ціль, процедуру навчання та процес декодування для різних задач, включаючи узагальнення, аналіз сентиментальності, відповіді на запитання та машинний переклад. Дослідники називають свою модель Text-to-Text Transfer Transformer (T5) [19] і навчають її на великому наборі даних, зібраних з Інтернету, для отримання найкращих результатів в ряді задач NLP. Недоліком моделі T5 є низька продуктивність методів, які її використовують для вирішення задач NLP. Для ліквідації цього недоліку пропонується, зокрема, зосередити увагу на дослідженні ефективніших методів вилучення знань та мовно-діагностичних моделей. Втім, незважаючи на те, що T5 має мільярди параметрів і може бути занадто важкою для застосування в бізнес-середовищі, представлені ідеї можуть бути використані для поліпшення ефективності різних задач NLP, включаючи узагальнення, відповіді на питання та аналіз сентиментальності [19].

Дослідницька група OpenAI звернула увагу на той факт, що потреба у маркованому наборі даних для кожного нового мовного завдання обмежує застосовуваність мовних моделей. Ця група запропонувала альтернативне рішення, яке полягає в масштабуванні мовних моделей для поліпшення швидкодії. Цим рішенням є авторегресивна параметрична модель GPT-3 зі ста сімдесятьма п'ятьма мільярдами параметрів. Оцінка ефективності цієї моделі проводилася на понад двох десятках задач NLP. Під час оцінки GPT-3 показала багатообіцяючі результати і навіть іноді перевершувала сучасний рівень, досягнутий за допомогою відрегульованих моделей [20]. Модель GPT-3 використовує ту ж модель та архітектуру, що і GPT-2, включаючи модифіковану ініціалізацію, попередню нормалізацію та оборотну токенізацію. Однак, на відміну від GPT-2, вона використовує чергування щільних і локально смугастих розріджених моделей уваги в шарах трансформатора, як у розрідженому трансформері. Недоліком моделі GPT-3 є небхідність її вдосконалення з метою зменшення розміру великих моделей для використання у реальних програмах та підвищення ефективності вибірок при тренуванні. Слід

визнати, що сучасну модель GPT-3 важко застосувати при вирішенні реальних бізнес-задач через її непрактичні вимоги до ресурсів.

Як альтернативу існуючому підходу до попередньої підготовки таких популярних мовних моделей, як BERT та XLNet, дослідники зі Стенфордського університету та Google Brain запропонували новий підхід ELECTRA, основою якого є виявлення заміненим маркером. Цей підхід дає змогу моделі навчатися на основі змісту усіх вхідних токенів замість невеликої замаскованої підмножини. Він не є змагальним, оскільки генератор, що виробляє токени для заміни, навчається з максимальною ймовірністю. Виявлення заміненим маркером означає, що деякі токени замінюються зразками з невеликої мережі генераторів [21]. Завдяки своїй обчислювальній ефективності модель ELECTRA може зробити застосування попередньо навчених кодерів тексту доступнішим для бізнесу.

Дослідники з Microsoft Research запропонували модель DeBERTa як наслідок двох основних вдосконалень моделі BERT. Цими вдосконаленнями є механізмами розкутої уваги та вдосконалений декодер маски. DeBERTa має два окремі вектори, що представляють зміст і позицію, а самоувага розраховується між усіма можливими парами, тобто вміст до вмісту, вміст до позиції, позиція до вмісту та позиція до позиції [22]. Автори моделі DeBERTa припускають, що вона потребує інформації про абсолютну позицію, щоб зрозуміти синтаксичні відтінки, такі як характеристика суб'єкта-об'єкта. Вбудовування абсолютної позиції надається останньому шару декодера безпосередньо перед шаром softmax, який дає вихідні дані. Для збільшення узагальнення як метод регуляризації використовується віртуальний змагальний алгоритм навчання, який називається інваріантним масштабуванням. Вбудовані слова порушуються незначною мірою і вчаться видавати такий самий результат, як і для невзбурених вбудованих слів. Слова вкладання векторів нормуються до стохастичних векторів (де сума елементів у векторі дорівнює 1), щоб бути інваріантним до кількості параметрів у моделі [22].

Стислий опис переваг і недоліків розглянутих моделей наведено у табл. 1.

Таблиця 1

Опис переваг та недоліків мовних моделей

Модель	Переваги	Недоліки
BERT	Враховує контекст з обох сторін слова	Велика кількість параметрів, значні обчислювальні витрати
GPT-2	Здатність генерувати тексти без задач, заданих користувачем	Обмежена в різних задачах обробки тексту
XLNet	Враховує контекст з обох сторін слова, без обмежень	Великі обчислювальні витрати
RoBERTa	Надійно оптимізований підхід до підготовки	Потребує великих обсягів даних для тренування
ALBERT	Має менше параметрів, зберігає точність	Обмежена точність порівняно з іншими моделями
StructBERT	Додає мовні структури для кращого розуміння тексту	Вимагає додаткового часу та ресурсів для тренування
T5	Універсальний підхід до обробки тексту	Великі обчислювальні витрати
GPT-3	Велика кількість параметрів, здатність швидко навчатися	Вимагає великих обчислювальних ресурсів

Кінець таблиці 1

Модель	Переваги	Недоліки
ELECTRA	Ефективніше використання обчислювальних ресурсів	Вимагає попередньої підготовки кодерів тексту
DeBERTa	Поліпшене декодування BERT	Вимагає додаткових обчислювальних ресурсів

На основі аналізу табл. 1 можна зробити такі висновки:

– моделі BERT та її варіації (RoBERTa, ALBERT) мають великий потенціал для різних задач обробки тексту, але вимагають значних обчислювальних ресурсів;

– моделі GPT-2 та GPT-3 підходять для генерації тексту, але можуть бути обмежені у точності та обчислювальних витратах;

– моделі XLNet, T5 та StructBERT враховують контекст з обох сторін слова, але вимагають значних обчислювальних витрат;

– моделі ELECTRA та DeBERTa надають покращене використання обчислювальних ресурсів, але можуть вимагати додаткової підготовки або ресурсів для тренування.

Слід відзначити, що великі обчислювальні витрати є загальним недоліком використання мовних моделей у методах NLP, які застосовуються для автоматизованого вирішення задачі формування та обробки заявок на отримання послуги. Тому пропонується основну увагу зосередити на дослідженні можливостей вдосконалення та подальшого розвитку моделі BERT, яка може використовуватися у переважній більшості цих методів.

3. Мета і задачі дослідження

Метою даного дослідження є розробка методу автоматизованого вирішення задачі формування заявок на отримання послуги як одного з основних елементів системи управління послугами. Досягнення цієї мети дозволить поліпшити процес та результати формування заявок на отримання послуги за рахунок використання методів та засобів штучного інтелекту.

Для досягнення цієї мети у статті вирішуються такі задачі:

– розробка комбінованого методу формування заявок на отримання послуги;

– експериментальна перевірка розробленого комбінованого методу на прикладі вирішення задачі автоматизації формування заявок для веб-сайту компанії з оренди автомобілів.

4. Моделі і методи дослідження

Предметом даного дослідження є комбінований метод формування заявок на отримання послуги з урахування обмежень, характерних для предметної галузі.

Для розробки цього методу як вихідні запропоновано використовувати методи ML та модель BERT. Їх використання робить значний крок у розвитку процесу формування та аналізу заявок на різні послуги. Це важливо для власників бізнесу різних галузей через можливість оформлювати заявки на отримання послуги без участі людини.

Модель BERT допомагає швидко та ефективно досягти розуміння семантики в текстах, коли тексти розглядаються в обох напрямках, враховуючи контекст зліва і справа від кожного слова. Попередньо навчена модель BERT може бути доопрацьована за допомогою лише одного додаткового вихідного шару, що дозволяє створити високоефективні моделі для широкого спектра задач NLP.

Архітектура моделі BERT являє собою багатошаровий двоспрямований кодувальник Transformer [A1 (4)]. В основі BERT лежить стек із L ідентичних шарів трансформера. У

кожному шарі є два типи підшарів. Перший підшар реалізує механізм «багатоголовкової» уваги. Його основне завдання полягає в тому, щоб при кодуванні певного слова враховувати контекст, визначений іншими словами у послідовності. Другий підшар являє собою позиційно-орієнтовану повнозв'язну пряму мережу. Ця мережа застосовується до кожної позиції послідовності окремо і включає два лінійних перетворення. Розмірність вхідних та вихідних даних цього підшару складає d_{model}, при цьому внутрішній шар має розмірність $d_{ff}=2d_{model}$. Важливо відзначити, що цей підшар використовує функцію активації GELU, яка демонструє кращу ефективність порівняно зі стандартною ReLU у рамках кодувальника Transformer. У кожному шарі кодувальника реалізуються також і залишкові зв'язки. Вони використовуються навколо кожного з двох підшарів, після чого відбувається нормалізація шару. В результаті вихід кожного підшару є $LayerNorm(x+Sublayer(x))$, де $Sublayer(x)$ – функція, реалізована всередині підшару. Всі підшари в моделі BERT генерують вихідні дані однієї і тієї ж розмірності d_{model}, що полегшує процес залишкового зв'язування. Відзначимо, що, незважаючи на те, що лінійні перетворення однакові для різних позицій усередині одного підшару, модель BERT використовує різні параметри для різних шарів.

Суть механізму «багатоголовкової» уваги полягає в тому, що кожна «головка» може спеціалізуватися на виявленні певного типу взаємозв'язків у даних, що робить цю модель загалом ефективнішою порівняно з моделлю, яка використовує одну «головку» уваги. Цей механізм дозволяє моделі повніше і точніше розуміти контекст та семантику тексту, що суттєво підвищує її продуктивність при вирішенні різних задач NLP. Функція самоуваги реалізується паралельно на різних проєкціях матриць запиту, ключа і значення. В результаті формуються *h* різних вихідних матриць, відомих як «головки уваги». Ці «головки уваги» потім конкатенуються і проєктуються в інший підпростір представлення, що призводить до створення остаточної вихідної матриці «багатоголовкової» уваги.

Загальну архітектуру моделі BERT представлено на рис. 1 [12].

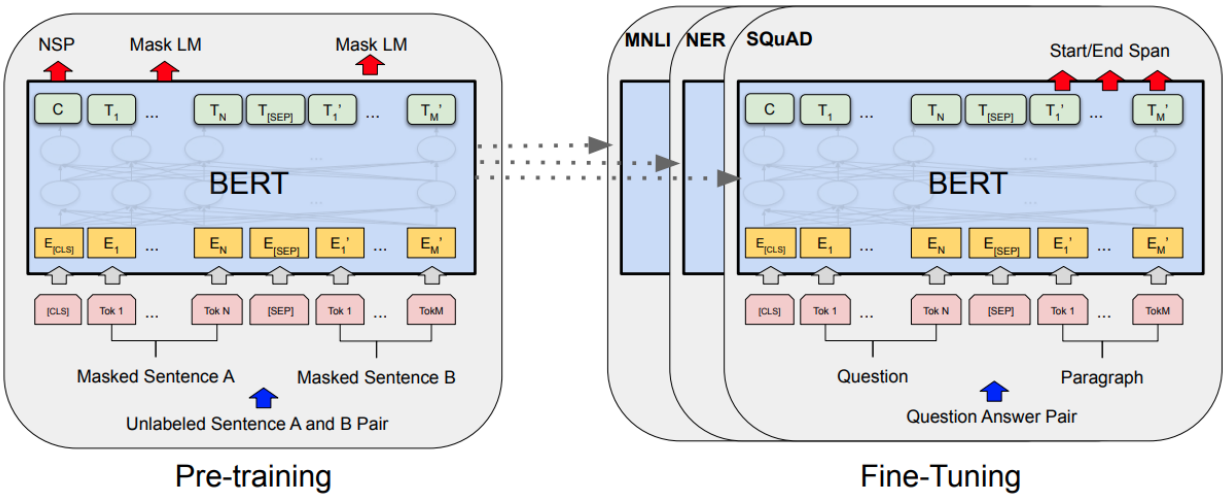

Рис. 1. Загальна архітектура моделі BERT

В процесі формування заявки на отримання послуг модель BERT може бути використано для вирішення різноманітних задач. Однією з цих задач є формування і попередня обробка

заявки, створеної користувачем у вигляді тексту, написаного природною мовою. Під час заповнення форми користувачем BERT аналізує введені дані та забезпечує формування пропозицій щодо подальшого формування та обробки заявки з мінімізацією участі користувача. Наприклад, маємо: *T* – текст форми; *V(t)* – векторне представлення токена *t*; *C* – контекстний вектор заповнення. Тоді опис контекстного вектору має вигляд

$$C = \frac{1}{|T|}\sum_{t \in T} V(t)\,. \tag{1}$$

З використанням отриманого вектору формується така пропозиція:

$$\Pr oposal(N) = \arg\max_{t \in N}(V(t) * C)\,. \tag{2}$$

Вираз (1) описує загальний результат застосування BERT Pre-traning; вираз (2) описує загальний результат застосування BERT Fine-Tuning.

Головний недолік моделі BERT полягає у великій кількості параметрів, які потребують обробки. Зокрема, значних витрат потребує створення та обробки формальних представлень шарів та підшарів трансформера BERT. Тому існує необхідність проведення досліджень, спрямованих на зменшення витрат, пов'язаних із застосуванням моделі BERT під час автоматизованого вирішення різноманітних прикладних задач в межах ІС та ІТ управління підприємствами та організаціями.

5. Вирішення задачі автоматизації процесів формування заявок на отримання послуги

5.1. Розробка комбінованого методу для вирішення задачі формування заявок

Однією з задач процесу формування заявки на отримання послуг є задача формування і попередньої обробки заявки, створеної користувачем у вигляді тексту, написаного природною мовою. Але головний недолік моделі BERT не дає змоги застосовувати цю модель у малих та середніх ІС та ІТ управління послугами підприємств або організацій. Тому авторами статті запропоновано розробити комбінований метод вирішення задачі формування та попередньої обробки заявки на отримання послуги. В цьому методі пропонується поєднати переваги застосування моделі BERT та менш витратних моделей і методів ML, які можуть надавати точніший результат в умовах обмеженого тексту опису заявки природною мовою користувача.

Результат розробки комбінованого методу вирішення задачі формування та попередньої обробки заявки на отримання послуги представимо як послідовність таких етапів.

Етап 1. Проведення аналізу текстового запиту з використанням моделі BERT.

Крок 1.1. Проведення попередньої обробки тексту перед подачею текстового запиту на вхід моделі BERT шляхом токенізації, видалення стоп-слова, лематизації тощо.

Крок 1.2. Подача тексту до моделі BERT, яка використовує контекстні вектори для представлення слів у тексті. Вектори отримуються під час BERT Pre-traning моделі на великому обсязі текстових даних

$$H = BERT\ (T), \tag{3}$$

де H – контекстні вектори, отримані як результат застосування BERT; T – вхідний текстовий запит.

Крок 1.3. Визначення ключових параметрів запиту на основі контекстних векторів

$$H = ExtractParametrs\ (H)\ , \tag{4}$$

де *H = (H1, …, Hi, …, Hm)*; *m* – кількість параметрів

Крок 1.4. Створення вектору параметрів для подальшого використання в лінійній регресії та моделі машинного навчання

$$V_{parameters} = [\ \text{Кв}]\ . \tag{5}$$

Крок 1.5. Визначення вагових коефіцієнтів для кожного параметра за допомогою лінійної регресії

$$W_{parameters} = LinearRegression\ (V_{parameters}). \tag{6}$$

Етап 2. Розрахунок вагових коефіцієнтів з використанням лінійної регресії.

Крок 2.1. Підготовка додаткових ознак для лінійної регресії на основі вектору вагових коефіцієнтів $W_{parameters}$

$$X_{linear} = PrepareFeatures\ (W_{parameters}, H). \tag{7}$$

Крок 2.2. Визначення вагових коефіцієнтів для лінійної регресії на основі підготовлених ознак

$$W_{linear} = LinearRegression\ (X_{linear}). \tag{7}$$

Крок 2.3. Розрахунок зважених ознак для подальшого використання у моделі машинного навчання

$$\hat{X}_{\ linear}\ =\ W_{linear} * X_{linear}. \tag{8}$$

Етап 3. Реалізація навченої моделі класифікації за допомогою ML.

Крок 3.1. Підготовка навчального набору даних для класифікації, який містить контекстні вектори *H*, отримані від BERT, зважені ознаки $\hat{X}_{\ linear}$ як результат лінійної регресії та відомі мітки класів*Yi* на основі типів заявок

$$D_{classification} = \{(H_i, \hat{X}_{\ linear_i}, Y_i)\}_{i=1}^{N}\ , \tag{9}$$

де H_i – контекстні вектори для і-го запиту як результат застосування BERT; $\hat{X}_{\ linear_i}$ – зважені ознаки для і-го запиту як результат лінійної регресії; Y_i – мітка класу (тип заявки) для і-го запиту.

Крок 3.2. Проведення навчання моделі класифікації з використанням навчального набору даних з використанням Support Vector Machine (SVM), Random Forest, чи іншої моделі ML

$$Model_{classification} = TrainClassificationModel\left(D_{classification}\right). \quad (10)$$

Крок 3.3. Класифікація нових запитів на основі їхніх контекстних векторів як результату застосування BERT та зважених ознак від лінійної регресії з використанням навченої моделі

$$Y_{predicted} = Model_{classification}\left(H_{new}, \hat{X}_{linear_{new}}\right). \quad (11)$$

Етап 4. Категоризація текстових описів з використанням моделі Байєса.

Крок 4.1. Збір та попередня обробка текстових даних (описи об'єктів заявки, відгуки користувачів, інші текстові ресурси тощо)

$$D_{text} = \{Text_i\}_{i=1}^{M}. \quad (12)$$

де $Text_i$ – текстові дані для і-го документа; M – кількість видів текстових документів.

Крок 4.2. Тренування моделі Байєса з використанням навчального набору текстових даних та наївного байєсівського класифікатора

$$Model_{Bayes}(Description_{rent}). \quad (13)$$

Крок 4.3. Категоризація текстових даних шляхом їх призначення відповідним типам заявок

$$Y_{Bayes} = MapCategoriesToLabels(Category_{text}), \quad (14)$$

де вектор міток Y_{Bayes} вказує на тип заявки для кожного текстового опису.

Етап 5. Надання рекомендацій користувачам системи.

Крок 5.1. Отримання результатів застосування моделі класифікації $Y_{Predicted}$ (11) та моделі категоризації текстового опису об'єкта Y_{Bayes}. (14).

Крок 5.2. Зважене поєднання отриманих результатів реалізації моделей (11) та (14), де ваги можуть бути налаштовані на основі точності та важливості кожної моделі, у вигляді комбінованого вектору

$$Y_{combined} = \alpha * Y_{predicted} + \beta * Y_{Bayes}. \quad (15)$$

де α та β – ваги для кожної моделі.

Крок 5.3. Надання рекомендацій з використанням отриманого комбінованого вектору $Y_{combined}$ для призначення рекомендацій користувачам системи щодо вибору об'єкта.

$$Recommendations = ProvideRecommendations\left(Y_{combined}\right). \quad (16)$$

5.2. Експериментальна перевірка результатів реалізації комбінованого методу для вирішення задачі формування заявок на оренду автомобілів через веб-сайт компанії

Сучасні послуги оренди автомобілів пропонують гнучку адаптацію заявок залежно від ситуацій, в яких користувач може обрати, наприклад, автомобіль з дитячим кріслом,

кондиціонером, аудіотехнікою, навігаційною системою тощо. Це робить послуги оренди або прокату автомобілю дуже популярними у сучасному світі. Але у більшості компаній, діяльність яких спрямована на надання цієї послуги, є проблеми, пов'язані з відсутністю автоматизації процесів формування заявок на оренду. Використання алгоритмів оптимізації заявки на оренду для динамічного встановлення цін в залежності від попиту має глибоке значення при процесах формування заявок.

Тому експериментальну перевірку розробленого комбінованого методу запропоновано провести на прикладі автоматизованого вирішення функціональної задачі формування заявок в межах веб-сайту компанії з прокату автомобілів.

Програмний додаток, який забезпечує автоматизоване застосування розробленого комбінованого методу, було розроблено на основі мови Python. Цей додаток включає в себе такі бібліотеки:

– Streamlit – бібліотека Python, яка використовується для швидкого створення веб-додатків. У цьому коді використовується для створення інтерфейсу користувача при рекомендації двигуна транспортного засобу;

– Langchain – користувацький пакет, який надає набір інструментів для побудови моделей природної мови. Він має кілька підмодулів, включаючи llms, які з'єднують з LLM, такими як OpenAPIs GPT;

– Azure.search.documents – бібліотека, елементом якої є клас SearchClient, що використовується для взаємодії з компонентом Azure Cognitive Search. Компонент Azure Cognitive Search використовувався як пошукова система для транспортних засобів. Дані транспортного засобу були завантажені як файли JSON;

– Scikit-learn – бібліотека, яка використовувалася для реалізації моделей і методів лінійної регресії.

Для вирішення задачі формування заявки на оренду автомобіля користувач повинен зайти на сайт і обрати функцію «Орендувати авто». Коли користувач натискає на кнопку «Орендувати авто», йому відкривається вікно, куди він повинен вписати свій запит. На цьому етапі користувачу пропонується ввести у довільній формі інформацію про поїздку (вказати терміни та місце отримання транспортного засобу, а також додати додаткові опції). Застосування розробленого комбінованого методу дозволяє користувачеві у вільному форматі і у своєму особистому стилі внести у форму заповнення запиту пояснення, що саме він шукає.

Приклад реалізації функції аналізу тексту запита користувача та вибору автомобіля наведено на рис. 2.

```python
import re

# Припустимо, що у вас є база даних автомобілів
car_database = [
    {'марка': 'Toyota', 'модель': 'Camry', 'вартість': 500, 'вмістимість': 5},
    {'марка': 'Honda', 'модель': 'Accord', 'вартість': 450, 'вмістимість': 4},
    # Додайте інші автомобілі та їх характеристики
]


def analyze_text_and_choose_car(text):
    # Аналіз тексту
    ages = re.findall(r'\b\d+\b', text)
    destination = re.search(r'\b[А-Яа-яІЇЇіЄє]+\b', text)
    start_date = re.search(r'\b\d{2}\.\d{2}\.\d{4}\b', text)
    end_date = re.search(r'\b\d{2}\.\d{2}\.\d{4}\b', text)

    # Логіка вибору автомобілів
    selected_cars = []

    for car in car_database:
        if int(ages[0]) <= car['вмістимість'] and int(ages[1]) <= car['вмістимість']:
            selected_cars.append(car)

    # Виведення результатів
    print(f"Вік дітей: {ages}")
    print(f"Місце поїздки: {destination.group() if destination else 'Невідомо'}")
    print(f"Дата початку поїздки: {start_date.group() if start_date else 'Невідомо'}")
    print(f"Дата завершення поїздки: {end_date.group() if end_date else 'Невідомо'}")

    print("\nПідходящі автомобілі:")
    for car in selected_cars:
        print(f"Марка: {car['марка']}, Модель: {car['модель']}, Вартість: {car['вартість']}")


# Зразок тексту
user_text = "Я батько, їду з двома дітьми, котрим по 8 та 10 років, до Харкова з 10.01.2024 до 17.01.2024. Підберіть мені, будь ласка, гарне авто :)"

# Виклик функції аналізу тексту та вибору автомобілів
analyze_text_and_choose_car(user_text)
```

Рис. 2. Функція аналізу тексту запиту користувача та вибору автомобіля

На першому етапі для аналізу тексту запиту, виокремлення віку дітей, місця поїздки, а також дат початку та завершення подорожі використовуються регулярні вирази. Ці дані служать вхідними параметрами для подальшого вибору автомобілів.

```
json_data = {
    "content": {
        "DOORS": 5,
        "COLOUR": "Grey",
        "VARIANT": "Dacia jogger",
        "MILEAGE": 23744,
        "YEAR": 2015,
        "PEOPLE": 5,
        "TRANSMISSION": "MANUAL",
        }
```

Рис. 3. Приклад JSON-файла

Для взаємодії з базою даних автомобілів реалізовано спеціальний механізм, який здійснює обмін інформацією про автомобілі у вигляді JSON-файлів (див. рис. 3).

Вихідною інформацією, яка формується в результаті застосування автоматизованого комбінованого методу, є інформація, отримана з поданої заявки, та список автомобілів, визнаних найрелевантнішими цій заявці, разом з характеристиками цих автомобілів. Приклад результату вирішення задачі формування заявки на оренду автомобілів наведено на рис. 4.

6. Обговорення результатів дослідження

Експериментальна перевірка розробленого комбінованого методу вирішення задачі автоматизації формування заявок для веб-сайту компанії з оренди автомобілів включала в себе підготовку даних, удосконалення алгоритмів, тестування, валідацію і аналіз результатів.

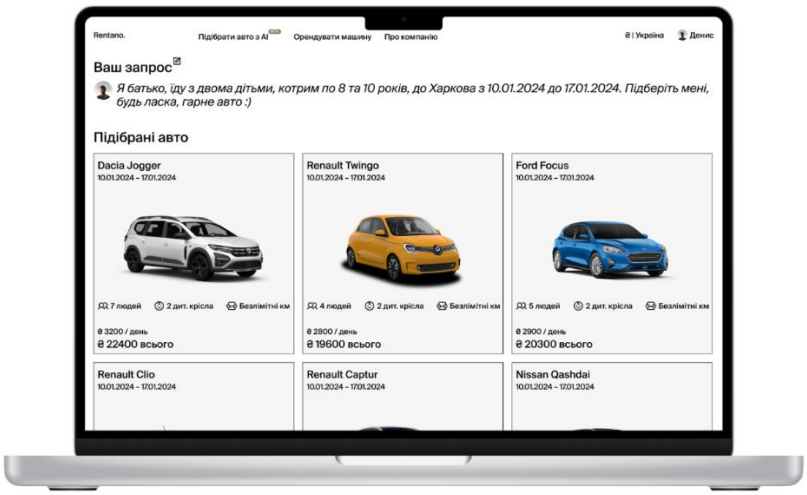

Рис. 4. Результат вирішення задачі формування заявки на оренду автомобілів

Розглянуто особливості підготовки вхідних даних під час формування заявки на оренду автомобілів. Наведено опис програмної функції, яка реалізує розроблений комбінований метод. Розглянуто особливості взаємодії цієї функції з базою даних та пошуковою системою. Продемонстровано можливість знаходження автомобілів, релевантних заявці, яку було сформовано користувачем у довільному текстовому форматі.

Розроблена програмна реалізація проходила тестування та валідацію на тестових наборах даних, які створювалися для імітації різноманітних сценаріїв запитів. Під час формування тестових наборів даних використовувались як реальні, так і синтезовані дані.

Результати, отримані під час валідації розробленої програмної функції в межах веб-сайту компанії з оренди автомобілів, показали покращення точності підбору автомобілів за заявками користувачів у порівнянні попередніми варіантами вирішення даної задачі (ручне вирішення, заповнення жорстко структурованих форм). Було також виявлено зменшення часу відгуку реалізованої функції на запити користувачів, що сприяє покращенню загального користувацького досвіду.

Головним недоліком отриманих результатів слід вважати недостатнє зменшення обчислювальних витрат на реалізацію розробленого комбінованого методу, що приводить до

появи жорсткіших вимог, які висуваються до серверів даних або хмар як IT-інфраструктури, в якій планується експлуатація веб-сайту компанії.

Виходячи з цього недоліку, головним напрямком подальших досліджень в галузі застосування методів ML та моделі BERT є дослідження зі зменшення кількості параметрів та розмірностей цих методів і моделі без значного погіршення результатів обробки.

7. Висновки

У ході даного дослідження було вирішено задачу розробки комбінованого методу автоматизованого формування заявок на отримання послуги. Розроблений метод являє собою поєднання моделі BERT з простішими методами ML та методами байєсівської класифікації. Таке поєднання зменшує обчислювальні витрати на реалізацію моделі BERT без суттєвого погіршення якості визначення послуги, якої потребує користувач.

Для експериментальної перевірки розробленого комбінованого методу було створено програмну функцію та механізми взаємодії цієї функції з базою даних та пошуковою системою. Апробація цієї функції проводилася в межах експлуатації веб-сайту компанії з оренди автомобілів. Результати апробації підтверджують можливість застосування розробленого комбінованого методу для вирішення функціональних задач формування і попередньої обробки заявок на отримання послуги у різних предметних галузях.

Як перспективи подальших досліджень запропоновано звернути особливу увагу на вивчення особливих заходів зі зменшення обчислювальної складності моделі BERT та її різновидів з метою застосування подібних моделей у різноманітних ІС та ІТ управління середніми та малими підприємствами та організаціями.

Перелік посилань:

1. Horovitz J. Service Strategy: Management Moves for Customer Results. Pearson Education. 2004. 205 p.

2. ISO/IEC 20000-1:2018. Information technology – Service management – Part 1: Service management system requirements. 2018-09-14. 31 p.

3. Система управління замовленнями RemOnline. *RemOnline*. URL: https://remonline.ua/features/order-management/ (дата звернення 20.12.2023).

4. Huang M., Pang J. Research on the Evaluation System of Online Car Service Quality. *Journal of Social Humanities*. 2023. Vol. 5. Iss. 1. https://doi.org/10.53469/jssh.2023.5(01).09 .

5. Ling L. The Design of Multifunctional Online Travel Service Management Platform and the Implementation of MySQL. *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA*). Coimbatore, India, 2022. P. 1293-1296. https://doi.org/10.1109/ICIRCA54612.2022.9985607.

6. Artificial Intelligence Powered Recommender as a Service. *Recombee*. URL: https://www.recombee.com/ (дата звернення: 18.12.2023).

7. Liang X., Kuang X., Xu Y. et al. The Construction of National Fitness Online Platform System under Mobile Internet Technology. *International Journal of System Assurance Engineering and Management.* 2021. Vol. 14. P.98-109. https://doi.org/10.1007/s13198-021-01198-5 .

8. Chang H., Shi T. Research and Design of a Job Service Platform Based on Recommendation Algorithm. *The Frontiers of Society, Science and Technology.* 2023. Vol. 5, iss. 14. P. 78-83. https://doi.org/10.25236/FSST.2023.051414

9. Enter the Era of Hyper-personalization with Experience OS. *Dynamic yield*. URL: https://www.dynamicyield.com/experience-os (дата звернення: 20.12.2023).

10. Amazon Personalize. AWS. URL: https://aws.amazon.com/personalize/ (дата звернення: 20.12.2023).

11. Battina D.S. Research on Artificial Intelligence for Citizen Services and Government. *International Journal of Creative Research Thoughts.* 2017. Vol. 5. Iss. 2. P. 769-773.

12. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL HLT 2019 – 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies – Proceedings of the Conference*. Minneapolis, 2019. Vol. 1. P. 4171-4186.

13. Zhng Q., Ding L., Liu J., Du B., Tao D. Can ChatGPT understand too? A comparative Study on ChatGPT and Fine-tuned BERT. *arXiv*. https://doi.org/10.48550/arxiv.2302.10198 (дата звернення 22.12.2023)..

14. Introducing ChatGPT Plus. OpenAI. URL: https://openai.com/blog/chatgpt-plus (дата звернення: 22.12.2023).

15. yang Z., Dai Z., Yang Y., Carbonell J., Salakhutdinov R., Le Q.V. XLNet: Generalized Autoregressive Pre-training for Language Understanding. *NIPS'2019: Proceedings of the 33rd International Conference on Neural Information Processing Systems.* Curran Associates inc., USA, 2019. Article No.: 517. P. 5753-5763. https://doi.org/10.48550/arXiv.1906.08237.

16. Liu Y. et al. RoBERTa: A Robustly Optimized BERT Pre-training Approach. *arXiv*. https://doi.org/10.48550/arXiv.1907.11692. (дата звернення 22.12.2023).

17. Lan Z., Chen M., Goodman S., Gimpel K., Sharma P., Soricut R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv*. https://doi.org/10.48550/arXiv.1909.11942 (дата звернення 22.12.2023).

18. Wang W., Bi B., Yan M., Wu C., Bao Z., Xia J., Peng L., Si L. StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding. *arXiv*. https://doi.org/10.48550/arXiv.1908.04577 (дата звернення 22.12.2023).

19. Raffel C. et al. Exploring the Limits of Transfer Learning with a United Text-to-Text Transformer. *arXiv*. https://doi.org/10.48550/arXiv.1910.10683 (дата звернення 22.12.2023).

20. Brown T.B. et al. Language Models are Few-shot Learners. *NIPS'20: Proceedings of the 34rd International Conference on Neural Information Processing Systems.* Vancouver BC Canada, 2020. Article No.: 159. P. 1877-1901. https://doi.org/10.48550/arXiv.2005.14165 .

21. Clark K., Luong M.-T., Le Q.V., Manning C.D. ELECTRA: Pre-training Text Encoders as Discriminators Rather than Generators. *arXiv*. https://doi.org/10.48550/arXiv.2003.10555 (дата звернення 20.12.2023).

22. He P., Liu X., Gao J., Chen W. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *arXiv*. https://doi.org/10.48550/arXiv.2006.03654 (дата звернення 20.12.2023).

Надійшла до редколегії 27.12.2023 р.

Левикін Віктор Макарович, доктор технічнх наук, професор кафедри ІУС ХНУРЕ, м. Харків, Україна, e-mail: viktor.levykin@nure.ua, ORCID: http://orcid.org/0000-0002-7929-515X (науковий керівник здобувача вищої освіти Діденка Дениса Олександровича).

Діденко Денис Олександрович, здобувач вищої освіти, група ІУСТм-22-1, факультет комп'ютерних наук, ХНУРЕ, м. Харків, Україна, e-mail: denys.didenko@nure.ua.

Альошкін Олексій Андрійович, здобувач вищої освіти, група УПГІТм-22-1, факультет комп'ютерних наук, ХНУРЕ, м. Харків, Україна, e-mail: oleksii.aloshkin@nure.ua.