

**ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ,
ПРАВА ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

Пояснювальна записка

до кваліфікаційної роботи на здобуття ступеня вищої освіти Бакалавр
на тему: «Локалізація візуальних і великих мовних моделей на основі
платформи LM Studio»

Виконав: здобувач вищої освіти
за освітньо-професійною програмою
Інформаційні управляючі системи
спеціальності 126 Інформаційні
системи та технології
ступеня вищої освіти Бакалавр
групи 126ІСТбд41
Засядько Д.В.
Керівник: Слюсар В.І.
Рецензент: Брикун О.М.

Полтава – 2024 року

ВСТУП

Актуальність теми кваліфікаційної роботи підтверджується необхідністю реалізації обробки природної мови (NLP) на основі нейронних мереж. Розвиток глибокого навчання та різних нейронних мереж, таких як CNN, RNN, GNN та механізми уваги, значно спростили процес розробки NLP моделей, забезпечуючи їхню здатність до складних завдань NLP. Впровадження генеративного штучного інтелекту (AI) на основі архітектури трансформерів (GPT) дозволяє працювати великим мовним моделям (LLM) з текстом і графікою, підтримувати реальні розмови, розуміти людські емоції та тон голосу, що дозволяє використовувати її в різних галузях (освіта, професійна діяльність, охорона здоров'я та ін.). Однак питання локалізації візуальних і великих мовних моделей потребує додаткових досліджень. Все це свідчить про актуальність теми роботи.

Метою кваліфікаційної роботи є підвищення ефективності обробки природної мови за рахунок використання локалізації open source LLM.

Завданнями кваліфікаційної роботи є:

- обґрунтування вибору інструментарію для реалізації локальних LLM;
- оцінка властивостей open source LLM;
- формування рекомендацій щодо локалізації візуальних і великих мовних моделей;
- техніко-економічне обґрунтування прийнятих рішень.

Об'єктом дослідження є процес обробки природної мови.

Предметом дослідження є програмні засоби локалізації візуальних і великих мовних моделей.

Методами дослідження є в рамках визначення інструментарію для локалізації LLM і техніко-економічного обґрунтування прийнятих рішень використовувався аналітичний метод досліджень, а для локалізації LLM – моделювання.

Інформаційна база кваліфікаційної роботи сформована з ресурсів, що містять інформацію про Інтернет-ресурси, що містять інформацію про NLP, LLM, нейронні мережі на основі архітектури трансформера, інструментарій для локалізації LLM.

Практична значущість роботи полягає у розробці рекомендацій щодо локалізації візуальних і великих мовних моделей – можуть бути використані для подальших досліджень за даною тематикою та при проектуванні локальних сервісів NLP.

Апробація результатів відбувалася в рамках XIX щорічної студентської наукової конференції «Сучасні інформаційні технології та інноваційні методики в економіці, менеджменті та бізнесі» Полтавського державного аграрного університету (14 травня 2024 р., м. Полтава).

За результатами досліджень здійснено публікації тез доповідей.

Структура кваліфікаційної роботи логічно пов'язана з завданнями досліджень і містить вступ, три розділи основної частини, висновки, список використаних джерел, додатки. Загальний обсяг пояснювальної записки кваліфікаційної роботи складає 64 сторінки формату А4. Вона містить 13 рисунків і 1 таблицю.

РОЗДІЛ 1

АНАЛІЗ ОСОБЛИВОСТЕЙ НЕЙРОМЕРЕЖНИХ МОВНИХ МОДЕЛЕЙ

1.1 Використання нейромережних мовних моделей для обробки природної мови

Протягом останнього року великі мовні моделі (Large Language Model, LLM) [1, 2] продемонстрували нові здібності у створенні креативного тексту, доказі теорем, прогнозуванні структур білка, відповідях на запитання щодо тексту та багато іншого. Вони є одним із наочних прикладів потенційних переваг, які штучний інтелект (Artificial Intelligence, AI) [3] може надати людству. Відповідно, обробка природної мови (Natural Language Processing, NLP) [4] – це область досліджень, яка розташована на стику інформатики, лінгвістики та AI. NLP передбачає аналіз зразків природної розмовної або письмової мови та їх обробку на комп'ютерах для вирішення якоїсь задачі або її спрощення. До прикладів використання мови, що не підпадає під поняття природної, можна віднести код, написаний мовою програмування (наприклад, Python [5]), або короткі рядки символів в електронній таблиці.

Нейромережні мовні моделі – це великі нейронні мережі [6], які навчаються передбачати наступне слово (частину слова) у тексті з урахуванням попереднього контексту. Незважаючи на простоту задачі, виявилось, що така постановка завдання призводить до появи вельми багатофункціональної нейронної мережі. Мовна модель потрібна для розуміння та генерації контенту природною мовою. Вони бувають породжувальні (генеративні) та дискримінантні. Генеративні моделі [7] – статистичні моделі, які засновані на аналізі самих даних і дозволяють створювати нові екземпляри даних. Дискримінантні моделі – моделі теж статистичні, але вирішують завдання класифікації даних [8].

Розвиток глибокого навчання призвів до широкого використання

різних нейронних мереж для вирішення задач NLP, включаючи згорткові нейронні мережі (CNN) [9], рекурентні нейронні мережі (RNN), нейронні мережі на основі графів (GNN) та механізми уваги. Ключовою перевагою цих нейронних моделей є їх здатність певною мірою спростити розробку моделі [10]. Традиційні не-нейронні підходи до завдання NLP залежать від дискретних, вручну створюваних функцій, тоді як нейромережні методи, зазвичай, використовують низькорозмірні і щільні вектори для неявного уявлення синтаксичних або семантичних аспектів мови.

Деякі дослідники навіть припустили, що мовні моделі можуть стати шляхом до створення AGI – сильного штучного інтелекту людського рівня. Припущення це виходить з того, що саме завдання передбачення наступного слова є AI-повним (що вимагає мислення) на рівні людини.

В свою чергу, на даний час, широко поширення набули авторегресійні генеративні мовні моделі на архітектурі трансформер (Generative Pre-Trained Transformer, GPT) [11].

Трансформер – це архітектура глибоких нейромереж, які як і RNN, призначені для обробки текстів природною мовою, перекладу, сумаризації, але не потребують обробки тексту по порядку. Це відкриває широкі можливості з розпаралелювання її роботи. Завдяки паралельному інтенсивному використанню процесів внутрішньої уваги конструкція перетворювача дозволяє моделі вивчати складні кореляції між вхідними та вихідними послідовностями. Модель перетворювача може збирати докладну контекстуальну інформацію, звертаючи увагу на різні вхідні компоненти протягом багатьох проходів, що підвищує її здатність до розуміння та прогнозування.

В якості прикладу такого підходу доцільно вказати GPT-4. Це велика мультимодальна LLM, яка створена OpenAI (четверта в серії GPT) [11]. Вона була випущена 14 березня 2023 р. та доступна для користувачів ChatGPT+. GPT-4 володіє мультимодальними можливостями (мультимодальність – різні

типи даних), що дозволяють працювати з текстовими та графічними на відміну від GPT-3.5, який обмежений лише текстовими входами. Крім того, модель здатна легко знімати складні зображення, такі як діаграми, меми і т.д. Ця унікальна функція стала ексклюзивною для американської компанії «Be My Eyes», яка спеціалізується на наданні візуальної допомоги людям з обмеженими можливостями зору, використовуючи силу штучного інтелекту та мультимодальні можливості GPT-4.

Покращена точність. GPT-4 демонструє значне покращення фактичної коректності в порівнянні з GPT-3.5. Модель робить менше фактичних помилок і на 40% перевершує GPT-3.5 в тестах на продуктивність. Це означає, що GPT-4 може надати більш точні та надійні відповіді на запити користувачів. Крім того, GPT-4 на 82% рідше генерує відповіді із забороненим або послаблюючим змістом. Це робить модель більш безпечною та дружньою для користувачів, знижує ризик неприйняттого контенту в результатах генерації.

Керованість. GPT-4 відрізняється підвищеною керованістю, що дозволяє моделі адаптуватися до вимог користувача і легше регулювати свою поведінку. На відміну від попередніх версій, де особистість, тон, багатомовність і стиль були зафіксовані, GPT-4 пропонує гнучкість у налаштуваннях цих параметрів. З GPT-4 ви можете вказати стиль і завдання ШІ в поле «Система», що допомагає моделі будувати досвід і генерувати більш якісні результати, що відповідають вашим вимогам. Користувачі отримують контент, який краще відповідає їх очікуванням і перевагам.

Найкраща успішність. GPT-4 демонструє здібності до міркування та навички вирішення проблем, що дозволило йому досягти значних успіхів у складних іспитах, таких як «Єдиний іспит для адвокатів», «LSAT для юристів» та «Біологічна олімпіада». Іншими словами, GPT-4 наближається до людського рівня в академічному та професійному плані. Таке покращення успішності свідчить про те, що GPT-4 може бути застосований в освітніх та професійних контекстах, де потрібне глибоке розуміння та вирішення

складних завдань (рис. 1.1).

У свою чергу, 13 травня 2024 р. відбулася презентація нової моделі GPT-4 Omni (ChatGPT-4o). Це флагманська модель робить природнішим процес спілкування людини з AI.

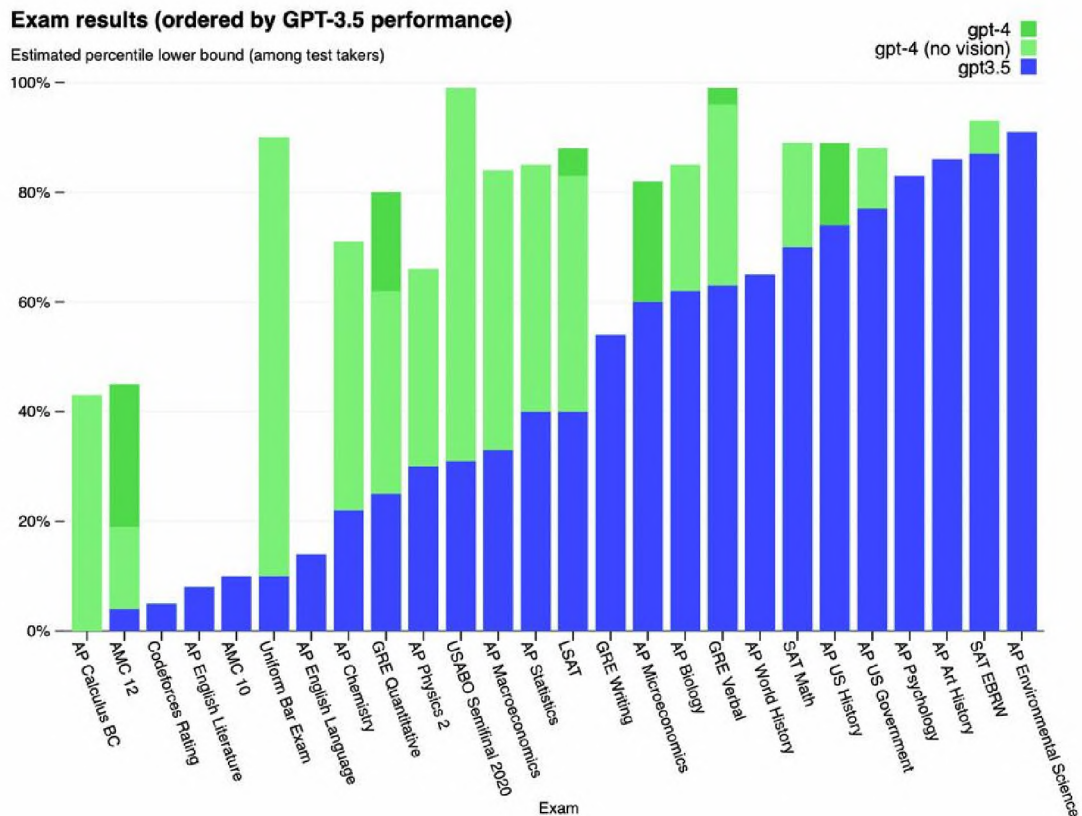


Рисунок 1.1 – Успішність GPT-4

Отримати доступ до ChatGPT-4o можна безкоштовно. Після реєстрації та налаштування платіжної системи треба виконати вхід до свого облікового запису і перейти до розділу GPT-4 Omni. Потім можна почати взаємодію з моделлю, вводячи запити у текстове поле інтерфейсу. GPT-4o буде доступний у вигляді програми для ПК. Нова модель GPT вміє сприймати, аналізувати та обробляти в режимі реального часу відео, аудіо та текст. ChatGPT-4o може підтримати розмову голосом у режимі реального часу. AI розуміє людські емоції та тон голосу, а також зможе відповідати різною інтонацією. Є кілька прикладів використання моделі ChatGPT-4o.

1. Помічник. ChatGPT-4o може бути репетитором, помічником на дзвонах, перекладачем та ін.

2. Універсальний помічник. GPT-4o вмiє створювати картини, розпізнавати текст, генерувати 3D-моделі та ін.

3. Гід. Через камеру смартфона ChatGPT-4o може розпізнати, де знаходиться людина та надати їй інформацію на запит. Наприклад, можна взяти чат-бот на прогулянку містом і попросити AI розповісти історію місця. При наведенні камери на пам'ятки ChatGPT-4o зможе розпізнати об'єкт та розповісти про нього.

На даний час, існує кілька конкуруючих до ChatGPT-4 моделей LLM:

- Llama 3 (к. Meta) [12];
- Gemini (к. Google) [13];
- ERNIE Bot (к. Baidu) [14];
- Claude 3 (к. Anthropic) [15];
- xAI (модель від к. Ілона Маска) [16].

При цьому існують відомості про розміри найбільш поширених LLM:

- ChatGPT – 175B;
- GPT-4 – ~500B;
- Llama 2 chat – 7B, 13B, 70B;
- Saiga 2 – 7B, 13B, 70B;
- StableBelluga – 7B, 13B, 70B;
- Qwen – 0.5B, 1.8B, 4B, 7B, 14B, 32B, 72B, 110B;
- FusionNet – 13B, 61B.

В цілому, основна цінність LLM полягає не тільки в умінні ChatGPT переписувати, перефразувати та перекладати тексти. Перші два пункти взагалі лише шкодять суспільству, а з перекладом уже давно непогано справляється Google Translate [17]. Справжня ж цінність ChatGPT – в аналізі найбільшого масиву інформації та пошуку в ньому тієї частини, яка необхідна досліднику, і пояснення її «мовою», яка потрібна. ChatGPT не

повинен і не може бути автором чи співавтором тієї чи іншої наукової роботи. Ця система є інструментом у роботі вченого. Так само, як, наприклад, пошукова система Google, більш просунуті системи, як Wolfram Alpha, системи цитування та індексування Web of Science та Scopus. Автор завжди повинен перевіряти та брати на себе відповідальність за той матеріал, який він створює незалежно від того інструменту, яким він користується.

1.2 Відмінності LLM Bing AI і ChatGPT

Bing AI [18] являє собою систему ШІ, створену для розумного пошуку та надання відповідей на запити користувачів. Він інтегрований у пошукову систему Bing та допомагає користувачам отримувати релевантні та точні результати пошуку. З іншого боку, ChatGPT є потужною моделлю генерації тексту, яка може використовуватися для різних завдань, включаючи створення чат-ботів, генерацію контенту та відповіді на запитання.

Навчання та дані: Bing AI заснований на обробці великого обсягу структурованих та неструктурованих даних, зібраних з вебсторінок. Він використовує алгоритми ML для аналізу та класифікації інформації з метою надання найбільш релевантних результатів пошуку. ChatGPT навчається на широкому спектрі текстових даних, включаючи книги, статті, блоги та інші джерела. Він прагне зрозуміти контекст та семантику тексту, щоб генерувати зв'язкові та граматично правильні відповіді на запити. ChatGPT здатний генерувати зв'язкові та інформативні відповіді різні запити.

Взаємодія з користувачем: Bing AI надає можливість користувачам ставити запитання та отримувати текстові результати пошуку. Він обробляє запити та надає інформацію на основі пошукових алгоритмів та індексації вебсторінок. З іншого боку, ChatGPT може використовуватися для створення інтерактивних чат-ботів та систем, де користувачі можуть ставити запитання та отримувати відповіді у природній мові. Він здатний генерувати діалоги та

проводити тривалі бесіди, імітуючи природний стиль спілкування.

Доступність та використання: Bing AI доступний через пошукову систему Bing та відповідне API, що дозволяє розробникам інтегрувати його функціональність у свої програми та сервіси. Користувачі можуть отримати результати пошуку, відповіді на запитання та іншу інформацію через інтерфейс Bing або програми, інтегровані з Bing AI. ChatGPT, у свою чергу, доступний через платформу OpenAI і надає API, яке дозволяє розробникам інтегрувати його у свої програми та сервіси для створення чат-ботів або інших систем, що базуються на текстовому введенні та виведенні.

Bing AI і ChatGPT – це дві різні моделі з різними функціоналами та цілями. Bing AI надає результати пошуку та відповіді на запити користувачів у рамках пошукової системи Bing, тоді як ChatGPT пропонує можливість генерувати текстові відповіді та використовувати їх у чат-ботах та інших системах, де важлива взаємодія природною мовою. Обидві моделі мають свої унікальні переваги та широкий потенціал застосування у різних галузях інформаційних технологій.

1.3 Аналіз кейсів практичного використання LLM

Щоб сформулювати уявлення про можливості використання LLM необхідно проаналізувати кейси їх практичного використання.

Google створила AI, який ставить діагнози точніше за живих лікарів. Google Research та DeepMind оголосили про вражаючі результати LLM Med-Gemini [19], орієнтованої на постановку діагнозів та формування планів лікування. LLM може обробляти інформацію з відео, аудіо, зображень та текстових джерел. При цьому модель враховує контекст і може розмірковувати, спираючись на нього. Крім того, Med-Gemini доступний пошук актуальної інформації в мережі, завдяки чому AI завжди знає про найпередовіші методи лікування. У стандартному тесті отримання медичної

ліцензії США точність Med-Gemini становила 91,1 %. Модель перевершила медичне сімейство моделей GPT-4. В узагальненні даних та складанні планів лікування AI Google виявився кращим за живі докторів.

У Китаї AI-лікарі натренувалися на AI-пацієнтах лікувати пацієнтів-людей краще, ніж люди-лікарі [20]. У китайському симуляторі лікарні E-лікарі (у їхній ролі виступають автономні агенти на базі великих мовних моделей – AALLM) проводять лікування E-пацієнтів (у їхній ролі інші AALLM), які «хворіють» на реальні людські респіраторні захворювання (динаміку яких моделюють також LLM, що мають доступ до великої бази медичної інформації, отриманої під час лікування реальних пацієнтів).

Мета імітаційного експерименту – дати можливість E-лікарям при лікуванні E-пацієнтів набиратися знань, щоб навчитися краще лікувати хвороби реальних людей у реальному житті.

У ході короткого експерименту E-лікарі пролікували 10 тис. E-пацієнтів (на що в реальному житті пішло б мінімум два роки). Результат був досить вдалий. E-лікарі, що підвищили свою кваліфікацію в ході цього імітаційного експерименту, досягли високої точності 93,06 % у підмножині набору даних MedQA, що охоплює основні респіраторні захворювання.

AI допоміг знизити споживання ліків. Фахівці з Mass General Brigham MESH [21] виявили, що чат-боти на основі AI можна застосовувати для контролю та обмеження прийому ліків. Вчені заявили, що 40 % людей похилого віку стикалися з поліпрагмазією – одночасне використання пацієнтом кількох ліків для лікування одного захворювання.

Спеціалісти застосували ChatGPT, щоб допомогти відмовитися від низки препаратів без шкоди здоров'ю пацієнта.

В рамках дослідження медици використовували різні клінічні сценарії та задали чат-боту низку питань для прийняття рішень. ChatGPT аналізував склад ліків та скасовував ті, які були неефективними при лікуванні. Автори зазначили, що інструмент AI був обережним у прийнятті рішень та часто зберігав режим лікування без змін.

Apple відновила переговори з OpenAI щодо впровадження їх технологій у iOS 18 [22]. Apple Inc. залучає OpenAI щодо використання технології компанії для нових функцій iPhone, які будуть випущені пізніше цього року. Компанії обговорюють умови можливої угоди та те, як технологія OpenAI буде інтегрована в iOS 18. Наступна версія ОС iPhone буде включати функцій базі власної LLM від Apple, але компанія також шукає партнерів для допомоги у впровадженні функції чат-бота. У минулому Т. Кук говорив, що особисто використовує ChatGPT від OpenAI, але також зазначав, що є «низка проблем, які потрібно вирішити».

Свіжий апдейт Magnific AI [23] дозволяє генерувати картини за ескізами (рис. 1.2). Magnific AI оновився і тепер вміє перетворювати грубі начерки на шедеври (рис. 1.3).



Рисунок 1.2 – Вихідний ескіз

Розробники Magnific AI стверджують, що за допомогою нових функцій можна створювати ілюстрації, відеоігри, портрети людей та тварин, а також проектувати дизайн інтер'єру.



Рисунок 1.3 – Згенеровані зображення по вихідному ескізу

Logitech представила мишу з ChatGPT всередині – Signature AI Edition M750 [24]. Її особливістю стала кнопка, що дозволяє запустити сервіс для роботи з неймережами Logi AI Prompt Builder. Кнопку для виклику помічника розмістили біля коліщатка прокручування. Logi AI Prompt Builder пропонує користувачу готові prompts, наприклад, перефразувати абзац тексту, зробити його коротшим або переписати. Також є запити для короткого

переказу статті, написання e-mail та ін. Спеціальні налаштування дозволяють змінити тон повідомлення та його складність.

Duolingo Max використовує GPT-4, який дозволяє студентам навчатися через ігри «Поясніть мою відповідь» та «Рольова гра», в основі яких – генеративний AI від к. OpenAI [25]. Наприклад, чат-бот пояснюватиме помилки у відповідях користувачів, наводитиме приклади, будуватиме діалоги з вигаданими персонажами.

Air India стала першою авіаційною компанією в Індії, яка заявила про використання ChatGPT для покращення сервісу на сайті [26]. Робота GPT4 спрямована на модернізацію веб-сайту та мобільних додатків, зручну систему повідомлень для клієнтів, портал обслуговування клієнтів з відстеженням запитів до служби підтримки в режимі реального часу, цифрового маркетингу, модернізацію контакт-центру, управління збоями та переобладнання самообслуговування.

Розробник CRM-системи представив свій Einstein GPT – інструмент, заснований на чат-боті OpenAI [27]. Він допомагає співробітникам складати електронні листи, планувати зустрічі, автоматично створювати особисті відповіді в чаті клієнтам.

Сервіс Be My Eyes для людей з обмеженими можливостями зору тепер використовує передові технології GPT-4 і API бачення, щоб допомогти користувачам відповісти на питання від простого читання тексту до складніших, наприклад, вибору між різними шаблонами [28].

Використовуючи GPT-4 та Vision API, люди з порушеннями зору тепер можуть отримати доступ до інформації та послуг, які раніше були недоступні їм, що може значно покращити їхню якість життя [29].

Наведені кейси дозволяють зробити висновки про основні напрямки застосування LLM. В якості такого варіанту можна розглянути таке завдання як, створення нейро-консультанта з функціоналом технічної підтримки за внутрішніми документами та правилами. Метою є розробка нейро-консультанта, який знайомий з усією внутрішньою документацією,

правилами та інструкціями, що допомагає реалізувати швидкий доступ до відповідей на будь-які внутрішні питання співробітників. Створення віртуального помічника для відповіді на запитання клієнтів компанії в чаті з урахуванням запитів та потреб клієнтів. Тобто, нейро-консультант дозволить клієнтам отримувати актуальну інформацію щодо продуктів та послуг компанії. Автоматизація рутинних процесів контролю роботи менеджерів та забезпечення цілодобового обслуговування клієнтів, навіть у неробочий для співробітників час. Ї буде аналізувати текстові документи, дані з аудіозаписів дзвінків, надаючи звіти про дотримання скриптів, правильне озвучування характеристик продуктів, закриття заперечень та виявлення потреб клієнтів.

Таким чином, використання LLM може бути в нагоді при створенні:

- нейро-помічника з інформації, розміщеної на сайті;
- нейро-консультанта з внутрішніх документів;
- нейро-консультанта для користувачів системи документації компанії;
- нейро-консультанта, що відповідає на запитання клієнтів щодо послуг, що надаються;
- нейро-консультанта для відповідей на запитання клієнтів.

По суті, LLM, що навчена на базі знань компанії, яка миттєво дізнається всю інформацію про вашу компанію та продукти, відповідає на запитання клієнтів за 12-16 секунд.

1.4 Обґрунтування необхідності локалізації великих мовних моделей

Використання локальної версії LLM може бути обґрунтовано кількома наступними важливими чинниками. Конфіденційність даних. При роботі з чутливими або конфіденційними даними критично важливо забезпечити їхню безпеку. Локальне використання LLM дозволяє повністю контролювати дані,

за винятком ризику їх витоку при передачі через інтернет-сервіси. Це особливо важливо для організацій, які працюють у таких галузях, як охорона здоров'я, фінанси або державні органи.

Низька затримка та висока доступність. Локальна версія LLM забезпечує мінімальні затримки при обробці запитів, оскільки виключаються затримки мережі і залежно від доступності віддалених серверів. Це дозволяє покращити продуктивність додатків і підвищити користувацький досвід.

Налаштування та адаптація під конкретні завдання. Локальна версія моделі надає можливість гнучко налаштовувати та донавчати модель під специфічні завдання та вимоги організації. Це дозволяє досягати більш високої точності та ефективності при вирішенні конкретних бізнес-завдань.

Економічна ефективність. У довгостроковій перспективі локальне використання LLM може бути економічно вигіднішим, оскільки виключаються регулярні витрати на підписки і платні API-запити до хмарних сервісів. Це особливо актуально для великих організацій із високим обсягом запитів.

Незалежність від зовнішніх послуг. Використання локальної версії LLM виключає залежність від зовнішніх постачальників послуг, що забезпечує більшу автономію та стабільність бізнес-процесів. Це також захищає від можливих збоїв чи змін у політиці постачальників.

Відповідність законодавству та нормативним вимогам. У деяких країнах та галузях діють суворі нормативні вимоги щодо зберігання та обробки даних. Локальне використання LLM допомагає відповідати таким вимогам, уникаючи проблем із регуляторами.

Підтримка локальних мов та особливостей. Локальні версії моделей можуть бути адаптовані для підтримки регіональних мов, діалектів та специфічних особливостей комунікації, що є важливим для забезпечення релевантності та точності відповідей.

Мова йдеться також про розширення меж переходу до туманних та кінцевих обчислень в інтересах підвищення інтелектуального функціоналу пристроїв, що входять до складу екосистем Інтернету речей (Internet of

Things). Якщо це можна зробити на цій крайній межі обчислювальної шкали, то це можна зробити і на будь-якому рівні між Raspberry Pi і великим та потужним серверним графічним процесором. Традиційно периферійний штучний інтелект тісно пов'язаний із комп'ютерним зором. Вивчення розгортання LLM і VLM на периферії додає захоплююче вимірювання в цю область, що тільки що зароджується.

Локальне використання LLM забезпечує високий рівень безпеки даних, продуктивність, гнучкість налаштування, економічну ефективність та незалежність. Це робить такий підхід кращим для організацій, які прагнуть максимальної ефективності та надійності власних IT-рішень.

РОЗДІЛ 2

ДОСЛІДЖЕННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ З МОЖЛИВІСТЮ ЛОКАЛІЗАЦІЇ

2.1 Техніки нейромережних моделей в завдання обробки природної мови

Як відомо, найпростішими завданнями NLP є: перевірка орфографії, підказка синонімів і пошук за ключовими словами [30]. Ці прості завдання досить легко вирішуються за допомогою детермінованого коду, заснованого на правилах, наприклад, з довідкових словників або предметних покажчиків. Моделі глибокого навчання невиправдано складні для таких випадків.

До завдань середнього рівня NLP відносяться: визначення рівня складності документа, прогнозування найбільш ймовірних наступних слів при введенні запиту в пошуковій системі, класифікація документів і вилучення з них такої інформації, як ціни або іменовані сутності. Ці варіанти NLP середньої складності добре підходять до застосування моделей глибокого навчання.

До найскладніших завдань NLP відносяться: машинний переклад (автоматичний вибір відповідей на запитання та чат-боти. Складність у цих сферах обумовлена необхідністю враховувати масу тонкощів (наприклад, гумор). Відповідь на запитання може залежати від проміжних відповідей на попередні питання, і зміст питання може передаватися довгим уривком тексту, що складається з багатьох речень. Вирішення складних завдань NLP, подібних до цих, виходить за рамки цієї книги; однак основи, які ми закладемо, допоможуть вам продовжити розвиток у цих напрямках. Для підвищення точності моделювання до даних природною мовою можна застосувати кілька видів попередньої обробки. На практиці, зазвичай, виконуються кілька основних технік щодо підвищення точності нейромережних моделей, які надалі доцільно дослідити.

Видалення стоп-слів. Стоп-словами називають слова, що часто зустрічаються, які, як правило, не мають великого значення, наприклад «і», «що», «який» і «тому». Щодо точного переліку стоп-слів немає єдиної думки, і в залежності від сфери застосування буває доцільно вважати (або не рахувати) певні елементи стоп-словами. Наприклад, розглядаємо модель класифікації відгуків про фільми на позитивні та негативні. Деякі списки стоп-слів включають заперечення, такі як «ні», «ні», і «не може», що може ускладнити визначення сенсу відгуку про фільм, тому такі слова, ймовірно, не слід вважати стоп-словами.

Перетворення всіх символів у нижній регістр. Слова на початку речень, що починаються з великої літери (наприклад, Вона) мають те ж значення, що і в середині речення (вона). Перетворюючи всі символи в корпусі в нижній регістр, ми ігноруємо поділ на великі та малі літери.

Лексемізація (tokenization) – весь документ (наприклад книга) розбивається на список окремих елементів мови (наприклад, слів), які називають лексемами.

Видалення розділових знаків: зазвичай облік розділових знаків не додає точності моделі природної мови, тому їх часто просто видаляють.

Стемінг (stemming) – це усічення слів до їх основи (stem). Наприклад, слова «дім» та «домашній» мають одну основу – «дім». Стемінг, особливо для невеликих наборів даних, може бути вигідним, тому що дозволяє об'єднати слова з однаковими значеннями в одну лексему. У цьому випадку з'являється більше прикладів контексту лексеми, що дозволяє таким методам, як word2vec або GloVe, точніше визначати відповідне розташування лексеми у векторному просторі слів.

Обробка n -грам: деякі слова використовуються разом настільки часто, що їх комбінацію краще розглядати як єдине поняття, ніж кілька окремих понять. Наприклад, New York – це біграма (n -грама з довжиною, що дорівнює двом), New York City – триграма (n -грама з довжиною, що дорівнює трьом). Об'єднання слів new, york та city має певне значення, яке,

ймовірно, краще зафіксувати у вигляді однієї лексеми (і однією точкою у векторному просторі слів) замість трьох.

Надалі може використовуватись алгоритм навчання без вчителя – word2vec. На його вхід можна передати будь-який набір даних природною мовою.

Сумаризація – це процес створення короткого, стиснутого викладу великого обсягу інформації, зберігаючи у своїй основний зміст вихідного тексту чи даних. Мета сумаризації – надати читачеві чи слухачеві короткий та ясний огляд основної інформації, не змушуючи його переглядати чи слухати всі деталі вихідного матеріалу. Існує два основних типи сумаризації.

Екстрактивна сумаризація – з вихідного тексту виходять найважливіші пропозиції або фрагменти, які потім комбінуються для створення стислого викладу. Це схоже на виділення найважливіших частин тексту маркером.

Абстрактивна сумаризація включає переформулювання і перефразування вихідного тексту для створення нового, стисненого викладу. Такий підхід вимагає більш глибокого розуміння вихідного тексту і може бути складнішим у реалізації, особливо для машинних методів.

Токен в контексті моделей мови, таких як GPT відноситься до мінімальної одиниці тексту, яка може бути оброблена моделлю. Токен може бути словом, частиною слова або навіть одиночним символом, залежно від схеми токенизації, що використовується.

Наприклад, пропозиція «ChatGPT is great!» може бути розбито на такі токени: [«Chat», «G», «PT», «is», «great», «!»]. Зауважте, що пробіли також враховуються у деяких токенах. Це лише один із можливих способів токенизації, і реальне розбиття може відрізнятися залежно від методу, що використовується.

Токенизація необхідна для того, щоб модель могла обробляти та генерувати текст. Коли ви подаєте текст на вхід GPT, він спочатку розбивається на токени, потім ці токени перетворюються на числові значення (зазвичай за допомогою словника або векторного уявлення), які потім

обробляються моделлю. Також важливо відзначити, що більшість моделей мають обмеження на максимальну кількість токенів, які можуть бути оброблені за одну ітерацію. Наприклад, ранні версії GPT-3.5 має ліміт 4096 токена за раз.

Chunk. У контексті GPT, «chunk» або «фрагмент» часто згадується щодо довжини введення або виведення тексту. У моделях GPT є обмеження на максимальну кількість «токенів», що вони можуть обробляти за одну ітерацію. На момент останнього оновлення для GPT-4 цей ліміт складав 4096 токенів.

Багато досліджень проведених останнім часом показало, що попередньо навчені моделі (Pre-Trained Models, PTM) великих корпусів текстів можуть навчитися універсальним мовним уявленням, які корисні наступних завдань NLP і дозволяють уникнути навчання нової моделі з нуля. З розвитком обчислювальної потужності, появою глибоких моделей та постійним удосконаленням навичок навчання архітектура PTM просунулась від поверхневої до глибокої архітектури.

Prompt. По суті процес виглядає так: ми подаємо на вхід моделі якийсь текст із запитом, а вона до нього дописує своє продовження. Якщо це продовження (генерація) збігається з нашими очікуваннями – модель, виходить, вирішила поставлене перед нею завдання. Той текст, що ми подаємо на вхід, називається prompt (або «запит»). Чим точніше він описує, що ми хочемо, тим краще модель зрозуміє, що потрібно робити. Крім цього, рекомендується додавати приклади.

Без опису мети та без прикладів у prompt, модель теж зазвичай розуміє проблему, але пропонує не такі добрі рішення (з погляду їхньої якості). Можна сказати, що деталізований prompt дозволяє GPT краще оцінити ймовірності слів, які потрібно генерувати як відповідь, спрямовуючи її в «потрібне русло». Але наскільки складним має бути prompt? І наскільки модель за розумінням близька до нас? Також можна використовувати ключову

фразу – «Let's think step by step» (давайте подумаємо крок за кроком). Виявилося, що це спонукає модель міркувати послідовно, робити висновки на основі власних суджень, і дійти правильної відповіді набагато частіше, ніж без цієї фрази. Причому цей «режим міркування» з'явився у «великій» моделі GPT-3 після подолання планки в сотню мільярдів параметрів. Більш детальну інформацію про prompt можна отримати з інструкції [31]. Тут варто звернути увагу на такий параметр як температура (рис. 2.1).

Спробуйте ще раз надіслати той самий запит кілька разів із температурою, встановленою на 1.

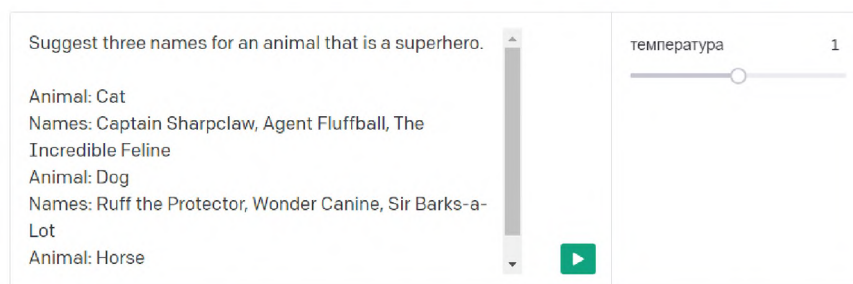


Рисунок 2.1 – Вибір параметру «температура»

Можна помітити, що якщо надіслана та сама підказка кілька разів, модель завжди повертатиме ідентичні або дуже подібні завершення. Це тому, що температура встановлена на «0». Якщо спробувати ще раз надіслати той самий запит кілька разів із температурою, встановленою на «1». Коли температура вище 0, надсилання того самого запиту щоразу призводить до різних завершень. Варто пам'ятати, що модель передбачає, який текст, найімовірніше, слідуватиме за попереднім текстом. Температура – це значення від 0 до 1, дозволяє контролювати, наскільки впевненою має бути модель під час створення цих прогнозів. Зниження температури означає, що буде менше ризиків, а завершення буде більш точним і детермінованим. Помірна температура 0,6 має добре працювати.

Принципи «zero-shot», «one-shot» та «few-shot» навчання відносяться до методів використання попередньо натренованих моделей на нових завданнях без додаткового або з мінімальним додатковим навчанням. У

контексті GPT ці принципи можна застосувати наступним чином. Zero-shot навчання. Модель виконує нове завдання без жодних прикладів того, як це завдання має бути виконане. Вона покладається лише на свої загальні знання та розуміння завдань, здобуті під час попереднього навчання. Як приклад, можна розглянути наступний випадок. Коли ви запитуєте GPT: «Напиши твір про штучний інтелект», модель не бачила жодного прикладу написання творів під час конкретного тренування на цьому завданні, але вона знає, що таке твір і як писати про штучний інтелект завдяки широкому попередньому навчанню на текстах.

One-shot навчання. Модель отримує одне або дуже обмежене число прикладів для виконання завдання. Цього достатньо, щоб модель зрозуміла, як виконати нове завдання.

Ви надаєте GPT один приклад запитання та відповіді, наприклад: «Запитання: Яке населення Франції? Відповідь: 67 мільйонів», а потім запитуєте: «Яке населення Німеччини?» Модель використовує цей єдиний приклад, щоб зрозуміти формат запитань та відповідей.

Few-shot навчання. Модель отримує декілька прикладів (більше одного, але не надто багато) для виконання завдання. Це дозволяє моделі краще зрозуміти патерни і вимоги завдання.

Ви надаєте GPT кілька прикладів запитань і відповідей: «Запитання: Яке населення Франції? Відповідь: 67 мільйонів», «Запитання: Яке населення Італії? Відповідь: 60 мільйонів" і так далі. Потім запитуєте: «Яке населення Німеччини?»² Модель використовує ці приклади, щоб точніше передбачити відповідь.

Таким чином, основна ідея полягає в тому, що модель використовує свій попередній досвід та мінімальний новий навчальний матеріал для виконання нових завдань, причому кількість цього матеріалу варіюється від жодного (zero-shot) до кількох прикладів (few-shot).

2.2 Відкриті мовні моделі

Якщо кілька років тому поява нової LLM з відкритим вихідним кодом (open source) була важливою подією в IT-світі, то сьогодні вже нікого не здивуєш. Щомісяця з'являються десятки open source мовних моделей, а щороку сотні. Щоб зорієнтуватися в цьому різноманітті, необхідно дослідити актуальні open source LLM.

Існують сотні open source LLM. Але це самостійні проекти – більшість їх розроблено з урахуванням кількох LLM, що мають назву базових моделей (Foundation Models). Створення та навчання такої моделі потребує великих фінансових витрат та обчислювальних потужностей. Тому робота над ними доступна лише великим науковим колективам та IT-компаніям: Google, OpenAI та ін. Наприклад, навчання GPT-3 обійшлося розробникам майже у 5 млн. доларів.

Foundation Models – це штучна нейромережа, навчена на великому обсязі даних, яку можна налаштувати для вирішення будь-яких завдань.

Після розробки нова модель випускається під закритою (пропрієтарною) або відкритою ліцензією (open source). В останньому випадку, інші компанії та окремі ентузіасти можуть доопрацювати та налаштувати її для вирішення своїх завдань. Це не потребує великих витрат та обчислювальних ресурсів. Тому саме з open source LLM часто працюють стартапи. Такі модифіковані моделі називають форками (fork – розвилка). Наприклад, до популярних базових моделей, що стали основою для відкритих LLM, відносять:

- LLaMA 2 – розроблена Meta спільно з Microsoft;
- BLOOM (BigScience Large Open-Science Open-Access Multilingual Language Model) – від проекту BigScience, створеного за участю компанії Hugging Face;
- GPT-2, викладена OpenAI кілька років тому, коли компанія планувала розробляти лише open-source-рішення;

- Falcon – новітня технологія від Інституту технологічних інновацій (ТІІ) з Абу-Дабі (ОАЕ);
- Сімейство моделей T5 від компанії Google.

Сучасні LLM-моделі можна представити у вигляді генеалогічного дерева, відстеживши їх еволюцію та взаємозв'язки (рис. 2.1) [32].

В даному випадку, open source моделі представлені зафарбованими прямокутниками, а пропріетарні – незафарбованими.

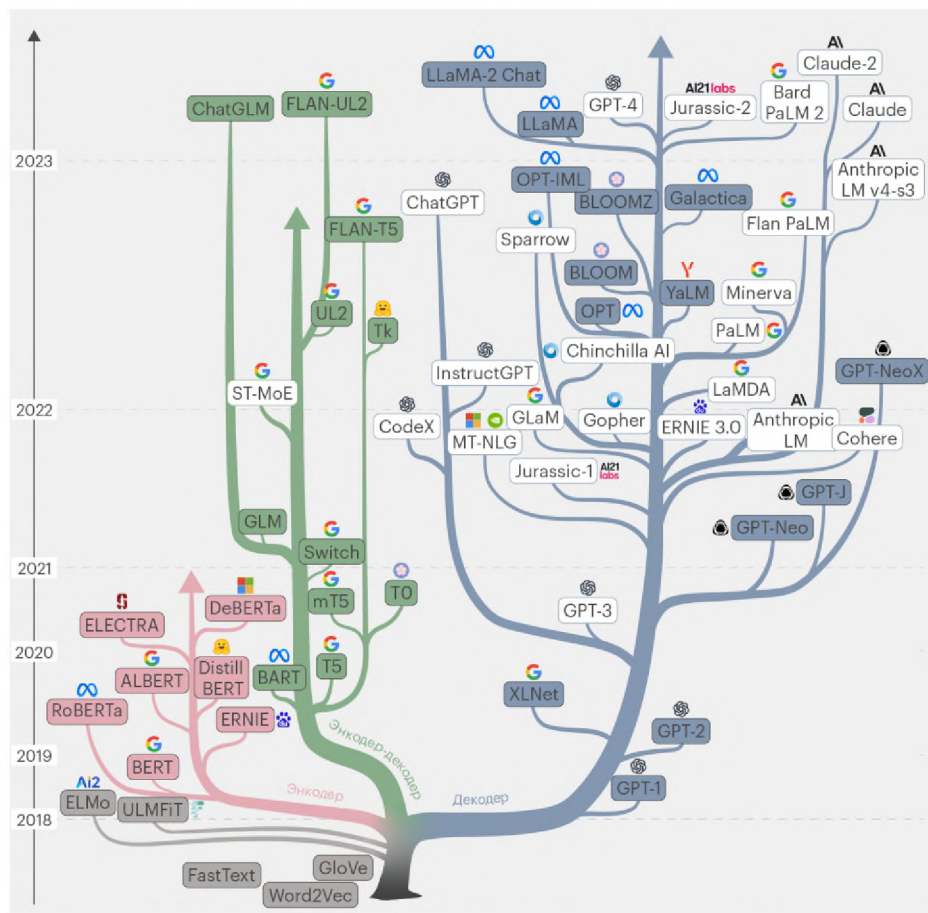


Рисунок 2.1 – Еволюційне дерево LLM

У LLM з відкритою ліцензією ті ж проблеми, що й у пропріетарних нейронках: часті галюцинації, обмеження довжини контекстного вікна, необхідність сприйняття інформації різної модальності тощо. Тому напрямки їхнього розвитку збігаються.

2.3 Якісні показники LLM

Галюцинації нейронної мережі – це явище, при якому штучна нейронна мережа (зазвичай, у контексті моделей ML, таких як GPT або нейронні мережі для комп'ютерного зору) генерує вихідні дані, які не відповідають реальності або очікуванням, що базуються на вхідних даних. Галюцинації можуть виявлятися різними способами, залежно від типу завдання та моделі.

1. Мовні моделі. У контексті моделей NLP, таких як GPT, галюцинації можуть виявлятися у вигляді генерації неправдивої або безглуздої інформації, яка виглядає правдоподібно, але не має підстав у фактичних даних. Наприклад, модель може «вигадувати» факти, імена чи події, яких немає.

2. Моделі комп'ютерного зору (Computer Vision, CV). Галюцинації можуть виникати, коли модель розпізнає об'єкти або елементи на зображеннях, яких там немає. Наприклад, модель може визначити наявність кішки на зображенні, де насправді зображено лише шматок меблів.

Причини галюцинацій можуть бути різними:

- нестача або якість даних – модель може навчатися на неповних, неточних або незбалансованих даних;
- архітектура моделі – деякі архітектурні особливості можуть сприяти виникненню галюцинацій, особливо якщо модель надто складна та схильна до перенавчання;
- контекст використання – у деяких випадках моделі можуть інтерпретувати контекст або запити неправильно, що призводить до створення помилкової інформації.

Галюцинації є однією з важливих проблем у галузі AI, оскільки вони можуть призводити до неправильних висновків або дій, особливо в критично важливих додатках, таких як медицина або автономні транспортні засоби. Тобто, LLM можуть видавати помилкові дані, які виглядають правдоподібно. Їх називають Повністю позбутися галюцинаціями. нейромерж не вдається досі. Лідер боротьби з галюцинаціями – закрита модель GPT-4, яка

помиляється в 3 % випадків (табл. 2.1). Однак, від неї не сильно відстає open source LLaMA 2 70B, що знаходиться за рівнем точності на рівні розрекламованої пропрієтарної Gemini від Google DeepMind.

Таблиця 2.1 – Якісні показники LLM

Модель	Частота галюцинацій, %	Коефіцієнт узгодженості фактів, %	Частота правильних відповідей, %	Середня довжина відповіді, у словах
GPT 4	3	97	100	81,1
GPT 4 Turbo	3	97	100	94,3
GPT 3.5 Turbo	3,5	96,5	99,6	84,1
Gemini Pro (Google)	4,8	95,2	98,4	89,5
LLaMa 2 70B	5,1	94,9	99,9	84,9
LLaMa 2 7B	5,6	94,4	99,6	119,9
LLaMa 2 13B	5,9	94,1	99,8	82,1
Cohere-Chat	7,5	92,5	98	74,4
Cohere	8,5	91,5	99,8	59,8
Claude 2 (Anthropic)	8,5	91,5	99,3	87,5
Phi-2 (Microsoft)	8,5	91,5	91,5	80,8
PaLM 2 (beta, Google)	8,6	91,4	99,8	86,6
Mixtral 8x7B	9,3	90,7	99,99	90,7
Titan Express (Amazon)	9,4	90,6	99,5	98,4
Mistral 7B	9,4	90,6	98,7	96,1
PaLM 2 Chat (beta)	10,0	90	100	66,2
PaLM 2 (Google)	12,1	87,9	92,4	36,2
PaLM 2 Chat (Google)	27,2	72,8	88,8	221,1

Збільшення довжини контекстного вікна. Чим більший розмір тексту, який LLM здатна обробити, тим вище її продуктивність і якість відповідей, що генеруються. Це з значним збільшенням обсягу даних, які може проаналізувати модель. Найкращі закриті моделі GPT-4 та Claude 100K можуть сприймати понад 100 тисяч токенів за раз. Нейронні мережі з відкритим кодом намагаються наздогнати конкурентів за цим показником. Наприклад, базовий Mistral 7B здатний працювати з 8000 токенів, а його новітній форк Nous-Yarn-Mistral-7B-128k від компанії Nous Research підтримує контекстне вікно 128 тисяч.

Обробка даних різних модальностей. Сучасні нейромережі вміють працювати не лише з текстом, а й із зображеннями, відео та аудіо.

Це вже реалізовано у кількох open source LLM:

- візуально-мовної моделі Nous-Hermes-2-Vision-Alpha;
- мультимодальної нейромережі Qwen-VL від китайської компанії Alibaba Cloud;
- мультимодальної версії LLaMA під назвою LLaVA-13B.

Зменшення вартості LLM. Однією з проблем нейромереж є висока вартість розробки базових моделей. Завдяки тому, що деякі з них випущені під відкритою ліцензією, витрати на їх донавчання та впровадження знижуються. Наприклад, доопрацювання та запуск open source Alpa і Vicuna-13B, заснованих на моделі LLaMA, коштували розробникам лише 600 і 300 доларів відповідно.

Один із механізмів зниження вартості – використання нейромереж як для генерації «синтетичних» навчальних даних, так і для оцінки якості роботи нової моделі. Такий підхід називають RLAI (Reinforcement Learning With AI Feedback) – навчання з підкріпленням від AI.

Можливість запуску мовних моделей на слабкому залізі. У більшості LLM з відкритим вихідним кодом менше параметрів у порівнянні зі своїми закритими конкурентами. Завдяки цьому, такі нейронні мережі запускаються на слабкому залізі, навіть домашньому комп'ютері.

Наприклад, та ж Mistral 7B має у 25 разів менше параметрів порівняно з GPT-3.5, що лежала в основі базової версії ChatGPT. Тому її використання вимагає менше обчислювальних потужностей – приблизно в 187 разів менше, ніж GPT-4, і в 9 разів менше, ніж GPT-3.5.

Поліпшення існуючих та створення нових архітектур нейромереж. Одна з головних проблем LLM, що визначає їх недоліки, – це особливості архітектури трансформера. Рішення чекають від стартапів, що працюють з open source моделями, та експериментуючими з їх внутрішньою структурою. Можливо, проблему вирішить архітектура Mixture of Experts (MoE, «модель змішаних експертів»), що копіює побудову GPT-4. Така модель складається з восьми нейромереж-експертів, кожна з яких відповідає

за свій набір завдань. Open source Mixtral 8x7B від французької компанії Mistral AI, яка використовує такий підхід, має у шість разів більшу швидкість генерації відповідей у порівнянні з вихідною LLaMA 2 70B. Побудова мультиагентних систем з урахуванням LLM. Змінювати архітектуру корисно, але є інший підхід до підвищення якості роботи мовних моделей. В його основі – побудова систем, що складаються з декількох нейромереж-агентів, які можуть домовлятися і взаємодіяти між собою для вирішення завдань користувача. Рівень кожної моделі можна оцінити за допомогою метрик якості (бенчмарків), що регулярно оновлюються. За цим показником усі LLM можна розділити на дві категорії.

1. Моделі, які демонструють результати, близькі до певної «якості відсікання». Як правило, базовим рівнем вважається ChatGPT (GPT-3.5-Turbo).

2. Моделі, які не задовольняють співвідношення ціна – якість. Це або занадто великі LLM, вартість яких зашкалює, або дуже маленькі, що містять менше 7 мільярдів параметрів. Останні зазвичай мають провали як роботи, що виявляються за допомогою окремих бенчмарків, пов'язаних з розумінням мови».

Ідеальний кандидат для таких систем – саме відкриті LLM, які не потребують великої кількості ресурсів для обчислень. І такі проекти вже є: AutoGPT, GPT-Engineer, LangChain та GPTEam.

Створення LLM для мов, відмінних від англійської мови. Нейросети воліють його взаємодії, оскільки у ньому написана більшість навчальних даних, використаних їх створення. Інші мови, якими у світі говорять десятки і сотні мільйонів людей, вважаються вторинними. Навчання для роботи з ними вимагає пошуку та складання якісних датасетів, а отже, і додаткових ресурсів. Навіть найкращі нейронні моделі на кшталт GPT-4 охоплюють лише сотню мов із понад 7000 відомих. Експерти чекають на вирішення цієї проблеми від опенсорсних LLM. Наприклад, в 2023 р. була анонсована розроблена в ОАЕ модель Jais, здатна спілкуватися арабською, і варіант

LLaMA для португальської мови. Робота продовжується і для рідкісних мов. У 2023 р. було запущено проект Massively Multilingual Speech (MMS). Його завдання – сформувати набори даних для 1100 не охоплених раніше мов.

Моделі з відкритим кодом поділяються на різні категорії за рядом параметрів: ступенем навчання, розміром та наявністю підтримки тих чи інших мов. Розберемо кожен із них. Ступінь навченості. Розробники часто викладають у відкритий доступ лише передбачені версії своїх нейронів – «претрейні». Наприклад, так вчинили фахівці з вихідною LLaMA. Такі мовні моделі перед публікацією проходять тривалий процес тренування на безлічі нерозмічених текстових даних. Це потребує великих обчислювальних потужностей та фінансових витрат. У результаті, у моделі формується тільки загальне розуміння мови.

Але використовувати «претрейн» для вирішення будь-яких завдань проблематично. Він може лише генерувати продовження текстових послідовностей, які вводять користувач. Наприклад, легко продовжить фразу з визначенням «Машинне навчання – це...». Якщо користувач спробує вести з ним діалог або відправить інструкцію для дій, то нейромережа почне видавати нісенітницю замість корисних відповідей.

Тому популярністю користуються не претрейн, а варіанти базових моделей, що пройшли додаткове тонке налаштування, яке називають «файн-тюнінг» (fine tuning). Як правило, в назвах таких LLM є слово Chat, якщо нейронку донавчили для ведення діалогу, або Instruct, якщо вона вміє виконувати інструкції за допомогою методу, аналогічного RLHF, використаного при навчанні ChatGPT.

Трапляються і більш специфічні варіанти донавчання. Наприклад, модель MPT-7B має версію StoryWriter, яка спеціалізується на написанні вигаданих історій з дуже довгим контекстом. Варто згадати і велику кількість LLM, що генерують програмний код. У назвах таких нейронів зазвичай є слово Code: StableCode, CodeGeneX і таке інше. Підтримка різних мов. За цим показником LLM ділять на три категорії: англійські; за допомогою

однієї місцевої мови, наприклад, української; мультимовні, які справляються одразу з кількома мовами, відмінними від англійської. Наприклад, при роботі з нейромережею в Україні для користувачів буде важливою є підтримка української; мови. Але базова мова для більшості моделей – англійська. Пов'язано це з тим, що саме на ньому є найбільша кількість даних, що використовуються в навчанні нейронних мереж. Інші мови вони освоюють за рахунок додаткових тренувань та внесення змін до архітектури.

Розмір моделі. Часи, коли вважалося, що чим більше нейромережа – тим краще поступово йдуть у минуле. Сучасні опенсорсні моделі при дуже скромних розмірах працюють не гірше за гігантські пропріетарні аналоги. Тому, сьогодні варто вибирати LLM за принципом золотієї середини – нейронка повинна мати найменший розмір, здатний впоратися з поставленим завданням.

Другий важливий параметр, який відноситься до розміру моделі, – тип LLM: повна або квантована. Квантування нейромережі зменшує вимоги до обчислювальних потужностей, наприклад, мінімального обсягу оперативної пам'яті. Але точність роботи самої мовної моделі у своїй знижується. «Часто зниження вартості хостингу моделі досягається шляхом квантування. В результаті її вдається запустити навіть на звичайних домашніх відеокарт типу GTX, RTX 3070-3090 від NVIDIA. Але при цьому спостерігається падіння якості її роботи на 5-15 % від вихідного варіанту (втім, у низці випадків це виявляється прийнятним)».

Не всі open source моделі однаково відкриті. Це залежить від типу ліцензії, яку вибирає розробник. Моделі, застосування яких можливе з низкою істотних обмежень, відносять до частково відкритих. Наприклад, творці LLaMA 2 пропонують користувачу перед завантаженням прийняти угоду з великим списком вимог та заборон. Один із пунктів забороняє використовувати нейромережу при кількості користувачів у проекті, що перевищує 700 мільйонів людей на місяць. І це не все. Результати роботи LLaMA 2 не можна використовувати для навчання інших LLM, крім самої

LLaMA та її похідних. В основі більшості LLM із відкритою ліцензією лежать кілька базових моделей (рис. 2.2) [33].

Основна більшість LLM поширюється під типовими ліцензіями відкритого ПЗ, серед яких можна виділити наступні базові.

1. Apache 2.0 дозволяє використовувати моделі для будь-яких цілей, модифікувати їх та розповсюджувати відповідно до умов ліцензії, без відрахування платежів розробнику. Під цією ліцензією створено переважну більшість відкритих LLM: T5, Mistral 7B та ін.

2. MIT License розроблена Массачусетським технологічним інститутом (MIT). Певною мірою збігається з Apache 2.0, але допускає повторне використання опенсорсного коду у складі пропріетарного ПЗ. Наприклад, ця ліцензія використовується для Phi-2 від Microsoft.

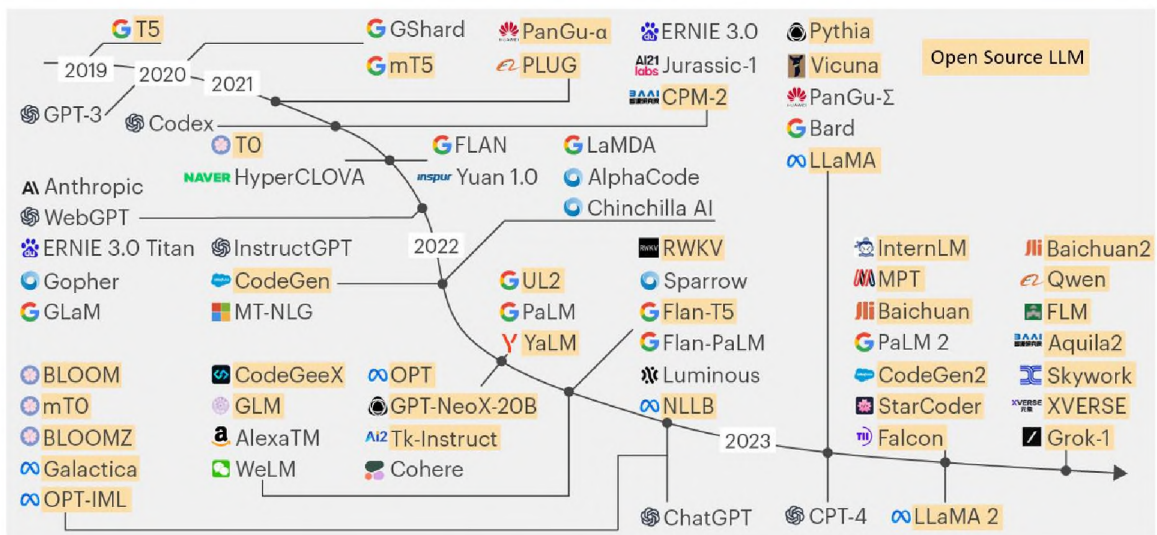


Рисунок 2.2 –Часова шкала розробки LLM з розміром понад 10 мільярдів параметрів

3. Open RAIL-M v1 підтримується спільнотою BigCode, створеною компанією Hugging Face. Ліцензія передбачає вільний доступ до моделей, можливість модифікації їх вихідного коду та спільне використання LLM та їх варіантів. Містить низку обмежень, пов'язаних із заборонаю на використання у неетичній чи протиправній діяльності. Під цією ліцензією розповсюджується модель BLOOM.

4. CC BY-SA 4.0 підтримується міжнародною некомерційною організацією Creative Commons. Дозволяє копіювати та розповсюджувати LLM, модифікувати та доповнювати їх для будь-яких цілей, включаючи комерційне використання. Але в останньому випадку поширювати нові моделі слід за тією самою ліцензією, що й оригінал. Під цією ліцензією є модель MPT-7B-Chat.

5. BSD-3-Clause. Ліцензія вільного ПЗ з мінімальними обмеженнями на використання та розповсюдження нейросіток. Дозволяє необмежене копіювання для будь-яких цілей за умови вказівки дисклеймерів про авторські права та відмови від гарантійних зобов'язань. Використовується рідко. Нам вдалося знайти одну популярну LLM з подібною ліцензією CodeT5+.

2.4 Великі мовні моделі для створення програмного коду

Окремий напрямок досліджень в галузі LLM – навчання нейронних мереж для написання програмного коду. Сьогодні для цього є кілька популярних open source моделей:

- StableCode від StabilityAI, що створила Stable Diffusion, може програмувати на Python, Java, Go, JavaScript, C і C++.

- SantaCoder – серія моделей розміром 1,1 мільярд параметрів, яка створених на базі GPT-2, навчена генерувати код мовами Python, Java та JavaScript;

- CodeGeeX та CodeGeeX2 від китайських фахівців: перша версія на 13 мільярдів параметрів була навчена 20 мовами програмування, друга – з розміром 6 мільярдів – вміє кодувати вже 100 мовами, у тому числі, Python, Java, C++, C#, JavaScript, PHP і Go (може бути підключена у вигляді плагіна до популярних IDE: Visual Studio Code, IntelliJ IDEA та Android Studio);

– Replit Code – LLM розміром 2,7 мільярда параметрів, навчена автодоповненням коду та навчалася на наборах даних, що містять 20 мов, включаючи Java, JavaScript, Python та PHP;

– StarCoder – це набір моделей з 15,5 мільярдами параметрів, навчених більш ніж 80 мовами програмування;

– CodeT5 та CodeT5+ – сімейство моделей від к. Salesforce Research. Як випливає з назви, LLM базується на базовій відкритій моделі T5 та має варіанти на 220 мільйонів, 770 мільйонів, 2 мільярди, 6 мільярдів та 16 мільярдів параметрів і здатна кодувати на Ruby, JavaScript, Python, Java, PHP, C, C++, C#;

– CodeGen2 та CodeGen2.5 – ще одна родина open source LLM з типорозмірами на 1, 3,7, 7 та 16 мільярдів параметрів від тієї ж Salesforce Research\$

– DeciCoder 1B – «скромна» модель з 1 мільярдом параметрів, яка вміє завершувати запропоновані людиною фрагменти програмного коду. Навчена мовами Python, Java та JavaScript. При цьому, як запевняють розробники, «забезпечує збільшення продуктивності в 3,5 рази, підвищену точність у тесті HumanEval і менше використання пам'яті в порівнянні з широко використовуваними LLM для генерації коду, такими як SantaCoder»;

– Code LLaMA – версія LLaMA 2, що пройшла донавчання для роботи з програмним кодом і має варіанти на 7, 13 та 34 мільярди параметрів. Справляється з Python, C++, Java, PHP, C# та TypeScript.

РОЗДІЛ 3

РЕКОМЕНДАЦІЇ ЩОДО ЛОКАЛІЗАЦІЇ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

3.1 Варіанти локалізації LLM

Згідно п. 1.4, в роботі досліджено варіанти інструментарію для реалізації локальних версій LLM.

1. Трансформери HuggingFace. Щоб запустити Hugging Face Transformers в автономному режимі без доступу до Інтернету, виконайте наведені нижче дії. Можна встановити через `pip` або `conda`, як описано у документації Hugging Face: `pip install transformers==4.xy`

Завантажте заздалегідь навчені моделі зі свого локального комп'ютера після їх завантаження з Hugging Face Hub з доступом до Інтернету. Збережіть модель за допомогою `save_pretrained()`, а потім завантажте її пізніше в автономному режимі.

```
from transformers import AutoModelForSequenceClassification,
BertTokenizerFast
```

Вперше завантажте та збережіть модель:

```
model = AutoModelForSequenceClassification.from_pretrained("bert-
base-uncased") model.save_pretrained("/my/local/directory/bert-base-
uncased")
```

Потім завантажте збережену модель в автономному режимі:

```
model = AutoModelForSequenceClassification.from_pretrained
("/my/local/directory/bert-base-uncased") tokenizer =
BertTokenizerFast.from_pretrained("/my/local/directory/bert-
base-uncased")
```

Встановіть змінні середовища `TRANSFORMERS_OFFLINE` та `HF_DATASETS_OFFLINE`, щоб увімкнути використання в автономному режимі:

```
export TRANSFORMERS_OFFLINE=1 export HF_DATASETS_OFFLINE=1
```

Клонуйте репозиторій моделі безпосередньо, якщо ви не завантажете окремі файли:

```
git clone https://huggingface.co/<username>/<repository>
/my/local/directory
```

Переконайтеся, що всі необхідні файли є у каталозі, в якому ви плануєте виконувати свої сценарії. Пам'ятайте, що встановлення TRANSFORMERS_OFFLINE тільки на 1 не буде працювати, якщо модель ще не доступна локально. Вам необхідно або завантажити модель з доступом до Інтернету та зберегти її локально, або клонувати репозиторій моделі.

2. gpt4all – це проект з відкритим вихідним кодом, який дозволяє будь-кому отримати доступ та використовувати потужні моделі AI, такі як GPT-3. Покрокові інструкції щодо встановлення та використання gpt4all мають наступний вигляд:

1. Монтаж. для gpt4all потрібно Python 3.7 або вище, а також кілька залежностей пакетів Python. Найпростіший спосіб встановити gpt4all через pip:

```
pip install gpt4all
```

2. Отримайте ключ API. Потрібний ключ API для доступу до моделей через gpt4all. Ви можете отримати безкоштовний ключ API, зареєструвавшись на сайті [L3].

3. Налаштуйте автентифікацію. Отримавши ключ API, встановіть змінне середовище з ім'ям OPENAI_API_KEY, вказавши як значення свій ключ API:

```
export OPENAI_API_KEY="YOUR_API_KEY_HERE"
```

4. Використовуйте gpt4all. Тепер можна використовувати gpt4all в Python для доступу до GPT-3 та інших моделей. Ось приклад, що спонукає Davinci (версія GPT-3) резюмувати уривок тексту:

```
from gpt4all import completions
summary = completions.generate(engine="text-davinci-003",
prompt="Summarize this text: [insert long text here]",
max_tokens=50)
print(summary.text)
```

3. Ollama [34] – це бібліотека з відкритим вихідним кодом, яка забезпечує легкий доступ до LLM. Системні вимоги: Python 3.7 або вище; бібліотека запитів; дійсний ключ API OpenAI. Для встановлення є команда:

```
pip install ollama
```

Для використання можна використовувати наступний код:

1. Встановити ключ API:

```
import os os.environ["OPENAI_API_KEY"] = "<YOUR_API_KEY>"
```

2. Імпортуйте та створіть клієнт:

```
from ollama import Client client = Client()
```

3. Створити текст:

```
response = client.generate( engine="text-davinci-003",
prompt="Hello world", max_tokens=100 )
print(response.generations[0].text)
```

Метод генерації дозволяє вказати механізм, запрошення та параметри, такі як максимальна кількість токенів, для налаштування вашого запиту. Також можна налаштовувати двигун, модерувати контент та багато іншого. Додаткову інформацію про всі доступні методи див. у документації Ollama. Підсумовуючи: Ollama дозволяє дуже легко використовувати GPT-3 та ін. моделі всього за кілька рядків коду Python після встановлення та налаштування.

4. LocalLLM – у поєднанні з хмарними робочими станціями робить революцію в розробці програм на основі AI, дозволяючи використовувати LLM локально на процесорі та в пам'яті в середовищі Google Cloud. Усунувши необхідність у графічних процесорах, ви зможете подолати проблеми, пов'язані з нестачею графічних процесорів, та розкрити весь потенціал LLM. Завдяки підвищеній продуктивності, економічній ефективності та покращеній безпеці даних LocalLLM дозволяє з легкістю створювати інноваційні програми.

5. Llama.cpp. Щоб встановити та використовувати Llama.cpp для локального навчання та виводу, виконайте такі дії:

Встановіть залежність:

```
Python 3 CMake Optional (for GPU acceleration) NVIDIA
drivers, CUDA, and cuDNN For Windows, use Visual Studio
Community with Desktop C++ Environment and Python 3 Clone the
repository: git clone --recursive
https://github.com/abetlen/llama-cpp-python.git
```

Якщо використовувати прискорення графічного процесора, встановіть змінне середовище, як описано в репозиторії (наприклад, Linux):

```
export CMAKE_ARGS="-DLLAMA_CUBLAS=ON" Install Llama.cpp:
```

Для локального складання та компіляції системи:

```
cd llama-cpp-python pip install -e.
```

За бажанням використовувати інтерфейс високого рівня, використовуйте команду:

```
python -m llama_cpp.server --model models/7B/llama-
model.gguf 3 GPU support: set FORCE_CMAKE=1 set CMAKE_ARGS=-
DLLAMA_CUBLAS=ON pip install --upgrade --force-reinstall llama-
cpp-python -- no-cache-dir
```

Необов'язково для Windows використовувати інтерфейс високого рівня:

```
python -m name_cpp.server --модель "path/to/your/model"
Download and place Llama models in models/ subdirectory within
Llama.cpp repository if they no included (models like llama-
model.gguf or ggml- * Files є доступні на Hugging Face або
GGML).
```

5. LM Studio – це інструмент із відкритим вихідним кодом, який оптимізує процес навчання, оцінки та використання сучасних мовних моделей на місцевому рівні [35]. Ось кроки для запуску LM Studio:

1. Монтаж. LM Studio вимагає Docker та Docker Compose. Встановити:

а) Встановіть Docker Engine та Docker Compose на свій комп'ютер.

б) Клонуйте репозиторій LM Studio GitHub:

```
git clone https://github.com/lm-studio/lm-studio.git
```

2. Конфігурація:

LM Studio використовує конфігураційні файли в папці config/.

Загальні файли включають:

а. model.yaml: вкажіть архітектуру моделі.

б. training.yaml: встановити параметри навчання

в. evaluation.yaml: налаштувати параметри оцінки.

3. Запуск LM Studio. Основний інтерфейс здійснюється через Docker Compose. Деякі поширені команди:

- а) Образи складання: `docker-compose build`
- б) Запустіть усі служби: `docker-compose up -d`
- в) Перегляд журналів: `docker-compose logs -f`
- д) Зупинити служби: `docker-compose down`

Служби надають різні порти, які ви можете використовувати для взаємодії з інтерфейсом користувача, API, блокнотами і багатьом іншим. Підсумовуючи, можна сказати, що LM Studio полегшує експериментування з моделями місцевої мови. Користувачам просто потрібно клонувати репозиторій, налаштувати файли налаштувань та використовувати прості команди Docker, щоб розпочати навчання, оцінку та використання моделей.

На основі проведених досліджень зроблений висновок про доцільність використання у роботі LM Studio в якості основного інструментарію для локалізації LLM.

3.2 Аналіз властивостей LM Studio

LM Studio від Hugging Face [35] – це інтегроване середовище розробки (IDE) для роботи з мовними моделями. Вона призначена для спрощення процесу створення, налаштування, навчання та впровадження мовних моделей, таких як GPT та інші трансформерні моделі. Основні можливості LM Studio включають:

1. Розробка та налаштування моделей: Надає інструменти для налаштування архітектури моделей, гіперпараметрів та інших аспектів навчання моделей.

2. Навчання та оцінка: Включає інструменти для ефективного навчання моделей на великих обсягах даних, а також оцінки продуктивності моделей.

3. Інтеграція з екосистемою Hugging Face: Дозволяє легко інтегрувати моделі з іншими інструментами та сервісами Hugging Face, такими як бібліотека Transformers та платформа для розгортання моделей.

4. Керування експериментами: Дозволяє відстежувати різні експерименти, порівнювати їх результати та керувати версіями моделей.

LM Studio призначений для дослідників, інженерів та розробників, що працюють у галузі NLP, та допомагає спростити робочі процеси, пов'язані зі створенням та використанням потужних мовних моделей.

LM Studio – це зручна програмна програма, яка спрощує використання LLM з відкритим вихідним кодом для всіх, незалежно від їх навичок програмування. Вона надає інтуїтивно зрозумілий інтерфейс для вивчення, завантаження та запуску різних моделей AI прямо на комп'ютері, що робить AI більш доступним та керованим. При цьому, навички програмування не потрібні. LM Studio розроблений з урахуванням простоти та дозволяє користувачам взаємодіяти з моделями AI через зручний інтерфейс без необхідності писати жодного рядка коду. LM Studio доступна для Mac, Windows та Linux. Конкретні системні вимоги залежать від моделей AI, які збираєтеся використовувати, оскільки для більших моделей потрібне потужніше обладнання. ПЗ включає функції, які дозволяють оцінити, чи підходить модель до комп'ютера на основі наявних специфікацій. Встановити LM Studio дуже просто.

1. Відвідайте вебсайт LM Studio.
2. Виберіть версію, сумісну з операційною системою (Mac, Windows, Linux).
3. Завантажте та відкрийте інсталятор.
4. Виконайте вказівки на екрані, щоб завершити процес встановлення.

Так, однією з відмінних рис останнього оновлення LM Studio є можливість одночасного запуску кількох моделей AI. Ця функція, доступна в режимі «Ігровий майданчик», дозволяє користувачам використовувати комбіновані можливості різних моделей для підвищення продуктивності та продуктивності. LM Studio інтегрується з Hugging Face, надаючи доступ до широкої бібліотеки моделей штучного інтелекту, включаючи мовні моделі, такі як GPT-3, BERT та ін. Незалежно від того, чи цікавитеся ви NLP, ML або

іншими областями AI, знайдете моделі, що відповідають вашим потребам. Останнє оновлення LM Studio безкоштовне для окремих користувачів. Однак найсвіжішу інформацію про ціни або тарифні плани, якщо вони є, можна знайти на вебсайті LM Studio. LM Studio включає функцію, яка дозволяє оцінити, чи підходить квантована версія моделі на комп'ютері на основі специфікацій. Вона містить рекомендації щодо сумісності та продуктивності, допомагаючи приймати обґрунтовані рішення про те, які моделі слід завантажити та запустити. Інформацію про комерційне використання можна переглянути в умовах обслуговування та ліцензійних угод на вебсайті LM Studio. Важливо ознайомитися з цими документами, щоб зрозуміти правові основи використання LM Studio у комерційних цілях. Вебсайт LM Studio містить вичерпну документацію, навчальні посібники та відповіді на найпоширеніші запитання, які допоможуть користувачам розпочати роботу та усунути поширені проблеми. Якщо вам потрібна більш персоналізована допомога, ви можете звернутися за допомогою до служби підтримки або на форуми спільноти.

3.3 Приклад локалізації LLM на основі LM Studio

Сімейство LLaMA [36]. Вихідна модель LLaMA представлена у лютому 2023 р. Вона має версії з типорозмірами 7, 13, 33 та 65 мільярдів параметрів. Перші з них можна було запустити всього на одному графічному процесорі, що стало міні-сенсацією в момент запуску. У липні 2023 р. вийшла покращена версія LLaMA 2, розроблена у співпраці з Microsoft. Ця LLM має варіанти на 7, 13 та 70 мільярдів параметрів. Незабаром на базі LLaMA з'явилася повністю відкрита версія OpenLLaMA. Вона стала основою для множини проєктів, що розвивають модель за рахунок експериментів з архітектурою та варіантів тонкого налаштування та навчання. LLaMA 2 70B – це умовно відкрита модель. Є вихідний код і ваги, але її не можна

застосовувати з комерційною метою, якщо користувачів буде більше 700 мільйонів чоловік на місяць. Це, мабуть, найвідоміша модель після пропріетарних ChatGPT та Claude 2. У 2023 р. основний прорив у масовому використанні нейронних мереж з відкритим кодом внесла LLaMA, на базі якої з'явилися десятки моделей: Mistral, Zephyr, Alpaca, Phi-2, Qwen, Yi та ін.

Llama 2 та Llama-2-Chat – це три LLM Llama 2 з 7, 13 та 70 мільярдами параметрів, та донавчені для ведення діалогів моделі Llama-2-Chat 7B, 34B та 70B. Моделі доступні open source для дослідницьких та комерційних цілей, крім моделі Llama 2 34B – вона була навчена, але не опублікована дослідниками. Відповіді моделей оцінювалися за двома показниками: корисність та безпека відповідей, і результати перевершили всі state-of-the-art моделі open source. У порівнянні з ChatGPT результати Llama-2-Chat 70B знаходяться в межах довірчого інтервалу відхилення. Спочатку дослідники навчили модель Llama 2 без вчителя на корпусах текстів із загальнодоступних онлайн-ресурсів. Для навчання використовувалося на 40 % більше даних, ніж першого покоління Llama, а контекст збільшився до 4000 токенів. Потім донавчалася з учителем для тонкого налаштування передбачення (рис. 3.1).

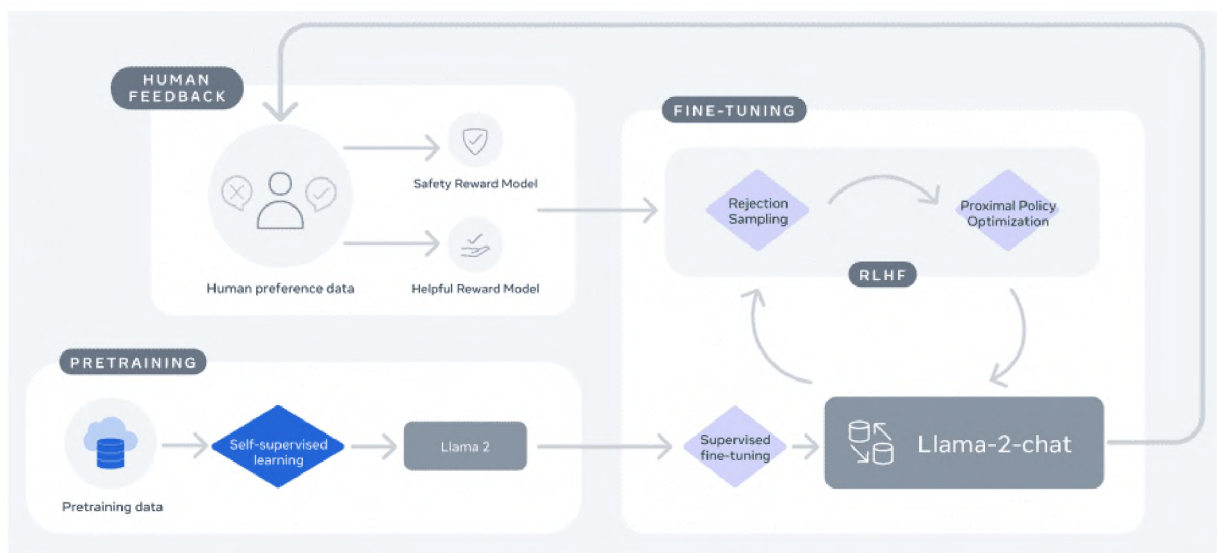


Рисунок 3.1 – Процес навчання моделей LLaMA-2 та LLaMA-2-Chat

Наступне ітеративне налаштування моделі проходило за методом навчання з підкріпленням на основі зворотного зв'язку від користувачів

(RLHF) з використанням методів «Вибірка з відхиленням» та «Proximal Policy Optimization», що додає шуму для уникнення локального мінімуму. Ключова ідея навчання – ітеративне накопичення даних моделі винагороди паралельно з покращенням моделі, щоб гарантувати результати у межах розподілу. Модель винагороди приймає відповідь моделі та відповідну підказку (включаючи контексти з попередніх звернень) як вхідні дані та видає скалярний показник якості (корисність та безпеку) відповідей моделі. Використовуючи такі оцінки відповідей як винагороди, відповіді Llama-2-Chat оптимізувалися для кращої відповідності переваг людей.

Результати оцінки людьми відповідей моделі Llama-2-Chat порівняно з відкритими та закритими моделями приблизно на 4000 підказок про корисність з 3-ма експертами-оцінювачами на кожну підказку – рис. 3.2.

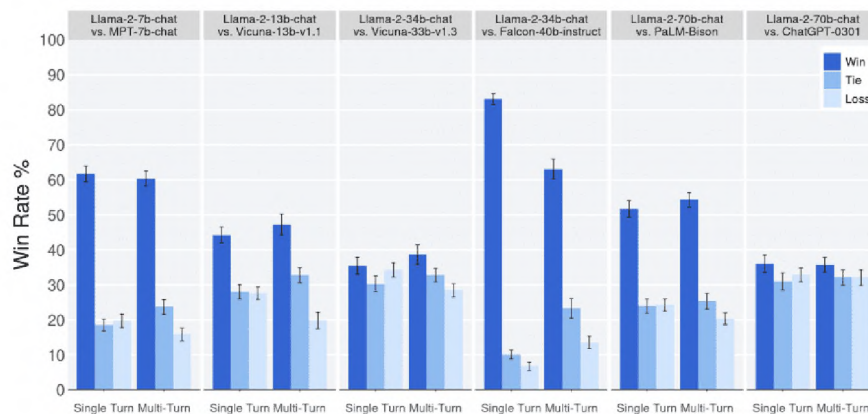


Рисунок 3.2 – Результати оцінки людьми відповідей Llama-2-Chat

В цілому, варті уваги кілька LLM із сімейства LLaMA для локалізації. Vicuna-13B від LMSYS Org – це одна моделей, що показує непогані результати в актуальних бенчмарках.

Mistral – модель від однойменного французького стартапу, що перевершує LLaMA 2 13B у всіх бенчмарках. На кінець вересня 2023 р. була найкращою LLM з розміром 7 млрд параметрів. Mistral 7B цікава тим, що, маючи лише 7 мільярдів параметрів, вона показує кращі результати, ніж версія LLaMA 2 із 13 мільярдами. Це дозволяє використовувати модель на будь-якому сучасному ноутбучі.

Zephyr-7B – це версія Mistral, що пройшла процедуру тонкого налаштування (файн-тюнінгу) за допомогою методу Direct Preference Optimization (DPO). Має 90,6 % частоту перемог над іншими LLM в AlpacaEval Leaderboard.

OpenChat – бібліотека мовних моделей з відкритим кодом. За оцінками, вона досягає якості ChatGPT (у версії від березня 2023 р.), а також перевершує чат-бот Grok. OpenChat 7B зроблений на базі Mistral 7B, але на відміну від нього проходить відомий тест на банан, який формулюється у вигляді питання до LLM: «Я на кухні, поклав тарілку на банан. Потім я відніс тарілку до спальні. Де зараз банан?»

Xwin-LM-70B-V0.1 – модель, створена на базі LLaMA 2. Як стверджують розробники, це перша модель, яка перевершила GPT-4 у бенчмарку AlpacaEval. Щоправда, розмір має досить великий – 70 мільярдів параметрів».

Окремої уваги заслуговують візуальні асистенти до LLM. Прикладом подібних моделей Large Language-and Vision Assistant (LLaVA) llava-llama-3-8b-v1_1 [38] – це модель LLaVA, налаштована з meta-llama/Meta-Llama-3-8B-Instruct і CLIP-ViT-Large-patch14-336 із ShareGPT4V-PT і InternVL-SFT від XTuner [39]. Як відомо, налаштування LLM з використанням машинних даних, що йдуть за інструкціями, покращили можливості техніки zero-shot в нових завданнях NLP, але ця ідея менш вивчена в галузі мультимодальних операцій. В LLaVA використані Мультимодальні дані інструктажу. В даній реалізації представлено спробу використовувати лише мову GPT-4 для генерації мультимодальних даних, що йдуть за інструкціями мови-образу. Модель LLaVA – комплексно навчена велика мультимодальна модель, яка з'єднує кодувальник машинного зору і LLM для загального візуального і мовного розуміння. LLaVA демонструє оптимістичні можливості мультимодального чату, іноді демонструючи поведінку мультимодального GPT-4 на невидимих зображеннях/інструкціях, і дає відносну оцінку 85,1 % порівняно з GPT-4 на синтетичному мультимодальному наборі даних. При

тонкому налаштуванні на Science QA синергія LLaVA та GPT-4 досягає точності 92,53 %. Автори опублікували згенеровані GPT-4 дані налаштування візуальних інструкцій, модель та саму кодову базу. LLaVa з'єднує попередньо навчальний візуальний кодировщик CLIP ViT-L/14 і LLM Vicuna, використовуючи просту проекційну матрицю. Візуальний чат LLaVA точно побудований на згенерованих мультимодальних даних, наступних інструкціях, для щоденних програм, орієнтованих на користувача. LLaVA працює на основі цього набору даних мультимодальних досліджень для наукової області. Приклад, використання візуального асистента наведений на рис. 3.3. Для підтвердження можливості практичної реалізації запропонованого підходу в роботі проведена локалізація llama-llava-3-8b-v1_1 в LM Studio версії 0.2.23 – рис. 3.4 (як варіант, код використання LLaVA з llama та llama.cpp наведений в додатку А).

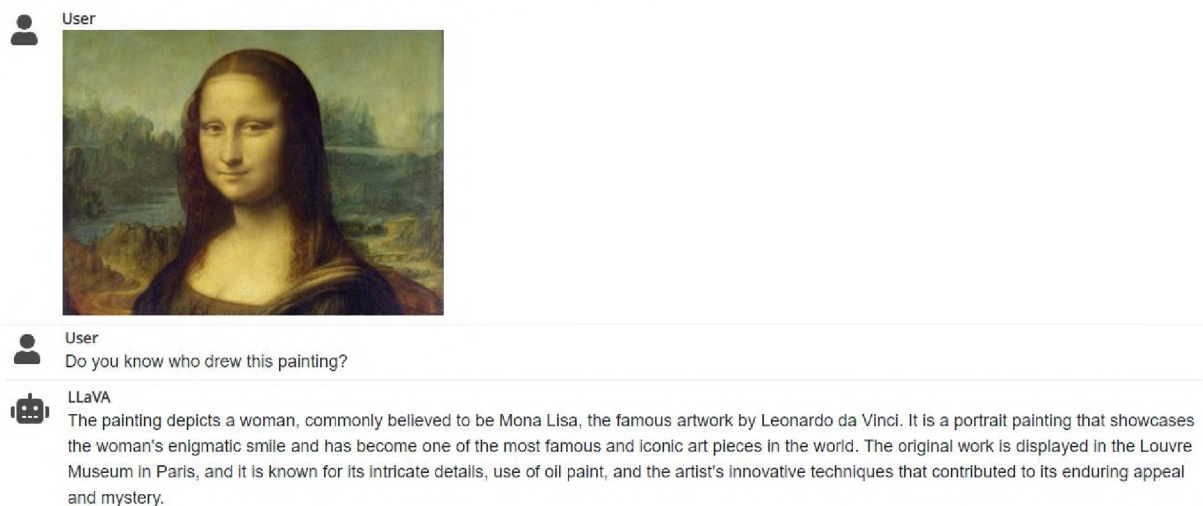


Рисунок 3.3 – Приклад використання LLaVA

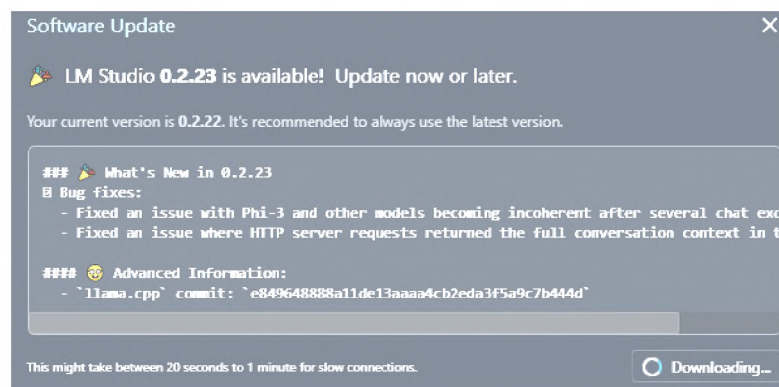


Рисунок 3.4 – Інсталяція ПЗ LM Studio 0.2.23

Після завантаження LLM в меню можна вибрати необхідну версію (рис. 3.5). В роботі використовуємо повну (f16) та квантовану (int4) версії моделі llava-llama-3-8 від Xtuner.

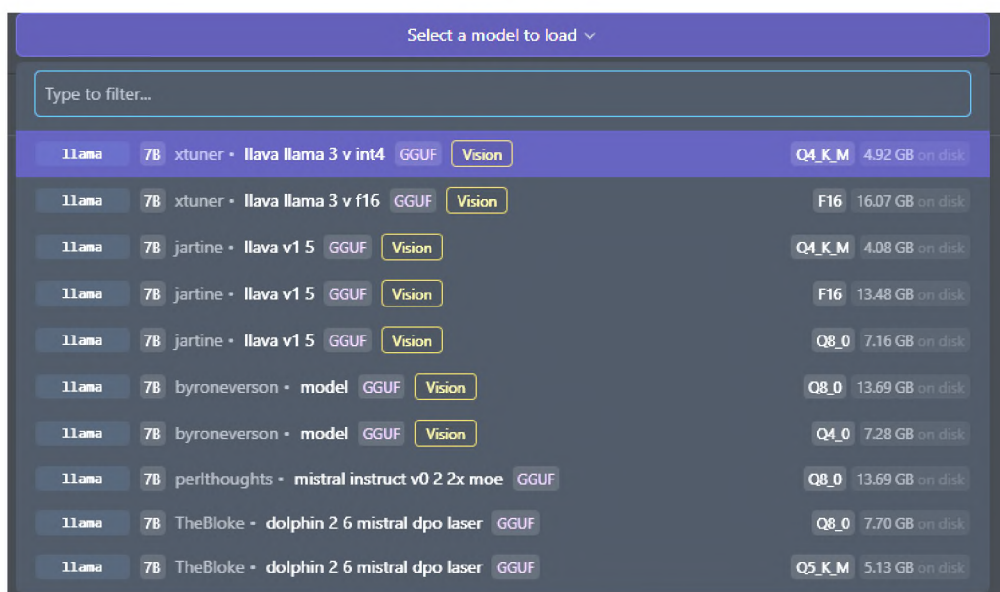


Рисунок 3.5 – Вибір LLM

Надалі проводимо тест «на банан» (див. п.3.2) – рис. 3.6. Для порівняння такий же тест виконаний для ChatGPT4o (рис. 3.7). Надалі був виконаний запит у чаті LLaVA: «Please describe this image» і введене зображення – рис. 3.8.

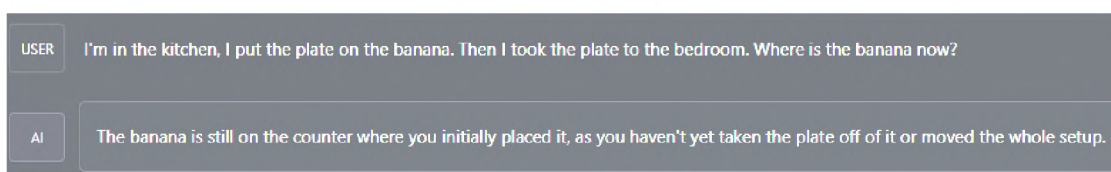


Рисунок 3.6 – Результат тестування LLaVA

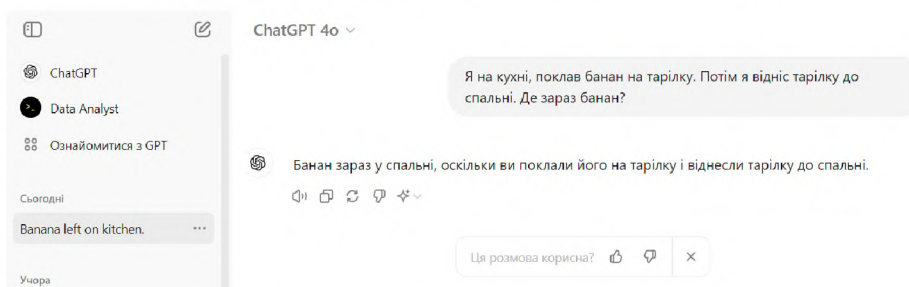


Рисунок 3.7 – Результат тестування ChatGPT4o

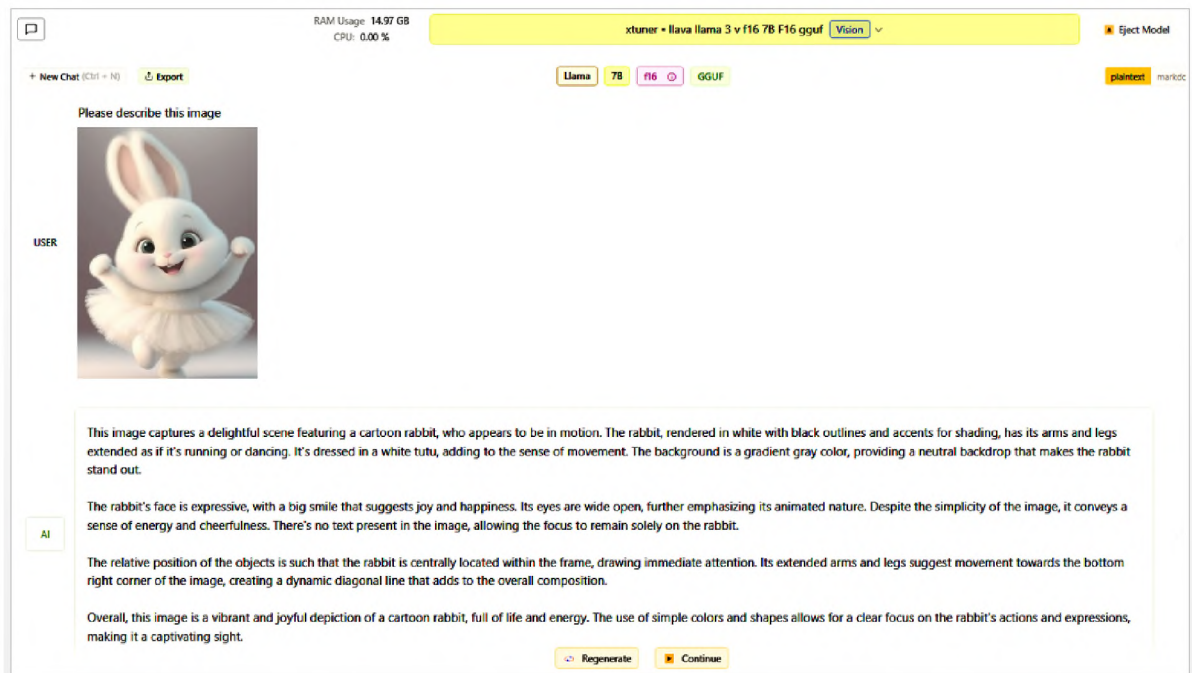


Рисунок 3.8 – Робота LLaVA з зображенням

Модель LLaVA (f16) дала наступну відповідь (переклад укр.): «На цьому зображенні чудова сцена з мультяшним кроликом, який, здається, рухається. Кролик, зображений у білому кольорі з чорними контурами та акцентами для затінення, має витягнуті руки й ноги, наче він біжить чи танцює. Незважаючи на простоту образу, він передає відчуття енергії та життєрадісності. На зображенні немає тексту, тому фокус залишається лише на кролику. Взаємне розташування об'єктів таке, що кролик розташований у центрі кадру, одразу привертаючи увагу. Він одягнений у білу пачку, що додає відчуття руху. Фон має градієнтний сірий колір, що забезпечує нейтральний фон, який виділяє кролика. Мордочка кролика виразна, з широкою усмішкою, яка навіює радість і щастя. Його очі широко відкриті, що ще більше підкреслює його жвавий характер. Його витягнуті руки та ноги вказують на рух до нижнього правого кута зображення, створюючи динамічну діагональну лінію, яка доповнює загальну композицію. Загалом це зображення є яскравим і радісним зображенням мультяшного кролика, сповненого життя та енергії. Використання простих кольорів і форм дозволяє чітко зосередити увагу на діях і виразах кролика, що робить його

захоплюючим». Статистичні параметри відповіді: time to first token: 14.34s; gen t: 151.13s; speed: 1.29 tok/s.

Аналогічним чином тестувалась спрощена (квантована) модель – LLaVA (int4). Вона дала наступну відповідь (переклад укр.): «На цьому зображенні зображено чарівну сцену з мультяшним кроликом, повним життя та енергії. Кролик, зображений білим кольором з чорними контурами та акцентами для штрихування, є центральною фігурою на зображенні. Здається, він танцює або біжить, на що вказують його витягнуті руки та ноги. Кролик прикрашений рожевою пачкою, що додає нотки кольору його однотонному вигляду. Ця деталь говорить про те, що кролик може займатися грою або святкуванням. Фон зображення градієнтного сірого кольору, що забезпечує нейтральний фон, який дозволяє яскравому білому кролику виділятися. Відсутність будь-яких інших об'єктів на зображенні зосереджує увагу виключно на кролику та його діях. На зображенні немає тексту, що не відволікає увагу глядача та дозволяє йому повністю оцінити анімаційні рухи кролика. Відносне розташування кролика є центральним у кадрі, ще більше підкреслюючи його важливість у зображенні. Загалом, це чудове зображення кролика з мультфільму, сповненого енергії та радості, зафіксованого в розпал грайливого чи святкового моменту». Статистичні параметри відповіді: time to first token: 10.13s; gen t: 65.85s; speed: 3.38 tok/s.

Отримані результати підтверджують можливість локалізації візуальних і великих мовних моделей на основі платформи LM Studio.

3.4 Оцінка переваг та недоліків відкритих LLM

Компанії обирають open source LLM через їх переваги над пропріетарними моделями за кількома показниками.

Безпека та конфіденційність даних. LLM open source можна розгорнути на власній інфраструктурі без надсилання інформації на сторонні сервери.

Завдяки цьому користувачі одержують повний контроль над даними, які обробляє нейронна мережа.

Зниження залежності від постачальників послуг ІТ. Користувачі можуть вибрати найбільш підходящий для себе варіант нейронної мережі з сотень open source LLM. Таким чином, компанія не прив'язується до одного постачальника AI-рішень і може вибирати найкращі моделі або навіть поєднувати їх між собою.

Економія коштів – open source LLM можна використовувати без оплати передплати або регулярних виплат розробникам за контрактами. Тому вони популярні у стартапів та компаній з обмеженим бюджетом.

Прозорість використовуваних LLM. Моделі з відкритим кодом можна вивчити зсередини і зрозуміти, як саме вони працюють з даними. Це дозволяє виявити та запобігти відправленню інформації на сторонні сервери.

Проекти з відкритим вихідним кодом підтримуються групами розробників та експертів. Завдяки цьому, баги і проблеми швидко усуваються, а документація докладно описує нюанси використання нейронної мережі. Це притаманно більшості open source моделей, але є й неприємні винятки. Нестандартні рішення та підходи. Відкриті LLM дозволяють експериментувати з AI, спираючись на нові базові моделі. Навіть невеликі стартапи можуть творчо переробляти такі нейромережі та використовувати їх як основу для власних унікальних розробок.

Однак у open source LLM є недоліки. Їх використання та обслуговування можуть вимагати більше часу та технічних знань від фахівців, ніж при використанні пропріетарних моделей. Останні зазвичай готові до роботи «з коробки». Розробки від маловідомих колективів можуть бути навчені на неповні або неякісні дані. Це знижує точність відповідей нейромережі та підвищує частоту галюцинацій. У open source моделей можливі недокументовані проблеми у роботі. Наприклад, відсутність сумісності між різними версіями LLM. Щоб розібратися в тому, яка open source LLM краща, фахівці створили віртуальні тестові арени, які

називають лідербордами. В них мовні моделі борються між собою. На таких сайтах кожна нейромережа оцінюється по ряду метрик якості (бенчмарків). При цьому варто розуміти, що ідеальної у всіх сенсах LLM немає. Модель може демонструвати видатні результати за одним показником, але бути аутсайдером за іншими бенчмарками.

Тому при виборі варто орієнтуватися на метрики, що найбільше відповідають задачі, для вирішення якої ми вибираємо LLM. Більшість тестових арен забезпечені зручним інтерфейсом, що дозволяє сортувати списки доступних моделей відповідно до параметрів, що цікавлять. На даний час, найбільш поширено застосовуваними є кілька лідербордів.

Open LLM Leaderboard. Платформа компанії Hugging Face, призначена для відстеження, ранжирування та автоматичної оцінки найновіших LLM та чат-ботів, представлених на однойменному сайті. Використовує оригінальну систему оцінки мовних моделей EleutherAI, що базується на розрахунку 7-ми бенчмарків.

Chatbot Arena Leaderboard. Ще одна відкрита платформа оцінки LLM на сайті Hugging Face. Основна ідея Chatbot Arena Leaderboard – це попарне порівняння якості відповідей моделей людьми-асесорами за допомогою рейтингу Elo. Справа в тому, що існують «шахрайські» способи навчити моделі показувати високі результати у бенчмарках, які при цьому не корелюють із їх реальними показниками якості. У такому разі ручна людська оцінка простим порівнянням частково вирішує цю проблему. Працює вона за краудсорсинговою схемою. Тут зібрано понад 200 тисяч відгуків реальних користувачів, що дозволяють оцінити мовні моделі за допомогою системи ранжирування Elo, подібної до рейтингу, що застосовується для розрахунку рівня гри шахістів. **AlpacaEval Leaderboard.** Автоматична система оцінки мовних моделей, що належать до класу Instruct. Заснована на методиці AlpacaFarm, яка перевіряє здатність LLM дотримуватись загальних інструкцій користувача. Як «суддя» та джерело еталонних відповідей у ній використовується AI на основі моделі GPT-4.

Chatbot Arena. Розробка LMSYS Org (Large Model Systems Organization) із Каліфорнійського університету в Берклі, що створила модель Vicuna-13B. Важливо, що лідерборд не оновлювався із травня 2023 р. Big Code Models Leaderboard.

Система оцінки LLM, призначених для створення програмного коду. Ще одна технологія Hugging Face. Лідерборд не оновлювався з листопада 2023 р., тож може містити неактуальні дані. Легко помітити, що лідером у порівнянні платформа Hugging Face. На ній є десяток бенчмарків під назвою The Big Benchmarks Collection. Тут легко налаштувати рейтинг для вибору найкращої моделі під конкретне завдання, наприклад, написання коду. Але у цьому напрямі працюють не лише конкретні компанії. Окремі open-source-спільноти намагаються створити єдину систему оцінки, здатну поєднати переваги всіх існуючих лідербордів. Так було створено LLM-Leaderboard, проєкт Людвіга Штумппа з Німеччини.

У більшості завдань відкриті моделі незначно поступаються пропрієтарним за сухими метриками. Так, наприклад, у задачі відповідей на запитання чи спрощення текстів користувач далеко не відразу помітить різницю між LLaMA 2 70B та ChatGPT. Крім того, розрив у рейтингах між закритими та відкритими моделями постійно скорочується.

3.5 Техніко-економічне обґрунтування прийнятих рішень

У сучасному світі LLM відіграють важливу роль у різних сферах діяльності, включаючи науку, бізнес та інформаційні технології. Локалізація LLM є ключовим аспектом для забезпечення ефективності та надійності їх застосування в конкретному мовному та культурному контексті.

Поточний стан IT-систем у багатьох організаціях характеризується недостатньою адаптацією до місцевих мовних особливостей. Це призводить до зниження ефективності комунікації та взаємодії з користувачами.

Локалізація LLM дозволить підвищити якість обслуговування, знизити витрати на підтримку та покращити взаємодію з клієнтами.

Ринок локалізованих LLM стрімко зростає. Попит на адаптовані до місцевих умов LLM збільшується в різних галузях, включаючи освіту, медицину, фінанси та маркетинг. Локалізація LLM надає організаціям конкурентну перевагу, дозволяючи їм швидше та точніше відповідати на запити користувачів, що в свою чергу підвищить задоволеність клієнтів і збільшить ринкову частку.

Інтеграція локалізованих LLM в існуючі IT-системи вимагає ретельної технічної оцінки. Основні технічні вимоги включають потужне апаратне забезпечення для обробки великих обсягів даних, ПЗ для налаштування та оптимізації моделей, а також людські ресурси з відповідними навичками. Важливо також забезпечити безпеку та конфіденційність даних, що обробляються моделями.

Фінансовий аналіз включає оцінку прямих та непрямих витрат на впровадження локалізованих LLM. Прямі витрати включають вартість ліцензій на програмне забезпечення, придбання апаратного забезпечення та оплату послуг спеціалістів. Непрямі витрати включають навчання персоналу та технічну підтримку. Очікувані фінансові вигоди включають зниження витрат на обслуговування, підвищення продуктивності та зростання доходів за рахунок покращення якості обслуговування клієнтів. Наприклад, при використанні LM Studio та open source LLM рівня GPT4, в тому числі з візуальним асистентом, не має потреби сплачувати абонплату ChatGPT4о к. OpenAI (≈ 800 грн/місяць), що складає 9600 грн на рік.

Ідентифікація та управління ризиками є важливим етапом процесу впровадження локалізованих LLM. Основні ризики включають технічні ризики, пов'язані з інтеграцією моделей, фінансові ризики, пов'язані з перевищенням бюджету, та ризики, пов'язані з людським фактором, такі як недостатня кваліфікація персоналу. Стратегії управління ризиками

включають регулярний моніторинг проекту, створення резервних планів та проведення навчальних програм для співробітників.

План впровадження включає наступні етапи:

1. Підготовчий етап: аналіз вимог, планування ресурсів, підготовка технічної інфраструктури.
2. Впровадження: інтеграція локалізованих LLM, тестування та налаштування моделей.
3. Навчання: навчання персоналу, проведення семінарів та тренінгів.
4. Експлуатація: моніторинг роботи моделей, технічна підтримка, регулярне оновлення та оптимізація.

Запропонований в роботі підхід до локалізації LLM дозволяє забезпечити: конфіденційність даних; низьку затримку та високу доступність; налаштування та адаптація під конкретні завдання; економічну ефективність; незалежність від зовнішніх послуг; відповідність законодавству та нормативним вимогам; підтримку локальних мов та особливостей. Такий підхід є привабливим для організацій, які прагнуть досягти максимальної ефективності та надійності своїх ІТ-рішень. Локалізація LLM є доцільним та економічно обґрунтованим рішенням для організацій, які прагнуть підвищити ефективність та надійність своїх ІТ-рішень. Впровадження локалізованих LLM дозволить покращити взаємодію з користувачами, знизити витрати на обслуговування та підвищити продуктивність.

ВИСНОВКИ

Нейромережні LLM, такі як генеративні та дискримінантні, використовуються для прогнозування наступного слова в тексті, що призводить до багатофункціональності. Генеративні моделі створюють нові екземпляри даних, тоді як дискримінантні вирішують завдання класифікації. Розвиток глибокого навчання значно спростив процес розробки моделей NLP, забезпечуючи їхню здатність до складних завдань. Прикладом є четверта генерація GPT, що забезпечує мультимодальну обробку.

Розвиток LLM включає покращення існуючих та створення нових архітектур, таких як архітектура Mixture of Experts (MoE), що копіює побудову GPT-4. Такі підходи дозволяють збільшити швидкість генерації відповідей та підвищити якість роботи моделей.

Практичні кейси використання LLM демонструють їх застосування у різних галузях: медицині, освіті, технічній підтримці, маркетингу та ін. Нейромережні LLM відкривають широкі можливості для розвитку AI та покращення якості життя людей, забезпечуючи високий рівень безпеки, продуктивності та ефективності в різних сферах діяльності.

При цьому, локалізація візуальних і великих мовних моделей дозволяє забезпечити конфіденційність даних, мінімізацію затримок, налаштування під конкретні завдання, економічну ефективність і незалежність від зовнішніх постачальників послуг, а також відповідність законодавчим вимогам і підтримку локальних мов (діалектів).

Зазвичай, для локалізації використовують open source LLM. Виконана оцінка переваг та недоліків open source LLM. Їх якість LLM залежить від різних параметрів (довжина контекстного вікна, здатність обробляти різні модальності даних, вартість розробки та ін.). Використання LLM на слабкому обладнанні стало можливим завдяки зменшенню кількості параметрів і квантуванню моделей. Такі моделі можуть працювати на звичайних домашніх комп'ютерах, що робить їх доступними для широкого кола користувачів.

В роботі досліджено кілька варіантів локалізації LLM, наприклад, Hugging Face Transformers ; gpt4all; Ollama; Locallm; Llama.cpp; LM Studio. В якості базового розглядається платформа LM Studio. Вона є інтегрованим середовищем розробки для роботи з LLM, яке спрощує процес створення, налаштування, навчання та впровадження моделей.

Для підтвердження можливості практичної реалізації запропонованого підходу в роботі проведена локалізація llama-3-8b-v1_1 в LM Studio версії 0.2.23. Отримані результати підтверджують можливість реалізації запропоновано підходу щодо локалізації LLM.

В рамках техніко-економічного обґрунтування розглянуті питання щодо оцінки поточного стану IT-системи; аналізу трендів ринок локалізованих LLM; інтеграції локалізованих LLM в існуючі IT-системи; проведення фінансового аналізу впровадження локалізованих LLM; ідентифікації та управління ризиками в рамках такого підходу. На їх основі запропонований план впровадження локалізованих LLM в IT-систему підприємства.

При цьому, очікувані фінансові вигоди включають зниження витрат на обслуговування, підвищення продуктивності та зростання доходів за рахунок покращення якості обслуговування клієнтів. Наприклад, при використанні LM Studio та open source LLM рівня GPT4, в тому числі з візуальним асистентом, не має потреби сплачувати абонплату ChatGPT4o к. OpenAI (≈ 800 грн/місяць), що складає 9600 грн на рік.

Таким чином, результатами роботи є рекомендації щодо локалізації візуальних і великих мовних моделей. Вони можуть бути використані для подальших досліджень за даною тематикою та при проектуванні локальних сервісів NLP.