

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

Аліна САВЧЕНКО

“__”_____2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
(ДИПЛОМНА РОБОТА, ПОЯСНЮВАЛЬНА ЗАПИСКА)
ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ

“МАГІСТРА”

ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ
“ІНФОРМАЦІЙНІ УПРАВЛЯЮЧІ СИСТЕМИ ТА ТЕХНОЛОГІЇ”

Тема: “Мовні моделі для аналізу великих об'ємів тексту”

Виконав: Драган Валентин Валерійович

Керівник: _____ к. т. н., доцент кафедри КІТ Климова Асія Сабирівна

Нормоконтролер: _____ Ігор РАЙЧЕВ

Київ – 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет Комп'ютерних наук та технологій

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, освітньо-професійна програма: 12 “Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології”

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Аліна САВЧЕНКО

«_____» _____ 2023р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи студента

Драгана Валентина Валерійовича
(прізвище, ім'я, по батькові)

1. **Тема роботи:** «Мовні моделі для аналізу великих об'ємів тексту»
Затверджена наказом ректора від 29.09.2023 р. за № 1976/ст.
2. **Термін виконання роботи:** з 02.10.2023р. до 31.12.2022р.
3. **Вихідні дані до проекту:** Бібліотека node.js, бібліотека express.js, мова JavaScript, Векторна база даних, OpenAI API, backend фреймворк NestJS.
4. **Зміст пояснювальної записки:** загальна характеристика, аналіз тексту, взаємодія великих мовних моделей та векторних баз даних, аналіз методів та технологій для інтеграції великих мовних моделей, розробка додатку аналізу тексту з використанням великих мовних моделей.
5. **Перелік обов'язкового ілюстративного матеріалу:** інформативні рисунки,
6. графічні скріншоти роботи системи, слайди презентації в MS PowerPoint.

7. Календарний план-графік

№ п/п	Завдання	Термін виконання	Підпис керівника
1	Загальний аналіз необхідних технологій.	02.10.23 – 05.10.23	
2	Аналіз структури існуючих технологій мовного аналізу з метою їх впровадження.	06.10.23 – 10.10.23	
3	Планування дизайну інтерфейсу користувача.	11.10.23 – 13.10.23	
4	Аналіз та вибір технологій для розробки серверної частини додатку.	14.10.23 – 15.10.23	
5	Аналіз методів та технологій для формування структури бази даних.	16.10.23 – 20.10.23	
6	Інтеграція API рішення штучного інтелекту	21.10.23 – 31.10.23	
7	Розробка основних розділів засобу	01.11.23 – 20.11.23	
8	Створення доповіді та презентації	21.11.23 – 28.11.23	
9	Оформлення пояснювальної записки дипломного проекту	29.11.23 – 10.12.23	

Дата видачі завдання: 02 жовтня 2023р.

Керівник дипломної роботи: _____ Асія КЛИМОВА
(підпис керівника)

Завдання прийняв до виконання: _____ Валентин ДРАГАН
(підпис випускника)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи “Мовні моделі для аналізу великих об'ємів тексту”: містить 109 сторінок, 25 рисунків, 1 таблицю, 10 наукових джерел, 2 додатки.

Ключові слова: БІБЛІОТЕКА, ARTIFICIAL INTELLIGENCE, ВЕКТОРНІ БАЗИ ДАНИХ, ВЕЛИКІ МОВНІ МОДЕЛІ, АНАЛІЗ ТЕКСТУ, WEB – ЗАСТОСУНОК

Мета кваліфікаційної роботи: створення застосунку аналізу тексту з використанням великих мовних моделей.

Об'єкт дослідження: Аналіз значних об'ємів тексту великими мовними моделями.

Предмет дослідження: Аналіз значних об'ємів тексту великими мовними моделями.

Метод дослідження: методи порівняльного аналізу методів реалізації, обробка літературних джерел.

Результат проекту: Розроблений застосунок з інтегрованою мовною моделлю, що здатний аналізувати значні об'єми тексту.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	9
ВСТУП.....	10
РОЗДІЛ 1. ВВЕДЕННЯ У ВЕЛИКІ МОВНІ МОДЕЛІ ТА АНАЛІЗ ТЕКСТУ	11
1.1. Попередня інформація та мотивація	13
1.2. Цілі та обсяг роботи.....	16
1.3. Методологія	17
1.4. Еволюція мовних моделей	19
1.5. Техніки аналізу тексту.....	21
1.6. Застосування великих мовних моделей.....	24
1.7. Технічний стек для аналізу тексту.....	26
1.8. JavaScript для бекенду та фронтенду	29
1.9. Векторна база даних для зберігання документації	31
1.9.1. Організація текстових представлень.....	31
1.9.2. Забезпечення ефективного аналізу подібності	31
1.9.3. Підтримка кластеризації та категоризації	32
1.9.4. Підтримка контекстного аналізу та розпізнавання образів	32
1.9.5. Оптимізація навчання моделі та її тонке налаштування.....	32
1.9.6. Сприяння передачі знань і співпраці	33
1.10. Хмарні рішення для використання мовних моделей	33
1.10.1. Хмарна інфраструктура та масштабованість.....	34
1.10.2. Розподілені обчислення та паралельна обробка.....	34
1.10.3. Керовані послуги для модельного хостингу.....	34
1.10.4. Прискорене апаратне забезпечення для висновків	35
1.10.5. Оптимізація витрат і управління ресурсами.....	35
1.10.6. Глобальна доступність і резервування.....	35
1.10.7. Заходи безпеки та відповідності	36
1.11. Методи OpenAI API для локального застосування	36
РОЗДІЛ 2. ЗАСТОСУВАННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ	39
2.1. Аналіз настроїв та відгуків користувачів	40
2.1.1. Покращення обслуговування клієнтів: III Контакт-центру Google	40
2.1.2. Рекомендація щодо вмісту: Алгоритм персоналізації Netflix	40
2.1.3. Розширений мовний переклад: Amazon AWS Translate	41
2.1.4. Аналіз ринку та прогнозування.....	41
2.2. Розпізнавання іменованих сутностей та посилання на них.....	42
2.2.1. Визначення іменованих сутностей.....	42
2.2.2. Виклик еталонного вирішення	42
2.2.3. LLM на передовій	43
2.2.4. Програми реального світу	43
2.3. Підсумовування та генерація тексту	44
2.3.1. Можливості генерації тексту.....	44
2.3.2. Сутність резюмування тексту.....	45
2.3.3. LLM в дії.....	45

2.3.4.	Етичні міркування.....	46
2.4.	Системи питань та відповідей.....	46
2.4.1.	Розуміння систем запитань і відповідей.....	46
2.4.2.	LLM: підвищення можливостей контролю якості.....	47
2.4.3.	Виклики та обмеження.....	47
2.4.4.	Етичні та суспільні наслідки.....	48
2.5.	Переклад тексту та локалізація.....	48
2.5.1.	Нюанс перекладу проти локалізації.....	49
2.5.2.	LLM: За межами буквального перекладу.....	49
2.5.3.	Локалізація за допомогою LLM.....	50
2.5.4.	Вплив у реальному світі.....	50
2.5.5.	Проблеми та обмеження.....	50
РОЗДІЛ 3.	ВИВЧЕННЯ ВИПАДКІВ.....	52
3.1.	Автоматизація підтримки клієнтів за допомогою чат-ботів.....	52
3.1.1.	Передумови: цифровий зсув у підтримці клієнтів.....	53
3.1.2.	Впровадження чат-ботів на базі LLM.....	53
3.1.3.	Отримані переваги.....	53
3.1.4.	Проблеми, які виникли.....	54
3.1.5.	Більш широкі наслідки.....	54
3.2.	Покращення процесу створення контенту та стратегій маркетингу.....	55
3.2.1.	Еволюція ландшафту створення вмісту.....	55
3.2.2.	LLM у створенні контенту.....	55
3.2.3.	Удосконалення маркетингових стратегій за допомогою LLM Insights ..	56
3.2.4.	Реалізовані вигоди.....	56
3.2.5.	Виклики, що виникли.....	56
3.3.	Покращення аналізу та підсумовування документів.....	57
3.3.1.	Традиційні проблеми аналізу документів.....	57
3.3.2.	LLM в дії: зміна парадигми.....	58
3.3.3.	Покращені методи підсумовування.....	58
3.3.4.	Реалізовані переваги.....	59
3.3.5.	Навігація викликів.....	59
РОЗДІЛ 4.	РЕАЛІЗАЦІЯ.....	60
4.1.	Готовність інфраструктури та системні вимоги.....	61
4.1.1.	Оцінка технічної інфраструктури для інтеграції LLM.....	61
4.1.2.	Управління життєвим циклом даних для LLM.....	62
4.1.3.	Оцінка набору навичок і навчання команди.....	62
4.2.	Стратегії інтеграції LLM.....	63
4.2.1.	Визначення випадків використання та встановлення цілей.....	63
4.2.2.	Розробка робочого процесу для прийняття LLM.....	64
4.2.3.	Управління змінами та культурна трансформація.....	64
4.3.	Робота з OpenAI API.....	65
4.3.1.	Налаштування середовища OpenAI API.....	65
4.3.2.	Найкращі методи автентифікації та безпеки.....	66
4.3.3.	Обробка запитів і відповідей API.....	67
4.4.	Розробка бекенда за допомогою NestJS.....	68

4.4.1.	Основи NestJS і налаштування проекту.....	68
4.4.2.	Створення RESTful API за допомогою NestJS	69
4.4.3.	Інтеграція OpenAI API у NestJS Framework	69
4.4.4.	Обробка та журналювання помилок	71
4.5.	Розробка зовнішнього інтерфейсу за допомогою JavaScript	72
4.5.1.	Розробка зручних інтерфейсів для функцій LLM	73
4.5.2.	Асинхронний зв'язок із серверною частиною.....	73
4.5.3.	Міркування безпеки зовнішнього інтерфейсу.....	74
4.6.	Включення векторної бази даних	75
4.6.1.	Вступ до векторних баз даних та їх значення	76
4.6.2.	Вибір правильної векторної бази даних	76
4.6.3.	Інтеграція векторної бази даних із NestJS	77
4.7.	Повна інтеграція стека.....	79
4.7.1.	Поєднання між інтерфейсом і сервером: найкращі методи	80
4.7.2.	Забезпечення потоку даних і цілісності між компонентами	81
4.7.3.	Тестування та забезпечення якості	81
4.8.	Розгортання та масштабування.....	82
4.8.1.	Стратегії розгортання програм LLM.....	82
4.8.2.	Моніторинг і налаштування продуктивності	83
4.8.3.	Масштабування програм LLM для високої доступності та навантаження	84
4.9.	Технічне обслуговування та постійне вдосконалення.....	85
4.9.1.	Впровадження циклів зворотного зв'язку для безперервного навчання	85
4.9.2.	Оновлення моделей і робота з зміщенням.....	86
4.9.3.	Довгострокова підтримка та оновлення екосистеми LLM	86
4.10.	Тематичні дослідження та реальні приклади	87
4.10.1.	Аналіз успішних реалізацій LLM.....	87
4.10.2.	Висновки, отримані зі складних розгортань.....	88
4.10.3.	Майбутні тенденції та нові практики впровадження LLM.....	88
РОЗДІЛ 5. ВИКЛИКИ ТА ЗАСОБИ ЇХ ПОДОЛАННЯ		91
5.1.	Етичні вимоги до генерації тексту.....	91
5.1.1.	Принципи створення етичних текстів.....	92
5.1.2.	Можливість неправильного використання	92
5.1.3.	Захист етичної чесності	92
5.1.4.	Роль нормативно-правової бази	93
5.2.	Впорядкування упереджень та справедливості	94
5.2.1.	Поширеність упередженості в LLM.....	94
5.2.2.	Регуляторні підходи.....	94
5.2.3.	Зменшення упередженості та сприяння справедливості	95
5.2.4.	Майбутні напрямки.....	95
5.3.	Забезпечення конфіденційності та безпеки даних	96
5.3.1.	Ландшафт загроз	96
5.3.2.	Методи збереження конфіденційності.....	96
5.3.3.	Регуляторні наслідки	97
5.3.4.	Майбутні напрямки.....	97

РОЗДІЛ 6. РЕЗУЛЬТАТИ ТА ОБГОВОРЕННЯ.....	98
6.1. Оцінка продуктивності мовних моделей.....	98
6.2. Реальний вплив та ефективність.....	101
РОЗДІЛ 7. ПЕРСПЕКТИВИ МАЙБУТНЬОГО РОЗВИТКУ	106
7.1. Дослідження та досягнення.....	106
7.1.1. Покращена взаємодія людини з комп'ютером	106
7.1.2. Мультимодальні можливості.....	107
7.1.3. Вирішення питання фактичної надійності.....	107
7.1.4. Розріджені експертні моделі.....	107
7.1.5. Інвестиції та залучення громади	108
7.1.6. Пристосування до мінливих світових знань.....	108
7.2. Трансформуючий вплив	108
7.2.1. Покращена взаємодія людини з комп'ютером	109
7.2.2. Динаміка суспільства	109
7.2.3. Вплив на охорону здоров'я	109
7.2.4. Наукові дослідження та написання	110
7.2.5. До узагальнення та універсальності.....	110
7.3. Інтеграція з новими технологіями	110
7.3.1. Багатомодельна співпраця:.....	111
7.3.2. Охорона здоров'я:.....	111
7.3.3. Технологічні платформи:.....	111
7.3.4. Міжгалузевий вплив:	112
7.3.5. Розширені можливості:.....	112
7.3.6. Розробка програмного забезпечення:	112
ВИСНОВКИ	113
Внесок у галузь	114
Вплив на промисловість та науку	116
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ	119
ДОДАТОК А	121
ДОДАТОК Б.....	123

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

LLM – Large Language Model (Велика мовна модель).

API – application programming interface (Прикладний програмний інтерфейс)

NMT – Neural machine translation (Нейронний машинний переклад)

GPU – graphics processing unit (Графічний процесор)

TPU – Tensor Processing Unit (Тензорний процесор)

AI – Artificial Intelligence (Штучний інтелект)

GPT – Generative pre-trained transformer (Генеративний попередньо навчений трансформер)

ВСТУП

В останні роки сфера обробки природної мови (Natural Language Processing, NLP) стала свідком чудової трансформації, спричиненої появою великих мовних моделей. Ці моделі, здатні обробляти величезні обсяги текстових даних, революціонізували наш підхід до завдань, пов'язаних з аналізом тексту. Буде представлено представлено вичерпний огляд ландшафту, що оточує великі мовні моделі, їх значення в області обробки природної мови і їх застосування в різних сферах аналізу тексту.

Експоненціальне зростання даних у цифрову епоху зумовило потребу в складніших інструментах для отримання значущої інформації з текстової інформації. Великі мовні моделі, навчені величезним об'ємам тексту, стали потужним інструментом у цій спробі. Їхня здатність розуміти контекст, робити висновки про семантику та генерувати зв'язний текст вивела їх на передовий край у методологіях аналізу тексту.

Мається на меті з'ясувати фундаментальні концепції, що лежать в основі великих мовних моделей. Буде досліджено їхню архітектуру, методології навчання та ключові досягнення, які привели до їхніх поточних можливостей. Крім того, буде описано різні типи великих мовних моделей та їхні сильні сторони в різних програмах аналізу тексту.

Крім того, будуть окреслені цілі та обсяг цієї дипломної роботи. Встановлюючи тверде розуміння великих мовних моделей та їхньої відповідності аналізу тексту, буде закладено основу для наступних розділів, де докладніше буде розглянуто конкретні програми та методології.

У підсумку, ця робота служить воротами в динамічну сферу великих мовних моделей і їх ключову роль в аналізі тексту. Буде досліджено дослідити безліч шляхів, якими ці моделі змінили сферу обробки природної мови, і як вони продовжують розширювати межі того, що можна досягти в царині аналізу тексту.

РОЗДІЛ 1. ВВЕДЕННЯ У ВЕЛИКІ МОВНІ МОДЕЛІ ТА АНАЛІЗ ТЕКСТУ

Поширення цифрової інформації призвело до безпрецедентного обсягу текстових даних, доступних в Інтернеті. Це величезне сховище тексту охоплює широкий діапазон доменів, від новинних статей і публікацій у соціальних мережах до наукових робіт і літературних творів. Ефективна навігація та витяг значущої інформації з цього колосального корпусу тексту стала критичною проблемою в різних сферах, включаючи обробку природної мови, пошук інформації, аналіз настроїв і рекомендації щодо вмісту.

В останні роки поява великих мовних моделей революціонізувала підхід до завдань аналізу тексту. Ці моделі, оснащені мільйонами або навіть мільярдами параметрів, мають здатність вивчати складні шаблони, розуміти контекст і генерувати текст, схожий на людську відповідь. Їхні вражаючі можливості сприяли прогресу в таких завданнях, як машинний переклад, аналіз настроїв, узагальнення тексту та відповіді на запитання тощо.

Цей розділ служить фундаментальним дослідженням у сфері великих мовних моделей та їх застосування в аналізі тексту. Він має на меті забезпечити повне розуміння базових принципів, методологій і проблем, пов'язаних із використанням потужності цих моделей для отримання значущої інформації з текстових даних. Шляхом систематичного вивчення ключових концепцій буде закладено основу для глибшого вивчення їх практичної реалізації та потенційного впливу в наступних розділах.

У цьому вступному розділі буде досліджено еволюцію мовних моделей, виділяючи ключові віхи, які проклали шлях до розробки великомасштабних моделей.

Кафедра КІТ (47)				НАУ 23 28 98 000 ПЗ			
<i>Виконав</i>	<i>Драган В.В.</i>			МОВНІ МОДЕЛІ ДЛЯ АНАЛІЗУ ВЕЛИКИХ ОБ'ЄМІВ ТЕКСТУ	<i>Літера</i>	<i>аркуш</i>	<i>аркушів</i>
<i>Керівник</i>	<i>Климова А.С.</i>					11	28
<i>Консульт.</i>					УС-212М		122
<i>Н. контроль</i>	<i>Райчев І. Е.</i>						

Також буде проаналізовано основні принципи, які лежать в основі їхньої роботи, проливаючи світло на такі поняття, як механізми самоуважності, архітектури трансформерів і цілі попереднього навчання.

Крім того, буде обговорено широкий спектр додатків, які виграють від цих моделей, ілюструючи їхню універсальність і можливість адаптації до різних областей аналізу тексту. На Рис. 1.1 продемонстровано широкі можливості детальної адаптації до більшості можливих потреб пов'язаних з аналізом тексту.

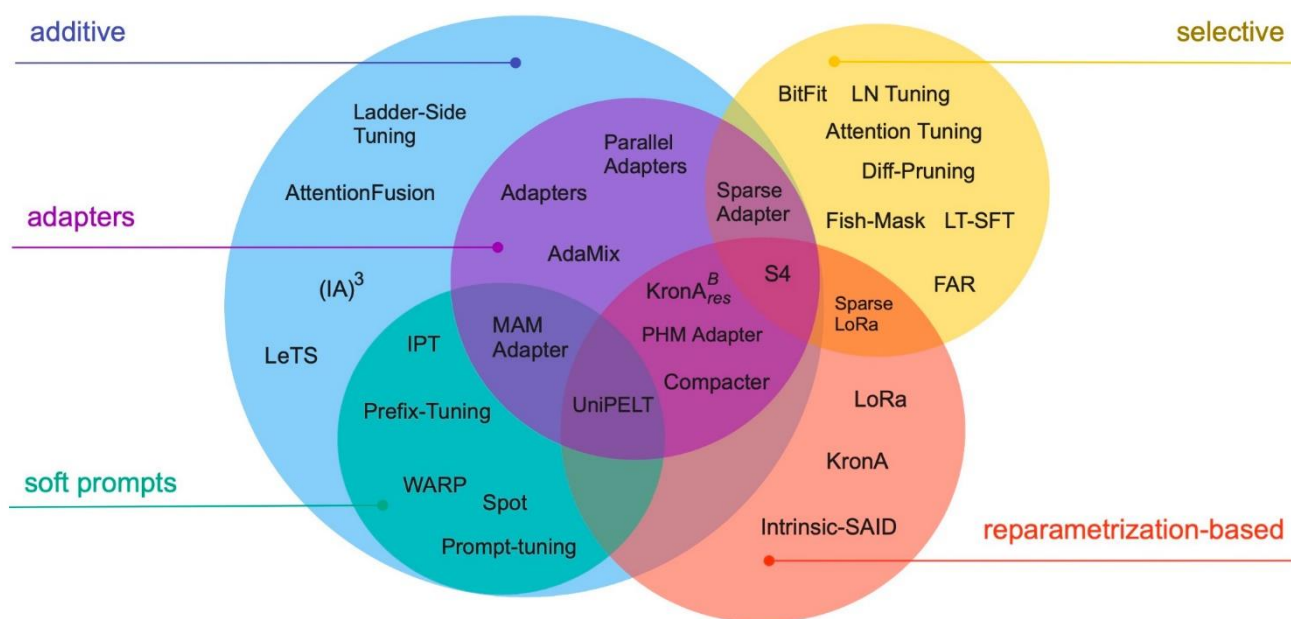


Рис. 1.1. Широкі можливості адаптації моделей для найрізноманітніших потреб.

Занурюючись у фундаментальні принципи та мотивацію великих мовних моделей, буде підготовано основу для всебічного дослідження їх практичної реалізації в наступних розділах. За допомогою цього запиту є на меті розкрити потенціал цих моделей як безцінних інструментів для розуміння та вилучення ідей із величезного моря текстової інформації, яка визначає цифрову епоху.

1.1. Попередня інформація та мотивація

Епоха інформації започаткувала еру, коли текст є не лише надлишковим, але й центральним у тому, як ми спілкуємося, навчаємося та приймаємо рішення. З експоненційним зростанням цифрового вмісту, починаючи від повідомлень у соціальних мережах і закінчуючи науковими журналами, потреба отримувати значущі ідеї з цього величезного сховища тексту ніколи не була такою гострою. У відповідь на цей виклик великі мовні моделі з'явилися як потужні інструменти в області обробки природної мови.

Мотивація цього підрозділу є те, щоб закласти основу для всебічного дослідження великих мовних моделей та їх застосування в аналізі тексту. Буде почато з вивчення історичної еволюції мовних моделей, простежуючи їх розвиток від ранніх систем, заснованих на правилах, до керованих даними моделей глибокого навчання, які домінують сьогодні. Цей історичний контекст дає цінну інформацію про мотивації та виклики, які спонукали до розвитку мовного моделювання.

Поглиблюючись, буде досліджено ключовий момент у сфері обробки природної мови. Архітектура “Трансформер” з її механізмами самоуважності зробила революцію в тому, як мовні моделі розуміють і створюють текст. Цей прорив відіграв вирішальну роль у створенні широкомасштабних мовних моделей, здатних виконувати широкий спектр мовних завдань.

Буде пролито світло на важливість завдань попереднього навчання, які передбачають навчання мовних моделей на величезній кількості текстових даних для вивчення тонкощів мови. Попереднє навчання стало наріжним каменем розробки сучасної мовної моделі та відіграло важливу роль у покращенні їхньої продуктивності в різних подальших завданнях.

У цьому розділі наголошується важливість на широкому спектрі додатків, які отримують вигоду від великих мовних моделей. Незалежно від того, чи це автоматичний переклад, аналіз настроїв, резюмування тексту чи відповідь на запитання, ці моделі продемонстрували свою адаптивність і універсальність у вирішенні різноманітних завдань аналізу тексту.

У цифрову епоху величезний обсяг тексту, який генерується щодня, вражає. Від публікацій у соціальних мережах до академічних статей, від оглядів клієнтів до новин, текст є засобом, за допомогою якого поширюється інформація та передаються ідеї. З цим вибухом текстових даних потреба в потужних інструментах для аналізу та вилучення з них значущої інформації стала першочерговою. Великі мовні моделі з'явилися як трансформаційна технологія в цьому відношенні, пропонуючи безпрецедентні можливості для розуміння, створення та обробки тексту в масштабі.

Еволюція обробки природної мови була відзначена значними віхами, причому великі мовні моделі являють собою недавній прорив. Традиційно ці системи покладалися на створені вручну правила та розробку функцій, які часто не вловлювали нюанси людської мови. З появою глибокого навчання та доступністю величезних обсягів текстових даних відбулася зміна парадигми. Великі мовні моделі, такі як OpenAI GPT-3, продемонстрували виняткову майстерність у широкому діапазоні завдань НЛП, від аналізу настроїв до машинного перекладу та навіть творчого письма (Рис. 1.2).

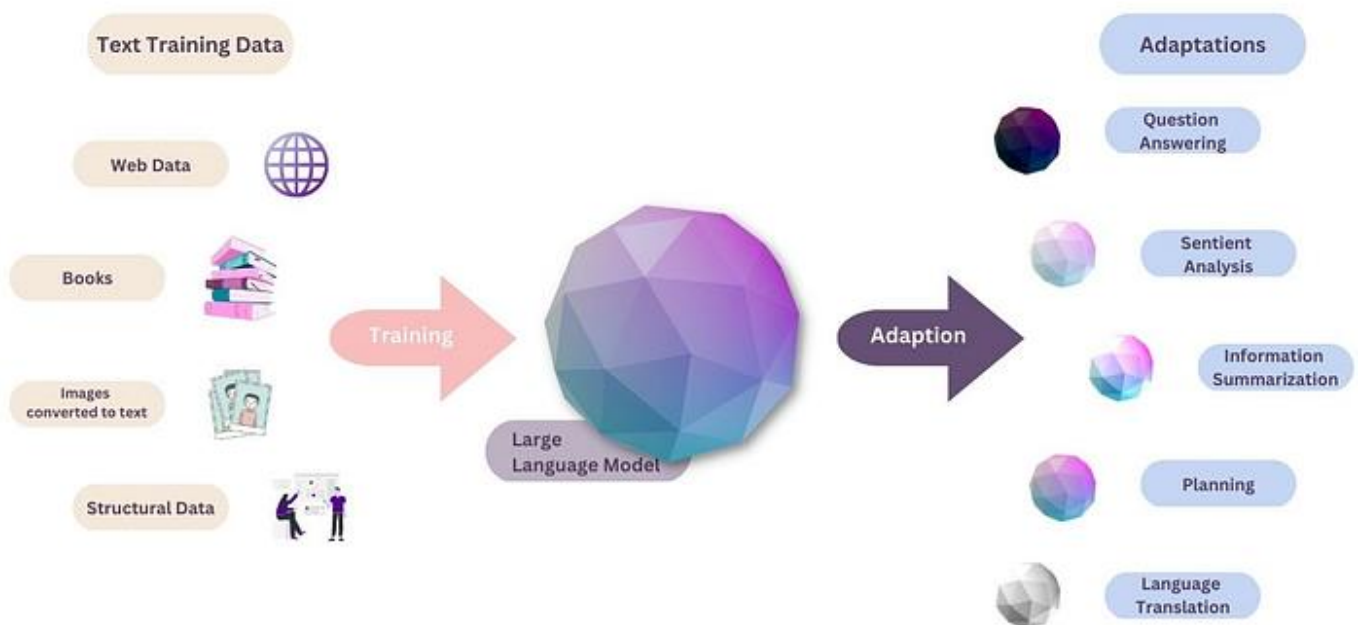


Рис. 1.2. Широкі можливості використання мовних моделей

Ця робота прагне заглибитися в сферу великих мовних моделей та їх застосування в аналізі тексту. Вона має на меті забезпечити повне розуміння базових принципів, архітектури та методологій навчання, які дають змогу цим моделям обробляти текст із такою безпрецедентною точністю та плавністю. Крім того, ця робота досліджуватиме різноманітний спектр завдань, які можна виконати за допомогою цих моделей, проливаючи світло на їхній потенціал революціонізувати галузі та сфери, що залежать від текстових даних.

Мотивація цього дослідження походить від зростаючого попиту на передові методи аналізу тексту в різних секторах. Такі галузі, як маркетинг, обслуговування клієнтів, журналістика та охорона здоров'я, все більше покладаються на вище згадані технології для отримання розуміння, автоматизації процесів і покращення взаємодії з користувачами. Використовуючи можливості великих мовних моделей, організації можуть розблокувати нові рівні ефективності та результативності роботи з текстовими даними.

Крім того, розуміння внутрішньої роботи великих мовних моделей має вирішальне значення для вирішення етичних міркувань і потенційних упереджень, які можуть виникнути в автоматизованому аналізі тексту. Оскільки ці моделі стають все більш поширеними в реальних додатках, вкрай важливо ретельно перевірити їхню поведінку, виявити потенційні недоліки та працювати над пом'якшенням будь-яких небажаних наслідків.

У цьому контексті ця робота також досліджуватиме етичні аспекти розгортання великих мовних моделей в аналізі тексту. Буде детально розглянуті питання, пов'язані з упередженістю, справедливістю та прозорістю, проливаючи світло на виклики та найкращі практики, які необхідно враховувати під час відповідального використання цієї технології. Вивчаючи як можливості, так і підводні камені, це дослідження має на меті забезпечити цілісну перспективу інтеграції великих мовних моделей у програми аналізу тексту.

1.2. Цілі та обсяг роботи

Основною метою цієї роботи є комплексне дослідження можливостей і потенційних застосувань великих мовних моделей у сфері аналізу тексту. Для досягнення цієї головної мети було визначено кілька конкретних цілей і напрямків.

Перш за все, у цьому розділі ми заглибимося в архітектурні основи великих мовних моделей, з'ясовуючи принципи, які лежать в основі їх чудової продуктивності в розумінні та створенні тексту. Сфера охоплює поглиблене дослідження архітектури нейронних мереж, механізмів уваги та методологій навчання, які завершилися розробкою цих моделей.

Крім того, дослідження має на меті каталогізувати та проаналізувати різноманітний набір завдань, які великі мовні моделі можуть вміло виконувати в рамках аналізу тексту. Це включає, але не обмежується: аналіз настроїв, розпізнавання іменованих об'єктів, узагальнення тексту, машинний переклад і системи відповідей на запитання. Вивчаючи можливості цих моделей у спектрі завдань, прагнеться надати вичерпний огляд їхньої універсальності та потенційної корисності.

Крім того, ця робота почне детальний розгляд практичного застосування великих мовних моделей у різних галузях і областях. Будуть представлені тематичні дослідження та реальні приклади, щоб проілюструвати, як ці моделі зараз використовуються для вирішення складних проблем і вдосконалення процесів, пов'язаних з аналізом тексту. Цей емпіричний підхід запропонує цінне уявлення про відчутні переваги та проблеми, пов'язані з розгортанням великих мовних моделей у практичних умовах.

Етичні міркування є критично важливим аспектом цього дослідження. Оскільки великі мовні моделі стають невід'ємною частиною автоматизованого аналізу тексту, вкрай важливо вирішувати потенційні упередження, проблеми справедливості та проблеми прозорості, які можуть виникнути. У цьому розділі буде розглянуто етичні аспекти використання цих моделей, запропоновано основу для

відповідального розгортання та підкреслено стратегії пом'якшення небажаних наслідків.

Обсяг цієї роботи охоплює порівняльний аналіз різних великих мовних моделей, включаючи варіації в архітектурі, навчальні дані та продуктивність різних завдань НЛП. Систематично оцінюючи ці моделі, є намір забезпечити детальне розуміння їхніх відносних сильних і слабких сторін, дозволяючи практикам і дослідникам приймати обґрунтовані рішення щодо їх вибору та розгортання.

Зрештою, цей розділ закладає основу для поглибленого дослідження різноманітних аспектів великих мовних моделей у аналізі тексту. Визначаючи чіткі цілі та сфери фокусування, це дослідницьке завдання має на меті зробити внесок у ширше розуміння технологій НЛП та їх трансформаційного потенціалу в різних галузях і сферах застосування.

1.3. Методологія

Методологія, використана в цій роботі, розроблена для систематичного дослідження та аналізу можливостей і застосувань великих мовних моделей в аналізі тексту. У наступних розділах описано ключові компоненти методології дослідження.

- **Збір даних:** буде зібрано різноманітний і репрезентативний набір текстових даних із різних джерел і доменів. Цей набір даних слугуватиме основою для навчання та оцінки великих мовних моделей у ряді завдань аналізу тексту. Особливу увагу буде приділено забезпеченню якості та різноманітності даних, що є вирішальним для отримання надійних результатів.
- **Вибір моделі:** для експериментування розглядатимуться кілька великих мовних моделей, включаючи, але не обмежуючись ними, GPT-3, BERT та їх варіанти. У процесі відбору керуватиметься їхньою популярністю, доступністю та ефективністю виконання контрольних завдань. Ці моделі будуть налаштовані за потреби для конкретних завдань аналізу тексту.

- **Експериментальний план:** набір контрольованих експериментів буде розроблено для оцінки ефективності вибраних моделей для різних завдань аналізу тексту. Це включатиме такі завдання, як аналіз настроїв, резюмування тексту, розпізнавання іменованих об'єктів, машинний переклад та інші. Кожен експеримент буде ретельно розроблено для оцінки конкретних аспектів продуктивності моделі.
- **Метрики оцінки продуктивності:** стандартні метрики оцінки, що стосуються кожного завдання аналізу тексту, використовуватимуться для оцінки продуктивності вибраних моделей. Для кількісного вимірювання продуктивності моделі використовуватимуться такі показники, як точність, оцінка F1, оцінка BLEU і розгубленість (perplexity).
- **Етичні міркування:** враховуючи етичні аспекти розгортання великих мовних моделей, етичні рекомендації та міркування будуть включені в методологію. Щоб забезпечити відповідальне використання цих моделей під час аналізу тексту, проводитимуться аудити упередженості, оцінки справедливості та оцінки прозорості.
- **Тематичні дослідження:** будуть проведені реальні приклади, щоб продемонструвати практичне застосування великих мовних моделей у різних галузях і областях. Ці тематичні дослідження дадуть цінну інформацію про те, як ці моделі зараз використовуються для вирішення конкретних завдань аналізу тексту.
- **Аналіз та інтерпретація даних:** дані, зібрані в результаті експериментів і тематичних досліджень, будуть проаналізовані за допомогою статистичних і якісних методів. Будуть виведені моделі, тенденції та ідеї, пов'язані з продуктивністю та застосуванням великих мовних моделей в аналізі тексту.
- **Порівняння та обговорення:** буде проведено порівняльний аналіз для оцінки відносних сильних і слабких сторін різних великих мовних

моделей. Отримані дані будуть детально обговорені, щоб отримати детальне розуміння варіацій моделі та їх наслідків.

- **Наслідки та рекомендації:** на основі результатів дослідження будуть окреслені наслідки для прийняття великих мовних моделей в аналізі тексту. Будуть надані рекомендації для практиків, дослідників і політиків, особливо в контексті етичних міркувань.

Цей розділ завершиться коротким викладом використаної методології, підкреслюючи ключові кроки, зроблені для дослідження цілей дослідження. Це також стане основою для наступних розділів, де детально обговорюватимуться результати дослідження та їхні наслідки.

1.4. Еволюція мовних моделей

Еволюція великих мовних моделей є важливою віхою в області обробки природної мови (ОПМ). У цьому розділі простежується траєкторія прогресу, який привів до розробки цих потужних моделей, висвітлюючи ключові прориви та інновації.

Початок сучасної обробки природної мови можна простежити до систем, заснованих на правилах, і статистичних моделей, які спиралися на створені вручну функції. Хоча ці підходи продемонстрували майстерність у певних завданнях, вони не змогли охопити нюанси складності людської мови. Перехід до методів, заснованих на машинному навчанні, ознаменував зміну парадигми, дозволивши моделям вивчати шаблони безпосередньо з даних.

Впровадження вбудованих слів, таких як Word2Vec і GloVe, революціонізувало спосіб представлення слів у векторних просторах. Ці вбудовування вловлювали семантичні зв'язки та прокладали шлях для більш контекстуально багатих мовних представлень. Цей перехід від розріджених одноразових кодувань до щільних безперервних представлень значно покращив продуктивність систем обробки мови.

Механізми привернення уваги стали ключовою інновацією з появою архітектури Transformer. Представлена в основоположній статті «Увага — це все, що вам потрібно», модель Transformer продемонструвала ефективність механізмів самоуважності для фіксації довготривалих залежностей у тексті. Ця архітектура сприяла розпаралелюванню, забезпечуючи ефективне навчання на великих наборах даних.

Прориви в апаратному прискоренні, зокрема наявність потужних графічних процесорів (GPU) і тензорних процесорів (TPU), зіграли вирішальну роль у збільшенні розміру та складності моделей NLP. Це уможливило навчання моделей із безпрецедентною кількістю параметрів, кульмінацією чого стала розробка таких моделей, як GPT-3 із 175 мільярдами параметрів та GPT-4 з 1.7 трильйонами параметрів.

Серія моделей GPT (Generative Pre-trained Transformer), розроблена OpenAI, є переломним моментом в еволюції великих мовних моделей. У цих моделях використовується підхід попереднього навчання та тонкого налаштування, коли вони спочатку проходять попереднє навчання на величезних обсягах текстових даних, а потім налаштовуються на конкретні завдання. GPT-3, найбільший у серії, продемонстрував неперевершену продуктивність у різноманітному діапазоні завдань мовної обробки.

Однак зростання розміру моделі не обійшлося без компромісів. Навчання великих моделей потребує значних обчислювальних ресурсів і може призвести до проблем під час розгортання через їх ресурсомісткий характер. Крім того, етичні міркування, пов'язані з упередженнями в даних і поведінці моделей, привернули все більше уваги, оскільки великі мовні моделі стають більш поширеними.

Оскільки область мовної обробки продовжує розвиватися, еволюція великих мовних моделей, ймовірно, триватиме. Майбутні дослідження можуть бути зосереджені на вдосконаленні методології навчання, вирішенні етичних проблем і вивченні нових архітектур для подальшого підвищення можливостей цих моделей.

Розуміння цієї еволюції (Рис. 1.3) надає цінний контекст для оцінки можливостей і потенційного впливу великих мовних моделей у програмах аналізу тексту.

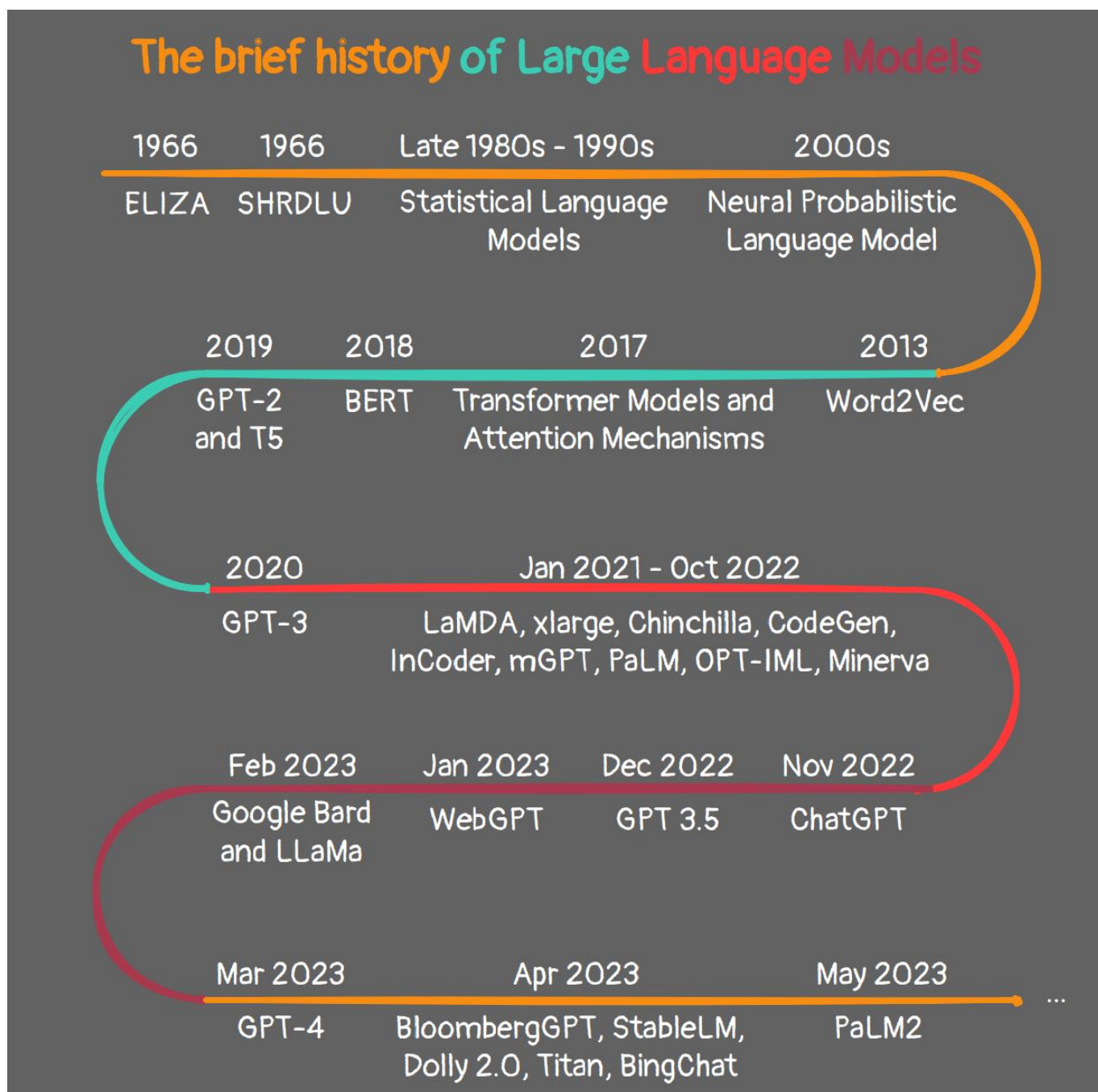


Рис. 1.3. Еволюція мовних моделей

1.5. Техніки аналізу тексту

Аналіз тексту охоплює різноманітний набір технік і методологій, які дають змогу отримувати значущі ідеї з текстових даних. У цьому розділі розглядається

низка методів аналізу тексту, від базових методів до просунутих підходів, надаючи вичерпний огляд інструментів, доступних для розуміння й обробки тексту.

- **Токенізація:** токенізація — це фундаментальний процес поділу тексту на окремі лексеми, як правило, слова або підсловні одиниці. Цей крок є важливим для наступних завдань аналізу тексту, оскільки він встановлює основну одиницю аналізу.
- **Попередня обробка тексту:** текстові дані часто вимагають попередньої обробки для видалення шуму, наприклад знаків пунктуації, стоп-слов і тегів HTML. Для скорочення слів до їхніх базових форм можна застосувати такі методи, як формування основи та лематизація.
- **Класифікація тексту:** класифікація тексту передбачає призначення попередньо визначених міток або категорій текстовим документам. Алгоритми машинного навчання, такі як Naive Bayes, Support Vector Machines і моделі глибокого навчання, зазвичай використовуються для цього завдання, що робить його корисним для таких завдань, як аналіз настроїв, виявлення спаму та класифікація тем.
- **Розпізнавання іменованих сутностей:** цей етап має на меті ідентифікувати та класифікувати сутності, такі як імена людей, організації, місця та дати, у тексті. Умовні випадкові поля і моделі глибокого навчання підвищили точність розпізнавання іменованих сутностей.
- **Аналіз настроїв:** Аналіз настроїв визначає настрої чи емоції, виражені в тексті, які зазвичай класифікуються як позитивні, негативні чи нейтральні. Підходи на основі лексики, моделі машинного навчання та великі мовні моделі часто використовуються для аналізу настроїв.
- **Резюмування тексту:** Техніка резюмування тексту згущує довші тексти в коротші, зв'язні анотації, зберігаючи важливу інформацію. Зазвичай використовуються екстрактивні та абстрактні методи підсумовування, причому абстрактні методи генерують резюме більш схожим на людину способом.

- **Машинний переклад:** машинний переклад передбачає переклад тексту з однієї мови на іншу. Статистичні моделі, системи на основі правил і моделі нейронного машинного перекладу (NMT) зробили революцію в цій галузі, причому моделі NMT демонструють чудову якість перекладу.
- **Моделювання тем:** методи моделювання тем, такі як прихований розподіл Діріхле і факторизація невід'ємної матриці, виявляють приховані теми всередині корпусу документів, уможливаючи виявлення та організацію вмісту.
- **Кластеризація тексту:** кластеризація тексту об'єднує схожі документи на основі їх вмісту. Для цієї мети зазвичай використовуються такі методи, як кластеризація k-середніх та ієрархічна кластеризація.
- **Системи запитань-відповідей:** системи запитань-відповідей використовують запитання природною мовою для отримання відповідної інформації з текстового корпусу. Ці системи можуть бути засновані на правилах або на основі великих мовних моделей, які продемонстрували чудову ефективність у цій області.
- **Розбір залежностей:** аналіз залежностей аналізує граматичну структуру речень, виявляючи зв'язки між словами. Ця техніка необхідна для розуміння синтаксису речень і може бути використана в різних завданнях НЛП.
- **Зв'язування іменованих сутностей:** цей етап розширює розпізнавання іменованих сутностей, не лише розпізнаючи сутності, але й пов'язуючи їх із зовнішніми базами знань або онтологіями. Це допомагає усунути неоднозначність сутностей і збагатити результати аналізу тексту.
- **Глибоке навчання для аналізу тексту.** Моделі глибокого навчання, включаючи згорточні нейронні мережі (CNN), рекурентні нейронні мережі (RNN) і трансформатори, значно вдосконалили сучасний аналіз тексту. Трансформери, зокрема, показали виняткову продуктивність у різних завданнях мовної обробки.
- **Етичні міркування в аналізі тексту:** Етичні міркування, такі як пом'якшення упередженості, справедливості і конфіденційності, є

невід’ємною частиною відповідального аналізу тексту. Методи виявлення та пом’якшення зміщення в текстових даних і поведінці моделі є вирішальними для вирішення цих проблем.

Розуміння та застосування цих методів аналізу тексту має важливе значення для ефективної обробки текстових даних і отримання інформації з них, завдання, яке стає все більш критичним в епоху перевантаження інформацією. Ці методи служать будівельними блоками для практичного застосування великих мовних моделей в аналізі тексту, як обговорюється в наступних розділах.

1.6. Застосування великих мовних моделей

Інтеграція великих мовних моделей у різні сфери та галузі започаткувала нову еру можливостей аналізу тексту. У цьому розділі досліджується різноманітний діапазон застосувань, у яких ці моделі продемонстрували виняткову продуктивність, революціонізувавши спосіб обробки текстових даних в організаціях.

- **Підтримка клієнтів і чат-боти:** великі мовні моделі використовуються в програмах підтримки клієнтів для автоматизації відповідей, надання миттєвої допомоги та обробки широкого спектру запитів клієнтів. Чат-боти на основі таких моделей, як GPT-3, продемонстрували здатність вести природні розмови та ефективно вирішувати проблеми користувачів.
- **Створення вмісту та копірайтинг:** у сфері створення вмісту великі мовні моделі чудово справляються зі створенням тексту, схожого на створений людиною, у різних стилях і тонах. Від публікацій у блогах і статей до маркетингових текстів і підписів у соціальних мережах, ці моделі пропонують потужний інструмент для творців контенту.
- **Інформаційно-пошукові та пошукові системи:** великі мовні моделі підвищують точність і релевантність результатів пошукової системи. Вони можуть розуміти складні запити та отримувати відповідну контексту інформацію, покращуючи роботу користувачів у системах пошуку інформації.

- **Аналіз медичного тексту та охорона здоров'я:** у сфері охорони здоров'я великі мовні моделі допомагають виконувати такі завдання, як узагальнення клінічних записів, розпізнавання сутностей та аналіз медичної літератури. Вони сприяють швидшому та точнішому пошуку інформації в медичному контексті.
- **Юридичний аналіз і аналіз контрактів:** Професіонали в галузі права отримують переваги від великих мовних моделей у таких завданнях, як аналіз контрактів, узагальнення юридичних документів і допомога в дослідженні. Ці моделі можуть аналізувати складну юридичну мову та ефективно отримувати відповідну інформацію.
- **Аналіз настроїв для дослідження ринку:** у дослідженні ринку великі мовні моделі використовуються для аналізу настроїв, щоб оцінити громадську думку щодо продуктів, послуг або брендів. Це дає цінну інформацію для прийняття рішень і маркетингових стратегій.
- **Послуги мовного перекладу:** великі мовні моделі значно підвищили якість послуг машинного перекладу. Їхня здатність вловлювати контекст і нюанси в мові привела до більш точного та природного звучання перекладів.
- **Модерування та фільтрація вмісту:** Великі мовні моделі відіграють вирішальну роль у модеруванні вмісту, автоматично виявляючи та відфільтровуючи неприйнятний або шкідливий вміст з онлайн-платформ. Це особливо важливо для підтримки безпечних та інклюзивних онлайн-спільнот.
- **Резюмування та агрегування новин:** у сфері журналістики великі мовні моделі використовуються для автоматичного створення резюме статей новин. Вони допомагають швидко засвоїти великий обсяг інформації, роблячи її більш доступною для читачів.
- **Освіта та електронне навчання:** великі мовні моделі можуть покращити освітні матеріали, створюючи пояснення, надаючи відгуки про завдання та навіть діючи як віртуальні викладачі. Вони полегшують персоналізований досвід навчання для студентів.

- **Персоналізований маркетинг і рекомендації:** Аналізуючи поведінку та вподобання клієнтів, великі мовні моделі дозволяють компаніям надавати високоперсоналізовані маркетингові повідомлення та рекомендації щодо продуктів, підвищуючи залученість клієнтів і їхню задоволеність.
- **Юридична звітність і звітність:** у галузях, які підлягають дотриманню нормативних вимог, великі мовні моделі допомагають у створенні точних і відповідних звітів шляхом розуміння та узагальнення юридичних документів і вимог.
- **Творче написання та створення ідей щодо змісту:** письменники та творці використовують великі мовні моделі для мозкового штурму ідей, створення творчого вмісту та подолання письменницького блоку. Ці моделі можуть надавати підказки, пропонувати сюжетні лінії та навіть складати вірші чи фантастику.

Застосування великих мовних моделей продовжує розширюватися в галузях промисловості, демонструючи їх універсальність і потенціал трансформації в обробці текстових даних. Використовуючи можливості цих моделей, організації можуть досягти більшої ефективності, точності та інновацій у своїх спробах аналізу тексту.

1.7. Технічний стек для аналізу тексту

Для реалізації ефективного аналізу тексту з використанням великих мовних моделей надзвичайно важливий надійний технічний стек. У цьому підрозділі описано ключові компоненти та технології, необхідні для створення надійної та масштабованої системи аналізу тексту.

OpenAI API є наріжним каменем інфраструктури аналізу тексту. Це API надає доступ до потужних мовних моделей, що дозволяє програмам генерувати текст, схожий на згенерований людиною, і виконувати широкий спектр завдань мовної обробки. За допомогою API розробники можуть легко інтегрувати можливості

великих мовних моделей у свої програми, розкриваючи потенціал для розширеного аналізу тексту.

Доповнюючи OpenAI API, для полегшення взаємодії з API використовується серверна частина NodeJS. NodeJS добре підходить для обробки асинхронних операцій, що робить його ідеальним вибором для керування запитами та відповідями до та з OpenAI API. Його керована подіями архітектура забезпечує ефективний зв'язок з API, уможливаючи обробку текстових даних у реальному часі.

Таблиця 1. Порівняння мов програмування для машинного навчання.

Особливість	Python	Node.js
Популярність у Спільноті ML	Надзвичайно популярний, основна мова для ML.	Менш популярний; зростаючий інтерес у спільноті ML.
Екосистема Бібліотек	Багата екосистема бібліотек з інструментами, такими як TensorFlow, PyTorch, SciKit-Learn.	Обмежена, але розвивається. Головним чином використовує TensorFlow.js для ML.
Продуктивність	Загалом повільніший через те, що є інтерпретованою мовою.	Швидший для I/O завдань, але не є інтуїтивно швидшим для CPU-затратних завдань ML.
Асинхронна Обробка	Підтримує асинхронні операції, але не є основною функцією.	Основна функція; відмінно справляється з асинхронними операціями.
Легкість Вивчення	Широко вважається легким для вивчення та використання.	Також вважається зручним для користувача, але вимагає розуміння JavaScript і його особливостей.
Підтримка Спільноти	Велика підтримка спільноти та документації.	Добра підтримка спільноти, особливо у веб-розробці.
Інтеграція з	Сильна підтримка маніпуляцій з	Здатний, але може вимагати

Особливість	Python	Node.js
Джерелами Даних	даними та інтеграції.	додаткових зусиль у порівнянні з Python.
Великі Мовні Моделі	Дуже підходить з широкою підтримкою та прикладами.	Можливо, але менш прямолінійно; менше прикладів та випадкових досліджень.
Розгортання	Легке розгортання моделей ML, з багатьма фреймворками, що підтримують Python.	Добре для інтеграції моделей ML у JavaScript-орієнтоване середовище.
Обробка в Реальному Часі	Здатний, але може вимагати більше налаштувань.	Природно підходить для застосунків реального часу через свою неблокуючу природу.

Проаналізувавши таблицю 1 можна побачити, що Python є оптимальним для більшої частини завдань машинного навчання, проте з урахуванням що поточний проект спирається на веб технології для взаємодії з користувачем та делегує відповідальність за логіку машинного навчання зовнішньому сервісу, з метою демонстрації можливостей у контрольованих умовах буде використано NodeJS.

Для зберігання відповідної інформації та даних, які можуть бути використані моделлю для аналізу, реалізована векторна база даних. Ця база даних служить репозиторієм для структурованих і неструктурованих даних, що забезпечує ефективний пошук і зберігання інформації. Векторні бази даних особливо вправно справляються з великовимірними даними, завдяки чому вони добре підходять для зберігання вбудованих текстів та інших відповідних функцій, які використовуються в процесі аналізу.

Amazon Web Services (AWS) використовується для розміщення векторної бази даних, що забезпечує надійність, масштабованість і надійність. AWS надає комплексний набір хмарних служб, які забезпечують високу доступність і

резервування даних, критично важливих для обробки великих обсягів текстових даних. Масштабованість інфраструктури AWS дозволяє безперешкодно розширюватися в міру зростання розміру та складності набору даних.

Завдяки інтеграції API OpenAI, серверної частини NodeJS, векторної бази даних і хостингу AWS технічний стек забезпечує потужну основу для аналізу тексту. Цей стек забезпечує плавну інтеграцію великих мовних моделей, ефективне керування взаємодіями API, ефективне зберігання та пошук відповідної інформації, а також надійну та масштабовану інфраструктуру для підтримки процесу аналізу. Разом ці компоненти утворюють цілісну екосистему, яка надає додаткам розширені можливості аналізу тексту.

1.8. JavaScript для бекенду та фронтенду

JavaScript, який часто називають мовою Інтернету, відіграє ключову роль у забезпеченні безперебійної взаємодії між серверними та зовнішніми компонентами системи аналізу тексту. У цій главі досліджується обґрунтування використання JavaScript як для серверної, так і для зовнішньої розробки, підкреслюється його універсальність і ефективність в інтеграції OpenAI API та управлінні векторною базою даних, розміщеною в хмарі.

Повсюдне поширення JavaScript у веб-розробці робить його природним вибором для впровадження серверної частини системи аналізу тексту. Зростання таких технологій, як Node.js, дозволило розробникам використовувати JavaScript для операцій на стороні сервера, створюючи уніфіковану кодову базу для всієї програми.

Синхронні та асинхронні операції: керована подіями неблокуюча природа JavaScript є особливо вигідною для взаємодії із зовнішніми службами, такими як OpenAI API. Асинхронні операції гарантують, що програма залишається відмовостійкою, навіть якщо обробляє кілька одночасних запитів.

Ефективна інтеграція API: Node.js, середовище виконання для виконання JavaScript на сервері, відмінно справляється з обробкою запитів HTTP та керування

відповідями. Це робить його ідеальним для бездоганної інтеграції з API OpenAI, забезпечуючи ефективне спілкування та обмін даними.

Модульність і екосистема пакетів: обширна екосистема пакетів npm надає велику кількість готових модулів і бібліотек. Ця екосистема прискорює розробку, пропонуючи легкодоступні інструменти для таких завдань, як обробка HTTP-запитів, керування базами даних і обробка даних.

Масштабованість і продуктивність: однопоточкова архітектура Node.js, керована подіями, забезпечує високу масштабованість для обробки великої кількості одночасних з'єднань. Ця масштабованість особливо важлива для додатків, які вимагають обробки текстових даних у режимі реального або майже в реальному часі.

Домінування JavaScript у інтерфейсі веб-додатків забезпечує плавну інтеграцію з серверною частиною, створюючи цілісну взаємодію з користувачем. Ця інтеграція має вирішальне значення для представлення результатів і взаємодії з користувачами в режимі реального часу.

Динамічні та інтерактивні інтерфейси користувача: JavaScript разом із HTML і CSS дозволяє створювати динамічні та інтерактивні інтерфейси користувача. Це важливо для відображення результатів, дозволу користувача вводити дані та забезпечення безперебійної взаємодії з системою аналізу тексту.

Оновлення та відгуки в режимі реального часу: фреймворки JavaScript, такі як React і Vue.js, полегшують розробку додатків у режимі реального часу, забезпечуючи миттєві оновлення та відгуки для користувачів під час взаємодії з системою. Цей динамічний характер підвищує залучення користувачів і швидкість реакції.

Обробка даних на стороні клієнта: здатність JavaScript виконувати код у веб-переглядачі користувача дає змогу інтерфейсу виконувати певні завдання обробки даних без необхідності надсилати додаткові запити до сервера. Це може призвести до швидшого часу відповіді та зменшення навантаження на сервер.

Використовуючи можливості JavaScript як на серверній, так і на зовнішній частині, розробники можуть створити потужну уніфіковану систему аналізу тексту.

Ефективна взаємодія серверної частини з API OpenAI і векторною базою даних, розміщеною в хмарі, у поєднанні з динамічними користувацькими інтерфейсами, увімкненими JavaScript на зовнішній частині, забезпечує безперебійну та стабільну взаємодію з користувачем. Така інтеграція технологій демонструє універсальність і ефективність JavaScript у створенні передових програм аналізу тексту.

1.9. Векторна база даних для зберігання документації

У сфері аналізу тексту роль векторної бази даних виходить за межі звичайного зберігання даних. Він служить динамічним репозиторієм не лише для організації та зберігання текстової інформації, але й як фундаментальний компонент у процесі отримання значущих ідей. У цьому розділі розглядається основна функція векторної бази даних як системи зберігання документації, з'ясовується її значення для підвищення ефективності та результативності програм аналізу тексту.

1.9.1. Організація текстових представлень

Векторна база даних відмінно справляється з інкапсуляцією складних текстових даних у структурованому та доступному форматі. Завдяки використанню векторних вставок текстова інформація перетворюється на багатовимірне числове представлення. Ці представлення служать мостом між необробленим текстом і обчислювальними процесами, які лежать в основі аналізу тексту. Індукуючи та класифікуючи ці вбудовані елементи, база даних забезпечує швидкий пошук і маніпулювання відповідною інформацією.

1.9.2. Забезпечення ефективного аналізу подібності

Однією з ключових переваг використання векторної бази даних є її здатність полегшувати аналіз подібності. Використовуючи векторне вбудовування, база даних дозволяє обчислювати показники подібності між документами або текстовими

елементами. Ця можливість є важливою в таких завданнях, як пошук інформації, де ідентифікація документів із подібним вмістом або контекстом є першорядною. Завдяки ефективним механізмам індексування та пошуку векторна база даних прискорює процес ідентифікації відповідних документів.

1.9.3. Підтримка кластеризації та категоризації

Векторні представлення, що зберігаються в базі даних, забезпечують основу для кластеризації та категоризації текстових даних. Використовуючи такі методи, як кластеризація k-середніх або ієрархічна кластеризація, можна згрупувати документи з подібними вбудованими елементами. Ця функціональність є неоціненною в таких завданнях, як тематичне моделювання, де систематизація корпусу документів у зв'язані категорії є важливою. Завдяки використанню обчислювальних можливостей векторної бази даних процес кластеризації стає точним і доцільним.

1.9.4. Підтримка контекстного аналізу та розпізнавання образів

Коли векторні вбудовування зберігаються в базі даних, вони служать основою для контекстного аналізу та розпізнавання образів. Ця можливість особливо важлива в таких завданнях, як аналіз настроїв і розпізнавання об'єктів. База даних дозволяє отримувати релевантну контекстну інформацію, дозволяючи моделі розрізняти тенденції настроїв або розпізнавати сутності в тексті. Ця контекстна обізнаність має вирішальне значення для отримання точної та значущої інформації з текстових даних.

1.9.5. Оптимізація навчання моделі та її тонке налаштування

Векторна база даних — це не просто пасивне сховище; він відіграє активну роль у навчанні та налагодженні моделей. Надаючи багате джерело анотованих і контекстуалізованих даних, база даних покращує здатність моделі навчатися та

узагальнювати на прикладах. Цей ітеративний процес уточнення моделі полегшується здатністю бази даних ефективно отримувати та надавати відповідні навчальні дані.

1.9.6. Сприяння передачі знань і співпраці

Окрім своїх технічних функцій, векторна база даних служить сховищем інституційних знань. Вона інкапсулює колективний інтелект спроб аналізу тексту організації, забезпечуючи безперебійну передачу знань між членами команди та проектами. База даних служить центральним центром для спільних зусиль, забезпечуючи безперервність і узгодженість ініціатив з аналізу тексту.

По суті, векторна база даних виходить за рамки своєї ролі простої системи зберігання; він постає як динамічний і невід'ємний компонент у конвеєрі аналізу тексту. Завдяки використанню можливостей векторних вбудованих даних ця база даних не лише організовує та зберігає текстові представлення, але й дає змогу моделям отримувати важливі ідеї. Його багатогранні функціональні можливості значною мірою сприяють ефективності та дієвості програм аналізу тексту, що робить його незамінним активом у сфері обробки природної мови.

1.10. Хмарні рішення для використання мовних моделей

Розгортання та масштабованість великих мовних моделей є критичним аспектом їх практичного застосування. Хмарні рішення забезпечують надійну інфраструктуру для розміщення, керування та масштабування цих моделей для задоволення різноманітних обчислювальних вимог. У цьому розділі розглядаються різноманітні хмарні рішення, створені для задоволення обчислювальних вимог великих мовних моделей, що дозволяє організаціям ефективно використовувати їхній трансформаційний потенціал.

1.10.1. Хмарна інфраструктура та масштабованість

Хмарні платформи, такі як Amazon Web Services (AWS), Google Cloud Platform (GCP) і Microsoft Azure, пропонують широкий спектр послуг і ресурсів, оптимізованих для робочих навантажень машинного навчання. Ці платформи забезпечують еластичну обчислювальну потужність, дозволяючи бездоганну масштабованість для адаптації до ресурсомісткого характеру великих мовних моделей. Використовуючи функції автоматичного масштабування та динамічний розподіл ресурсів, організації можуть адаптуватися до змінних навантажень, забезпечуючи оптимальну продуктивність.

1.10.2. Розподілені обчислення та паралельна обробка

Хмарні рішення сприяють розподіленим обчисленням, що є важливою можливістю для навчання та розгортання великих мовних моделей. Такі методи, як архітектури серверів параметрів і розподілені інфраструктури навчання, дозволяють одночасно навчати моделі на кількох GPU або TPU. Такий розподілений підхід значно прискорює процес навчання, дозволяючи ефективно розробляти великомасштабні моделі.

1.10.3. Керовані послуги для модельного хостингу

Хмарні постачальники пропонують керовані послуги для розміщення великих мовних моделей, що спрощує процес розгортання. Такі сервіси, як AWS SageMaker, Google Cloud AI Platform і Azure Machine Learning, дозволяють організаціям розгортати моделі без потреби в складному управлінні інфраструктурою. Ці платформи пропонують інструменти для версії моделі, моніторингу та автоматичного масштабування, спрощуючи операційні аспекти розміщення великих мовних моделей.

1.10.4. Прискорене апаратне забезпечення для висновків

Щоб досягти продуктивності в режимі реального часу або майже в режимі реального часу, хмарні рішення надають доступ до спеціалізованого апаратного забезпечення, наприклад GPU і TPU. Ці прискорювачі підвищують швидкість і ефективність моделювання, дозволяючи організаціям швидко обробляти текстові дані. Хмарні постачальники пропонують ряд опцій GPU і TPU, що дозволяє організаціям вибирати апаратні конфігурації, які відповідають їхнім конкретним вимогам.

1.10.5. Оптимізація витрат і управління ресурсами

Хмарні рішення пропонують низку моделей ціноутворення, включаючи екземпляри на вимогу, зарезервовані та точкові екземпляри, що дозволяє організаціям оптимізувати витрати на основі моделей використання. Крім того, хмарні платформи надають інструменти моніторингу та управління ресурсами для відстеження використання ресурсів, ефективного розподілу ресурсів і визначення можливостей для економії коштів. Така деталізація контролю дозволяє організаціям ефективно керувати своїми обчислювальними ресурсами.

1.10.6. Глобальна доступність і резервування

Хмарні провайдери пропонують глобальну мережу центрів обробки даних, що забезпечує низьку затримку доступу до обчислювальних ресурсів з різних регіонів. Крім того, хмарні рішення включають стратегії резервування та реплікації даних, захищаючи від потенційних збоїв або збоїв. Цей глобальний слід і архітектура резервування підвищують надійність і доступність великих мовних моделей.

1.10.7. Заходи безпеки та відповідності

Хмарні платформи впроваджують надійні заходи безпеки та рамки відповідності для захисту конфіденційних даних і забезпечення дотримання нормативних вимог. Такі функції, як віртуальні приватні хмари (VPC), шифрування в стані спокою та під час передачі, керування ідентифікацією та доступом (IAM) і сертифікати відповідності забезпечують безпечне середовище для розгортання великих мовних моделей і керування ними.

Використовуючи хмарні рішення, розроблені для великих мовних моделей, організації можуть подолати обчислювальні проблеми, пов'язані з розгортанням і масштабуванням цих моделей. Поєднання еластичної обчислювальної потужності, можливостей розподіленого обчислення, спеціалізованого апаратного забезпечення та інструментів оптимізації витрат дає змогу організаціям використовувати весь потенціал великих мовних моделей масштабованим, економічно ефективним і безпечним способом.

1.11. Методи OpenAI API для локального застосунку

OpenAI API надає програмний доступ до мовних моделей OpenAI, дозволяючи розробникам взаємодіяти з ними у своїх програмах. Воно дає змогу створювати текст, виконувати такі завдання, як відповіді на запитання, узагальнення, переклад тощо. API підтримує як одноповоротні (single-turn) завдання, так і багатоповоротні (multi-turn) розмови.

Ключові компоненти:

Кінцева точка API: Кінцева точка API – це URL-адреса, за якою є можливість надіслати HTTP-запити для взаємодії зі службами OpenAI. Базовою кінцевою точкою для API OpenAI є <https://api.openai.com/v1>.

Автентифікація: Щоб використовувати API, знадобиться ключ API, наданий OpenAI. Цей ключ використовується для автентифікації запитів і надання доступу до служб OpenAI.

Методи HTTP: API переважно використовує метод POST для надсилання запитів до кінцевої точки. Це використовується для таких завдань, як створення тексту, підсумовування тощо. Деякі виклики API також можуть використовувати метод GET для певних операцій.

Запит корисного навантаження: Корисне навантаження запиту – це дані, які надсилаються до API. Для таких завдань, як створення тексту, передаються такі параметри, як `model`, `prompt`, `max_tokens` тощо, залежно від конкретного завдання.

Відповідь: API відповідає даними JSON, які містять результати запитаної операції. Це може включати згенерований текст, відповіді на запитання або іншу відповідну інформацію.

Основний порядок використання:

Ініціалізація клієнта API: У додатку Node.js зазвичай використовується бібліотека, наприклад `axios` або `node-fetch`, щоб надсилати запити HTTP. Необхідно налаштувати клієнт API за допомогою ключа API.

Побудування запиту: Залежно від завдання створюється JSON-об'єкт із потрібними параметрами. Наприклад, для створення тексту можна вказати модель і надати підказку.

надсилання запиту: Необхідно використати вибрану бібліотеку HTTP, щоб надіслати запит POST до кінцевої точки OpenAI API. Включіть свій ключ API в заголовки запиту для автентифікації.

Обробка відповіді: Коли API обробить запит, він поверне відповідь JSON. Необхідно витягнути відповідну інформацію з відповіді, наприклад створений текст або відповідь (Рис. 4).

```
const axios = require('axios');

const apiKey = 'YOUR_API_KEY';
const apiUrl = 'https://api.openai.com/v1/...'; // Specify the specific API

const prompt = 'Once upon a time...';
const maxTokens = 100;

const headers = {
  'Authorization': `Bearer ${apiKey}`,
  'Content-Type': 'application/json',
};

const requestBody = {
  model: 'davinci', // Specify the desired model
  prompt: prompt,
  max_tokens: maxTokens,
};

axios.post(apiUrl, requestBody, { headers })
  .then(response => {
    const generatedText = response.data.choices[0].text;
    console.log(generatedText);
  })
  .catch(error => console.error(error));
```

Рис. 1.4. Приклад використання OpenAI API для генерації тексту.

РОЗДІЛ 2. ЗАСТОСУВАННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

У сучасну цифрову епоху здатність опрацьовувати та інтерпретувати величезні обсяги тексту має вирішальне значення для незліченних галузей, починаючи від охорони здоров'я та закінчуючи фінансами та від засобів масової інформації до наукових кіл. В основі цієї революції в аналізі тексту лежить клас обчислювальних моделей, відомих як великі мовні моделі (LLM). Ці моделі, які часто ґрунтуються на техніках глибокого навчання та забезпечуються прогресом у апаратному забезпеченні, знищили бар'єри, з якими раніше стикалися у сфері обробки природної мови (NLP). Як наслідок, великі мовні моделі тепер є лідерами в безлічі додатків, охоплюючи як прості, так і складні рішення.

У попередньому розділі було запропоновано розуміння основоположних концепцій великих мовних моделей, нюансів їхньої архітектури та еволюційної траєкторії. У цій главі розглядається глибше, проливаючи світло на багатогранне застосування великих мовних моделей. Від покращення розмовних здібностей чат-ботів до допомоги в аналізі літератури; від генерації прогнозованого тексту до аналізу настроїв; і від допомоги в медичних діагнозах, заснованих на розповідях пацієнтів, до допомоги дослідникам у перегляді літератури – великі мовні моделі трансформують саму структуру нашої взаємодії з текстом.

У цьому розділі буде отримано всебічне розуміння поточного ландшафту програм, які використовують LLM, виклики, які вирішуються цими моделями, і трансформаційний вплив, який вони зробили. Також буде розглянуто деякі реальних сценарії, коли LLM затьмарили традиційні методи, а в деяких випадках навіть людський досвід.

Кафедра КІТ (47)				НАУ 23 28 98 000 ПЗ			
Виконав	Драган В.В.			МОВНІ МОДЕЛІ ДЛЯ АНАЛІЗУ ВЕЛИКИХ ОБ'ЄМІВ ТЕКСТУ	Літера	аркуш	аркушів
Керівник	Климова А.С.					39	12
Консульт.					УС-212М 122		
Н. контроль	Райчев І. Е.						

2.1. Аналіз настроїв та відгуків користувачів

Інтеграція великих мовних моделей (LLM) в робочі процеси великих корпорацій — це не просто технологічне оновлення; це стратегічна трансформація, яка впливає на зацікавлених сторін, від внутрішніх команд до кінцевих споживачів. Одним з найважливіших аспектів, який визначає успіх цих інтеграцій, є думка та відгуки цих користувачів, які дають цінну інформацію про ефективність, результативність і сфери вдосконалення цих програм. У цьому підрозділі буде досліджено реальні приклади того, як відомі корпорації впроваджують LLM у свої системи, а також подальшу реакцію та відгуки користувачів.

2.1.1. Покращення обслуговування клієнтів: ШІ Контакт-центру Google

Одним із секторів, які суттєво змінили LLM, є обслуговування клієнтів. Яскравим прикладом є AI Contact Center від Google. Ця система використовує можливості LLM, щоб розуміти запити клієнтів і відповідати на них у людський спосіб. Підприємства, які використовують цю технологію, повідомляють про різке скорочення часу очікування та більш ефективне вирішення проблем, що було позитивно сприйнято кінцевими користувачами. Наприклад, клієнти часто діляться відгуками про природний хід розмови та висловлюють вдячність за швидкі й точні відповіді. Однак є також конструктивна критика, коли користувачі припускають, що штучний інтелект може покращити розпізнавання складних емоційних сигналів або суворо технічних запитів.

2.1.2. Рекомендація щодо вмісту: Алгоритм персоналізації Netflix

Netflix, гігант у потоковій індустрії, використовує LLM у своїй системі рекомендацій контенту. Ці складні моделі аналізують величезні набори даних, включаючи текстові огляди та відгуки, щоб зрозуміти вподобання користувачів. Підписники часто висловлювали подив точності рекомендацій, які нібито розуміють

їхні вподобання та настрої. Тим не менш, деякі відгуки вказують на випадкове незадоволення сприйнятим ефектом «ехо-камери», що свідчить про бажання ширшого охоплення вмісту за межі звичок безпосереднього перегляду.

2.1.3. Розширений мовний переклад: Amazon AWS Translate

Глобальні корпорації з різноманітними клієнтськими базами, як-от Amazon, використовують LLM для високоякісних послуг мовного перекладу в реальному часі, необхідних для міжнародного бізнесу. AWS Translate від Amazon допомагає компаніям легко перекладати вміст веб-сайтів, повідомлення служби підтримки клієнтів і описи продуктів. З точки зору відгуків користувачів, компанії підкреслюють безперебійну комунікацію, яку AWS Translate забезпечує з міжнародними зацікавленими сторонами, хоча вони також відзначають, що певні нюанси культурного вираження та ідіоми іноді втрачаються під час перекладу.

2.1.4. Аналіз ринку та прогнозування

У фінансовому секторі такі компанії, як Bloomberg, використовують обробку природної мови (NLP), підмножину LLM, для аналізу ринкових тенденцій, звітів про прибутки, новин і платформ соціальних мереж. Трейдери та аналітики, які використовують термінали Bloomberg, розповіли, як ця інформація в реальному часі дає їм перевагу в прийнятті обґрунтованих рішень. Однак вони також застерігають від надмірної залежності від цих інструментів, наголошуючи на тому, що хоча LLM надають цінну кількісну інформацію, якісне людське судження є настільки ж необхідним.

У кожному з цих прикладів відгуки користувачів відіграють важливу роль у вдосконаленні алгоритмів LLM. Позитивні відповіді свідчать про їх ефективність, тоді як критика є безцінною для висвітлення областей, які потребують подальших досліджень і розробок. Оскільки великі мовні моделі продовжують інтегруватися у нові аспекти корпоративної діяльності, цей цикл зворотного зв'язку з користувачами

матиме ключове значення для формування еволюції цієї технології, гарантуючи, що вона не лише служить своїм функціональним цілям, але й узгоджується з людським аспектом бізнес-операцій.

2.2. Розпізнавання іменованих сутностей та посилання на них

Одним із найбільш критичних застосувань великих мовних моделей (LLM) в аналізі тексту є завдання розпізнавання іменованих об'єктів (named entity recognition, NER) і вирішення їх посилань. Ця здатність відіграє ключову роль у вилученні структурованої інформації з неструктурованого тексту, дозволяючи більш глибоке розуміння та контекстну інтерпретацію вмісту. У цьому підрозділі пояснюється важливість розпізнавання іменованих сутностей, виклики, пов'язані з цим, і поточні рішення, які пропонують LLM.

2.2.1. Визначення іменованих сутностей

На початку важливо визначити, що таке іменовані сутності. У контексті аналізу тексту названі сутності зазвичай стосуються конкретних слів або фраз, які позначають об'єкти реального світу, наприклад осіб (наприклад, «Барак Обама»), організації (наприклад, «Організація Об'єднаних Націй»), місця (наприклад, «гора Еверест») і навіть грошові значення чи дати. Ці сутності відрізняються від загальних термінів або понять, оскільки вони представляють унікальні екземпляри або сутності, які можна ідентифікувати в реальному світі.

2.2.2. Виклик еталонного вирішення

Після ідентифікації сутностей наступним кроком є розуміння їхнього контексту та вирішення посилань. Це відоме як кореферентна роздільна здатність. Розглянемо речення: «Ганна пішла в магазин, тому що їй потрібні були продукти». Хоча «Ганна» — це названа сутність (особа), «вона» — це посилання на цю названу

сутність. Вирішення таких посилань має вирішальне значення для повного розуміння розповіді тексту або отримання повної інформації. Це стає особливо складним у довгих текстах із кількома сутностями та переплетеними посиланнями.

2.2.3. LLM на передовій

Сучасні LLM, що базуються на архітектурах глибокого навчання, показали надзвичайну майстерність у задачах розв'язання як NER, так і кореферентного вирішення. Традиційні методи значною мірою поклалися на системи, засновані на правилах, або простіші моделі машинного навчання, які часто мали проблеми зі складними реченнями чи нюансованими посиланнями. Навпаки, програми LLM, такі як серія GPT OpenAI або BERT від Google, використовують величезні обсяги навчальних даних для розпізнавання шаблонів і контекстів, що дозволяє їм ідентифікувати та пов'язувати об'єкти з високою точністю.

2.2.4. Програми реального світу

Розпізнавання іменованих сутностей та їхніх посилань має широке застосування:

Аналіз новин: ЗМІ та дослідники використовують цю можливість, щоб автоматично отримувати інформацію з новинних статей, підсумовуючи події, учасників і місця.

Охорона здоров'я: медичні записи часто містять розповіді пацієнтів, де лікарі згадують симптоми, діагнози та лікування. Розпізнавання цих сутностей допомагає в організації й аналізі даних.

Фінансовий сектор: у фінансах такі об'єкти, як назви компаній, біржові символи та посилання на валюту, вилучаються зі звітів, новин або аналізу для підтримки автоматизованих торгових систем або аналізу ринку.

Юридична сфера: в юридичних документах визначення юридичних осіб, як-от фізичних осіб, компаній або регуляторних органів, і розуміння їхньої ролі чи контексту в тексті може допомогти в аналізі або дослідженні справи.

Підсумовуючи, розпізнавання іменованих сутностей та їхніх посилань є основою багатьох завдань аналізу тексту. Великі мовні моделі, завдяки своїй здатності обробляти величезну кількість даних і розпізнавати складні закономірності, зробили революцію в цій області, пропонуючи неперевершену точність і ефективність. Оскільки ці моделі продовжують розвиватися й удосконалюватися, ми можемо очікувати ще більш складні та детальніші можливості аналізу тексту в найближчому майбутньому.

2.3. Підсумовування та генерація тексту

Оскільки наша цифрова ера породжує потік текстової інформації, здатність генерувати послідовний, релевантний текст і стисло підсумовувати довгі документи стає все більш важливою. Великі мовні моделі (LLM) швидко стали важливими для вирішення цих завдань. У цьому підрозділі розглядаються можливості великих мовних моделей у створенні та резюмуванні текстів, окреслюється їхнє значення, основні виклики та трансформаційні наслідки для різних секторів.

2.3.1. Можливості генерації тексту

Генерація тексту означає здатність машини створювати зв'язний і релевантний контексту текст на основі заданих підказок або даних. У той час як попередні моделі створювали спрощені та часто повторювані результати, сучасні LLM можуть генерувати вміст, який у багатьох випадках неможливо відрізнити від тексту, написаного людиною.

Застосування генерації тексту:

Створення контенту: від дописів у блогах до новинних статей, LLM можуть допомогти авторам, генеруючи чернетки вмісту або пропонуючи вдосконалення.

Розмовні агенти: чат-боти та віртуальні помічники, такі як Siri від Apple або Alexa від Amazon, використовують генерацію тексту, щоб відповідати на запити користувачів у людській манері.

Творче написання: LLM також наважуються створювати вірші, оповідання та сценарії, демонструючи свій потенціал у мистецьких областях.

2.3.2. Сутність резюмування тексту

Резюмування тексту має на меті стиснути довгі документи в коротші версії, зберігаючи важливу інформацію та повідомлення. Йдеться не просто про скорочення тексту, а й про розуміння контексту, відокремлення основних моментів від деталей і створення стислого, але інформативного резюме.

Застосування резюмування тексту:

Дослідження та наука: підсумовування тривалих дослідницьких робіт або статей дозволяє вченим швидко осягнути основні висновки та методології.

Бізнес-звіти: керівники отримують переваги від узагальнених версій розширених звітів, що дозволяє швидко приймати рішення.

Агрегація новин: платформи можуть надавати користувачам стислі підсумки новинних статей, задовольняючи стрімкі споживчі звички сучасних читачів.

2.3.3. LLM в дії

Сучасні LLM з їхньою глибокою нейронною архітектурою особливо вправно справляються із завданнями генерації та підсумовування. Такі моделі, як GPT-4 або BERT, можна точно налаштувати для певних доменів, створюючи текст, який поважає галузевий жаргон, стиль і нюанси.

Однак, хоча LLM є високоефективними, вони не є непогрішними. Існують проблеми, пов'язані з тим, щоб згенерований текст залишався неупередженим, точним і справді відображав вихідний зміст. Забезпечення надійності та автентичності створеного вмісту є предметом постійних досліджень і вдосконалень.

2.3.4. Етичні міркування

З потужністю LLM у створенні тексту приходять відповідальність. Існують етичні проблеми, пов'язані з наданням оманливої інформації, поширенням упереджень або створенням вмісту, який може змусити читачів подумати, що він створений людиною. У міру того, як великі мовні моделі стають більш інтегрованими в наші цифрові екосистеми, встановлення керівних принципів і гарантій буде мати першорядне значення.

Отже, можливості LLM у створенні тексту та резюмуванні є трансформаційними, спрощують робочі процеси, покращують взаємодію з користувачами та розкривають нові творчі можливості. Однак, як і у випадку з будь-яким потужним інструментом, відповідальне розгортання цих моделей має вирішальне значення для максимізації переваг і пом'якшення потенційних ризиків.

2.4. Системи питань та відповідей

Одне з найбільш динамічних застосувань великих мовних моделей (LLM) в аналізі тексту лежить у сфері систем відповідей на запитання (QA). Такі системи, які прагнуть надавати точні, лаконічні та відповідні контексту відповіді на запити користувачів, відіграють ключову роль у безлічі секторів, від підтримки клієнтів до академічних досліджень. Цей підрозділ заглиблюється в тонкощі систем забезпечення якості, що працюють на базі LLM, досліджуючи їх робочу механіку, значення, проблеми та ландшафт їхніх реальних додатків.

2.4.1. Розуміння систем запитань і відповідей

Системи контролю якості розроблені для імітації людського розуміння та чуйності. Користувач ставить запитання, а система переглядає свою базу знань, якою може бути заздалегідь визначений набір даних або величезний простір Інтернету, щоб отримати найбільш прийнятну відповідь. Суть потужної системи

забезпечення якості полягає не лише в отриманні релевантної інформації, але й у тлумаченні контексту та нюансів питання.

2.4.2. LLM: підвищення можливостей контролю якості

У той час як традиційні системи забезпечення якості часто покладалися на відповідність ключових слів і алгоритми, засновані на правилах, LLM запровадили зміну парадигми. Такі моделі, як BERT, архітектури “Transformer” і серія GPT, навчаються на масивних текстових наборах даних, що дозволяє їм розуміти контекст, семантику та навіть нюанси запитань. Це призводить до більш точних відповідей з урахуванням контексту.

Застосування систем контролю якості на базі LLM:

Підтримка клієнтів: компанії інтегрують QA-ботів на свої веб-сайти, щоб надавати миттєві відповіді на часті запити клієнтів, покращуючи взаємодію з користувачами.

Академічні дослідження: Дослідники можуть запитувати величезні бази даних наукових статей, вилучаючи певну інформацію або розуміючи тенденції досліджень.

Медична діагностика: клініцисти використовують системи забезпечення якості для доступу до відповідної медичної літератури чи історії пацієнтів, допомагаючи прийняти обґрунтоване рішення.

Платформи електронного навчання: студенти можуть задавати запитання віртуальним викладачам, отримуючи пояснення на теми, які вони вважають складними.

2.4.3. Виклики та обмеження

Незважаючи на свою майстерність, системи контролю якості на базі LLM не позбавлені проблем:

Неоднозначні запити: запитання з кількома дійсними інтерпретаціями можуть спричинити труднощі, вимагаючи від систем шукати роз’яснень або робити припущення.

Упередженість і дезінформація: оскільки великі мовні моделі навчаються на величезних наборах даних, вони можуть ненавмисно поширювати упередження або неправдиву інформацію, наявну в цих наборах даних.

Ресурсоміткість: високоякісні відповіді в режимі реального часу часто потребують значних обчислювальних ресурсів, що створює проблеми з масштабованістю.

2.4.4. Етичні та суспільні наслідки

Розвиток систем забезпечення якості, керованих LLM, також породжує кілька етичних проблем:

Конфіденційність даних: особливо в таких секторах, як охорона здоров'я, забезпечення конфіденційності запитів користувачів має першорядне значення.

Надмірна довіра: простота та ефективність систем контролю якості може призвести до надмірної залежності, що потенційно може призвести до відволікання критичних людських суджень.

Переміщення робочих місць: Оскільки роботи з контролю якості стають поширеними в службі підтримки клієнтів, виникають занепокоєння щодо переміщення людей, що вимагає обговорення адаптації та перенавчання робочої сили.

Таким чином, хоча системи запитань-відповідей на базі LLM обіцяють неперевершену ефективність і точність, їх відповідальне та етичне розгортання має вирішальне значення. Збалансування масштабів інновацій з етичними міркуваннями визначатиме, як ці системи формуватимуть нашу майбутню взаємодію з цифровими платформами.

2.5. Переклад тексту та локалізація

У все більш глобалізованому світі подолання мовних бар'єрів має першочергове значення. Це виходить за рамки простого перекладу слів — йдеться

про адаптацію вмісту, щоб культурно та контекстуально резонувати з різноманітною аудиторією. Є можливість дослідити сферу перекладу та локалізації тексту, де великі мовні моделі (LLM) стали трансформаційними агентами. У цьому підрозділі розкрито можливості LLM в цих сферах, підкреслюючи їх значення, виклики та широкий спектр наслідків.

2.5.1. Нюанс перекладу проти локалізації

Хоча переклад і локалізація часто використовуються як взаємозамінні, вони служать різним цілям:

Переклад передбачає перетворення тексту з однієї мови на іншу, зберігаючи цілісність і значення вихідного вмісту.

Локалізація йде глибше, адаптуючи перекладений вміст до культурних, суспільних і контекстуальних нюансів цільової аудиторії.

2.5.2. LLM: За межами буквального перекладу

Традиційні системи перекладу, що працюють на основі правил або статистичних моделей, часто мають проблеми з ідіоматичними виразами, складними структурами речень або культурними посиланнями. LLM, такі як BERT від Google або моделі OpenAI, пропонують зміну парадигми:

Розуміння контексту: Аналізуючи текст, LLM виявляють найбільш точний переклад для слів із кількома значеннями.

Робота з ідіомами та фразами: замість дослівного перекладу магістри LLM можуть ефективно інтерпретувати та перекладати ідіоматичні чи розмовні вирази.

Граматична точність: LLM можуть краще адаптуватися до складних граматичних правил різних мов, забезпечуючи вільний і природний переклад.

2.5.3. Локалізація за допомогою LLM

Локалізація стосується не лише мови; це про почуття, культурну повагу та контекст. LLM з їх величезними наборами навчальних даних демонструють потенціал у цих деталях:

Культурна адаптація: розпізнавання та адаптація контенту, який може бути культурно чутливим або невідповідним.

Формати дати та числа: автоматична адаптація до місцевих умов, будь то формати дати, валюта чи одиниці вимірювання.

Релевантність вмісту: адаптація вмісту відповідно до місцевих інтересів, норм і цінностей.

2.5.4. Вплив у реальному світі

Наслідки перекладу та локалізації, керовані LLM, є глибокими:

Бізнес і комерція: компанії можуть легко виходити на глобальні ринки, забезпечуючи резонанс своїх продуктів, веб-сайтів і маркетингових матеріалів на місцевому рівні.

Освіта: навчальний матеріал можна адаптувати для різних груп студентів, сприяючи інклюзивності та розумінню.

Потокове передавання вмісту: платформи можуть пропонувати фільми, серіали чи документальні фільми з локалізованими субтитрами чи описами, покращуючи враження від перегляду.

2.5.5. Проблеми та обмеження

Незважаючи на те, що LLM відмінно справляються з багатьма завданнями перекладу та локалізації, проблеми залишаються:

Тонкощі та нюанси: певні культурні чи мовні нюанси можуть бути складними для сприйняття навіть складними моделями.

Надмірна стандартизація: ризик створення загального гомогенізованого вмісту, якому бракує справжнього місцевого колориту.

Етичні наслідки. Неправильний переклад або неправильна локалізація можуть призвести до непорозумінь або навіть образити, підкреслюючи важливість людського контролю.

Підсумовуючи, оскільки LLM-и продовжують удосконалювати свої можливості перекладу та локалізації, світ стане більш взаємопов'язаним і доступним. Однак ця подорож потребує поєднання технологічного прогресу з культурною повагою і розумінням.

РОЗДІЛ 3. ВИВЧЕННЯ ВИПАДКІВ

Теоретичні основи та різноманітні застосування великих мовних моделей, описані в попередніх розділах, пропонують широкий огляд їхніх можливостей і значень. Однак, щоб по-справжньому зрозуміти їх трансформаційний потенціал, необхідно заглибитися у сценарії реального світу. У цій главі розглядається детальне прикладне дослідження, де докладно розглядається, як LLM використовуються в контексті конкретної галузі. Завдяки цьому поглибленому дослідженню є прагнення подолати розрив між теоретичними знаннями та практичними застосуваннями, пропонуючи глибоке розуміння нюансів, проблем і досягнень, пов'язаних із впровадженням LLM в реальних умовах.

3.1. Автоматизація підтримки клієнтів за допомогою чат-ботів

У епоху цифрових технологій, де миттєвість і ефективність є найважливішими, традиційні парадигми підтримки клієнтів зазнають швидких змін. Провідною цією зміною є інтеграція чат-ботів — автоматизованих розмовних агентів на базі великих мовних моделей (LLM). Ці чат-боти, хоч і не зовсім нові, зазнали суттєвої еволюції у своїх можливостях і витонченості, в основному завдяки прогресу в технології LLM. Цей підрозділ детально занурюється в тематичне дослідження, яке розглядає впровадження, переваги, проблеми та наслідки автоматизації підтримки клієнтів за допомогою чат-ботів, керованих LLM.

Кафедра КІТ (47)				НАУ 23 28 98 000 ПЗ			
Виконав	Драган В.В.			МОВНІ МОДЕЛІ ДЛЯ АНАЛІЗУ ВЕЛИКИХ ОБ'ЄМІВ ТЕКСТУ	Літера	аркуш	аркушів
Керівник	Климова А.С.					52	8
Консульт.					УС-212М		122
Н. контроль	Райчев І. Е.						

3.1.1. Передумови: цифровий зсув у підтримці клієнтів

Потреба в цілодобовій підтримці клієнтів у поєднанні з постійно зростаючою цифровою базою користувачів вимагала більш масштабованого та ефективного рішення, ніж традиційні телефонні лінії довіри чи підтримка електронною поштою. Чат-боти — спочатку засновані на правилах системи з обмеженими шаблонами відповідей, які тепер перетворилися на динамічних контекстно-залежних агентів.

3.1.2. Впровадження чат-ботів на базі LLM

Серцем сучасних чат-ботів є LLM, яка дозволяє їм:

Розуміти запити: на відміну від своїх попередніх аналогів, які покладалися на конкретні ключові слова, боти, керовані LLM, розуміють наміри та контекст користувача.

Створювати природні реакції: вони залучають користувачів до плавних людських розмов, забезпечуючи ясність і легкість взаємодії.

Навчатися та адаптуватися: з часом ці боти вдосконалюють свої відповіді на основі відгуків користувачів і взаємодії з ними.

3.1.3. Отримані переваги

Перехід до чат-ботів LLM приніс безліч переваг:

Масштабованість: компанії можуть обробляти величезну кількість запитів одночасно, не потребуючи значних людських ресурсів.

Узгодженість: боти надають уніфіковані відповіді без помилок, забезпечуючи стандартизований рівень якості обслуговування.

Збір даних: кожна взаємодія пропонує цінну інформацію про поведінку, потреби та вподобання користувачів, допомагаючи в бізнес-аналітиці.

3.1.4. Проблеми, які виникли

Незважаючи на трансформацію, ця подорож не без перешкод:

Складні запити: ботам, незважаючи на їхню складність, може бути складно з багатогранними або нюансованими запитами.

Людська взаємодія: деякі користувачі все ще віддають перевагу співчуттю, розумінню та імпровізації, які пропонують люди-агенти.

Технічні збої: будучи керованими програмним забезпеченням, боти можуть час від часу стикатися з відмовами чи системними помилками.

3.1.5. Більш широкі наслідки

Широке впровадження чат-ботів LLM має суспільні та бізнес-наслідки:

Ландшафт роботи: у той час як боти обробляють рутинні запити, зростає попит на агентів-людей для вирішення складних проблем, перенесення посадових ролей і обов'язків.

Етичні проблеми: на перший план виходять питання, пов'язані з конфіденційністю даних, згодою користувача та потенційним зловживанням зібраною інформацією.

Бізнес-стратегія: оскільки боти є основним напрямком взаємодії з клієнтами, компанії повинні переконатися, що вони втілюють цінності, тон і політику компанії.

Підсумовуючи, автоматизація підтримки клієнтів за допомогою чат-ботів, керованих LLM, є прикладом зміни парадигми в тому, як компанії взаємодіють із клієнтами та обслуговують їх. Як показує цей приклад, незважаючи на численні переваги, продуманий, збалансований підхід є важливим для подолання пов'язаних проблем і наслідків.

3.2. Покращення процесу створення контенту та стратегій маркетингу

У сучасному інформаційному ландшафті контент, безсумнівно, є головним. Незалежно від того, чи йдеться про дописи в блогах, оновлення в соціальних мережах чи сценарії відео, зміст діє як основний канал між компаніями та їх цільовою аудиторією. З появою великих мовних моделей (LLM) процес створення контенту та подальші маркетингові стратегії зазнали інноваційного перегляду. У цьому підрозділі пропонується глибоке прикладне дослідження того, як LLM були бездоганно вплетені в генерацію контенту та маркетинг, змінюючи традиційні методології та відкриваючи нову еру залучення аудиторії.

3.2.1. Еволюція ландшафту створення вмісту

Створення контенту, яке колись було домінуванням винятково для людської творчості та проникливості, зазнало сейсмічних змін. Сплеск цифрових платформ і невгамовний апетит до свіжого актуального контенту зумовили необхідність більш ефективного підходу з можливістю масштабування. LLM-и стали безцінними союзниками в цій справі, пропонуючи як допомогу, так і автоматизацію.

3.2.2. LLM у створенні контенту

LLM довели важливу роль у різних аспектах створення контенту:

Генерація ідей: LLM можуть аналізувати поточні тенденції, уподобання користувачів і історичні дані, щоб запропонувати теми контенту, які, ймовірно, будуть резонувати з цільовою аудиторією.

Складання та редагування: вони можуть допомогти авторам, пропонуючи вдосконалення, виправляючи помилки та забезпечуючи узгодженість тону.

Локалізація: як обговорювалося в попередніх розділах, LLM можуть адаптувати контент, щоб задовольнити різноманітні мовні та культурні демографічні особливості, розширюючи глобальне охоплення.

3.2.3. Удосконалення маркетингових стратегій за допомогою LLM Insights

Окрім створення контенту, LLM відіграють ключову роль у формуванні маркетингових стратегій:

Аналіз аудиторії. Проаналізувавши величезні набори даних, LLM може надати розуміння вподобань аудиторії, поведінки та моделей залучення.

Персоналізовані кампанії: LLM можуть генерувати індивідуальні маркетингові повідомлення або рекомендації щодо продукту для окремих користувачів, підвищуючи рівень залучення та конверсії.

Аналіз ефективності: після кампанії магістри LLM можуть оцінити ефективність маркетингових стратегій, пропонуючи корисну інформацію для майбутніх починань.

3.2.4. Реалізовані вигоди

Інтеграція LLMs у сфери контенту та маркетингу надала численні переваги:

Ефективність і масштабованість: підприємства можуть створювати високоякісний контент у прискореному темпі, обслуговуючи різноманітні платформи та аудиторію.

Прийняття рішень на основі даних: Маркетингові стратегії, що базуються на отриманих LLM ідеях, як правило, більше узгоджуються з потребами аудиторії та тенденціями ринку.

Економічна ефективність: з часом LLM можуть зменшити накладні витрати, пов'язані з великим контентом і маркетинговими командами, пропонуючи вищу рентабельність інвестицій.

3.2.5. Виклики, що виникли

Ця подорож, хоч і багатообіцяюча, має ряд перешкод:

Надмірна автоматизація: використання LLM може призвести до вмісту, якому бракує справжніх людських емоцій, креативності чи спонтанності.

Етичні наслідки: використання LLM для персоналізованого маркетингу викликає занепокоєння щодо конфіденційності даних користувачів і потенціалу для маніпулятивних кампаній.

Технічні обмеження. Незважаючи на те, що LLM є просунутими, вони іноді можуть неправильно тлумачити нюанси, що призводить до помилок у контенті чи маркетингу.

Таким чином, поєднання LLM зі створенням контенту та маркетинговими стратегіями означає величезну зміну в тому, як компанії спілкуються та взаємодіють зі своєю аудиторією. Це тематичне дослідження підкреслює трансформаційний потенціал цієї інтеграції, а також наголошує на важливості збереження людського контакту в дедалі більш автоматизованому ландшафті.

3.3. Покращення аналізу та підсумовування документів

У час, переповнений інформацією, компанії, дослідники та професіонали часто стикаються з об'ємними документами, які вимагають швидкого розуміння та дій. Традиційні методи аналізу та узагальнення документів вимагали багато часу, трудомісткості та були повні суперечностей. Інтеграція великих мовних моделей (LLM) у цю область віщує нову еру, революціонізуючи спосіб очищення, інтерпретації та використання величезних текстових даних. Цей підрозділ розпочинає тематичне дослідження трансформаційного впливу LLM на аналіз та підсумовування документів, проливаючи світло на складні механізми, відчутні переваги та нові виклики цього нового підходу.

3.3.1. Традиційні проблеми аналізу документів

До появи складних обчислювальних моделей аналіз документів здійснювався переважно вручну:

Перевантаження обсягу: перегляд великої кількості сторінок, щоб отримати релевантну інформацію, був складним і трудомістким.

Неузгодженості: різні люди можуть по-різному інтерпретувати або узагальнювати вміст, що призводить до розбіжностей у отриманих ідеях.

Обробка складності: для точного тлумачення багатозначних документи з нюансами часто потребують досвіду в певній галузі.

3.3.2. LLM в дії: зміна парадигми

Впровадження LLM в аналіз документів і підсумовування було нічим іншим, як революційним:

Швидкий аналіз: LLM можуть швидко обробляти величезні документи, ідентифікуючи та вилючаючи ключову інформацію.

Контекстуальне розуміння: LLM розуміють ширший контекст, гарантуючи, що короткий опис зберігає суть і критичні нюанси оригінального вмісту.

Налаштування: користувачі можуть вказати бажану довжину або фокус короткого опису, а LLM можуть відповідним чином адаптувати результати.

3.3.3. Покращені методи підсумовування

Завдяки LLM мистецтво резюмування досягло помітних успіхів:

Абстрактне підсумовування: замість того, щоб просто витягувати речення, LLM можуть переписати вміст стисло, зберігаючи значення меншою кількістю слів.

Ієрархічні підсумки: для великих документів LLM можуть створювати багаторівневі підсумки, пропонуючи як огляди високого рівня, так і докладні підсумки підрозділів.

Візуальна інтеграція: LLM можуть інтегрувати графічні елементи або діаграми в підсумки, покращуючи розуміння.

3.3.4. Реалізовані переваги

Впровадження LLM в аналіз документів принесло багато переваг:

Ефективність часу: те, на що раніше витрачалися години, якщо не дні, тепер можна виконати за лічені хвилини.

Узгодженість: підсумки, створені LLM, підтримують єдиний стандарт, усуваючи мінливість, спричинену людиною.

Масштабованість: LLM можуть обробляти декілька документів одночасно, задовольняючи сценарії з високим попитом, як-от юридичні перевірки, академічні дослідження або бізнес-аналітика.

3.3.5. Навігація викликів

Шлях, хоч і багатообіцяючий, не позбавлений перешкод:

Надмірна довіра: Виключна залежність від LLM може призвести до недогляду за нюансами, які може розрізнити людський досвід.

Допустимі помилки: LLMs, хоча й просунуті, іноді можуть неправильно інтерпретувати складний або неоднозначний вміст.

Етичні проблеми: потенційне зловживання LLM для спотворення або надмірного спрощення важливих документів породжує етичні дилеми.

Підводячи підсумок, можна сказати, що інтеграція LLM в аналізі та підсумовуванні документів означає трансформаційний зсув, що дозволяє окремим особам і підприємствам орієнтуватися в інформаційному потоці з нововіднайденою гнучкістю. Цей приклад підкреслює величезний потенціал таких досягнень, а також нагадує про обов'язковість етичного та розумного застосування.

РОЗДІЛ 4. РЕАЛІЗАЦІЯ

При переході від теоретичних основ і широкого застосування великих мовних моделей (LLM) до конкретної сфери їхнього практичного розгортання, ця глава служить істотним переходом до сфери реалізації. Ця глава є наріжним каменем поточного дослідження, перетворюючи абстрактні концепції на відчутну реальність. Саме тут буде досліджено тонкощі інтеграції LLM в існуючі системи, розгадуючи виклики, стратегії та виконання в реальному світі, які лежать в основі функціонування цих складних інструментів.

Ландшафт реалізації різноманітний і багатогранний. Він охоплює спектр, який варіюється від початкових етапів вбудовування LLM у робочі процеси компанії до масштабування їхніх можливостей у різних областях і, далі, до підтримки їхньої ефективності серед даних і конфіденційності, що розвиваються. Кожен розділ цієї глави присвячений конкретному аспекту процесу впровадження, розбираючи дрібниці, які сприяють успішній інтеграції LLM.

Буде розпочато з оцінки готовності організації до впровадження LLM, враховуючи технічну інфраструктуру, управління життєвим циклом даних і необхідні навички робочої сили. Після цієї базової оцінки буде заглиблено в стратегії розгортання, які подолають розрив між можливостями LLM і бізнес-цілями. Ці стратегії охоплюють не лише криву впровадження технологій, але й культурні зміни, необхідні організаціям для повного використання переваг магистратури.

Наступні підрозділи розберуть приклади LLM у дії, ілюструючи трансформаційний вплив на продуктивність, інновації та залучення користувачів у секторах технологій та програмного забезпечення. Буде уважно розглянуто проблеми, які виникли під час впровадження — від технічних перешкод до етичних труднощів — і рішення, які були створені у відповідь.

Кафедра КІТ (47)				НАУ 23 28 98 000 ПЗ			
Виконав	Драган В.В.			МОВНІ МОДЕЛІ ДЛЯ АНАЛІЗУ ВЕЛИКИХ ОБ'ЄМІВ ТЕКСТУ	Літера	аркуш	аркушів
Керівник	Климова А.С.					60	31
Консульт.					УС-212М		122
Н. контроль	Райчев І. Е.						

Крім того, буде вивчено постійний процес підтримки, оновлення та вдосконалення систем LLM. Динамічний характер мови та промисловості вимагає відданості безперервному навчанню та адаптації, і цей розділ містить дорожню карту для навігації цими вічними водами.

4.1. Готовність інфраструктури та системні вимоги

Перш ніж занурюватися в практичне застосування великих мовних моделей (LLM), дуже важливо створити базову структуру, яка може підтримувати складність і масштаб таких моделей. Готовність інфраструктури полягає в забезпеченні того, що технічне середовище готове для інтеграції LLM, яка зазвичай вимагає значних обчислювальних ресурсів і ефективних механізмів обробки даних. Однак варто відзначити, що використання служб, таких як API OpenAI, може значно знизити вимоги до локальної системи, оскільки важка робота, пов'язана з моделюванням, перекладається на хмарні сервери.

4.1.1. Оцінка технічної інфраструктури для інтеграції LLM

Оцінка технічної інфраструктури є першим кроком в оцінці того, чи може існуюча система вмістити LLM. Це передбачає перевірку можливостей сервера, стабільності мережі та рішень для зберігання даних. Для систем, у яких очікується внутрішній запуск LLM, для керування обчислювальним навантаженням зазвичай рекомендуються високопродуктивні графічні процесори, надійний розподіл пам'яті та багатоядерні процесори. Однак організації, які вирішують використовувати API OpenAI для доступу до моделі, можуть працювати з мінімальним локальним обладнанням, оскільки обробка виконується віддалено, що вимагає лише стабільного підключення до Інтернету та можливості здійснювати безпечні виклики API. Незважаючи на це, для вирішення допоміжних завдань, таких як початкова обробка даних, обробка запитів та інтеграція вихідних даних моделі в подальші програми, все ще потрібна певна локальна обчислювальна потужність.

4.1.2. Управління життєвим циклом даних для LLM

Управління життєвим циклом даних (Data lifecycle management, DLM) має вирішальне значення при роботі з вхідними та вихідними даними LLM. Ефективна стратегія DLM забезпечує цілісність, безпеку та доступність даних протягом усього шляху всередині організації. При впровадженні LLM компанії повинні встановити процеси збору, зберігання, обробки, спільного використання, архівування та видалення даних. Використання LLM через API OpenAI означає, що дані, які надсилаються до та з API, мають бути зашифровані та оброблятися відповідно до стандартів відповідності. Крім того, якість і формат даних можуть суттєво вплинути на продуктивність LLM, що вимагає надійних систем для очищення та форматування даних перед подачею їх у модель.

4.1.3. Оцінка набору навичок і навчання команди

Успішна інтеграція LLM залежить не лише від технології; це також залежить від навичок і готовності залученої команди. Слід провести оцінку набору навичок, щоб визначити, чи має поточна команда необхідні знання в таких сферах, як машинне навчання, інтеграція API, кібербезпека та наука про дані. Якщо виявлені прогалини, слід запровадити навчальні програми, щоб прискорити роботу членів команди. OpenAI надає документацію та ресурси, які можуть допомогти в цьому навчальному процесі. Крім того, команди повинні знати етичні міркування та галузеві правила, що стосуються використання створеного штучним інтелектом контенту та конфіденційності даних.

Дотримуючись цих трьох основ готовності інфраструктури та системних вимог, організації можуть підготуватися до ефективного використання можливостей LLM. Незалежно від того, використовується обчислювальна потужність локальних систем чи хмарний API OpenAI, фундаментальна робота з оцінки та підготовки інфраструктури є ключовим вкладенням у майбутнє технологічних можливостей організації.

4.2. Стратегії інтеграції LLM

Інтеграція великих мовних моделей (LLM) в існуючі бізнес-процеси та робочі процеси — це складна робота, яка потребує ретельного планування та стратегічного передбачення. Це не просто технологічна модернізація, а трансформаційний крок, який може змінити операційний дух організації. У цьому розділі розглядаються стратегічні міркування, які скеровують успішну інтеграцію LLM у бізнес-процеси, від виявлення ефективних випадків використання до проектування робочих процесів і управління наступними культурними змінами.

4.2.1. Визначення випадків використання та встановлення цілей

Перед початком технічної роботи важливо узгодити можливості LLM з бізнес-цілями. Визначення правильних варіантів використання для інтеграції LLM починається з поглибленого аналізу бізнес-процесів, проблемних точок і областей, де обробка природної мови може принести користь. Після визначення потенційних додатків критично важливим стає встановлення чітких і вимірюваних цілей для кожного випадку використання. Цілі можуть включати підвищення рівня задоволеності клієнтів, оптимізацію внутрішньої комунікації або вдосконалення процесів прийняття рішень за допомогою прогнозної аналітики.

Компанії повинні оцінити здійсненність, вплив і рентабельність інвестицій кожного потенційного випадку використання. Наприклад, відділ обслуговування клієнтів може використовувати LLM, щоб скоротити час відповіді та підвищити релевантність відповідей, прагнучи покращити конкретні KPI, як-от оцінки задоволеності клієнтів або середній час обробки. Успіх інтеграції залежить не лише від самої технології, а й від того, наскільки точно вона відповідає стратегічним цілям компанії.

4.2.2. Розробка робочого процесу для прийняття LLM

Після встановлення цілей наступним кроком є розробка робочих процесів, які вбудовують LLM в організаційну структуру. Це передбачає переосмислення та часто реструктуризацію поточних робочих процесів, щоб пристосуватись до нових можливостей, які приносять LLM. Інтеграція має бути безперебійною, покращуючи робочий процес, не спричиняючи непотрібної складності.

Наприклад, LLM, інтегрована в систему керування електронною поштою для підтримки клієнтів, має бути розроблена з робочим процесом, який дозволяє сортувати вхідні електронні листи, генерувати чернетки відповідей і вивчати відгуки, надані супервайзерами. Це вимагає визначення тригерів для втручання LLM, логіки шляхів прийняття рішень і точок взаємодії між людиною та LLM.

Розробка робочого процесу також повинна враховувати вимоги до інфраструктури, такі як необхідні API для інтеграції LLM, конвеєри даних для навчання та подачі моделі, а також налаштування інтерфейсу користувача, необхідні для оптимальної взаємодії з користувачем. Створення прототипів і тестування цих робочих процесів є критично важливими кроками, щоб переконатися, що вони відповідають поставленим цілям і без проблем вписуються в повсякденну діяльність користувачів.

4.2.3. Управління змінами та культурна трансформація

Успішне впровадження LLM виходить за рамки технології; воно охоплює людей всередині організації. Управління змінами має ключове значення для підготовки робочої сили до нових інструментів і процесів. Він починається з чіткого інформування про переваги та зміни, які принесе інтеграція LLM, після чого слідує комплексні навчальні програми, щоб переконатися, що співробітники готові працювати разом із передовими моделями ШІ.

Ця культурна трансформація часто вимагає переосмислення робочих ролей, впровадження нових наборів навичок і сприяння середовищу постійного навчання

та інновацій. Наприклад, представникам служби підтримки клієнтів потрібно буде перейти від набору відповідей до кураторства та контролю за контентом, створеним ШІ.

Лідерство відіграє вирішальну роль у цій трансформації, підтримуючи впровадження LLM і сприяючи зміні мислення від побоювання до ШІ до розуміння його ролі як партнера, що додає цінність. Ця культурна зміна життєво важлива для реалізації повного потенціалу LLM і вимагає постійних зусиль, ресурсів і уваги з боку вищого керівництва до операційних команд.

Кожен із цих підрозділів представляє рівень стратегічної структури, необхідної для ефективної інтеграції LLM. У сукупності вони формують дорожню карту для організацій, які прагнуть використати трансформаційну силу мовних моделей для підвищення цінності бізнесу, операційної ефективності та інновацій.

4.3. Робота з OpenAI API

Прагнучи використовувати можливості великих мовних моделей (LLM), OpenAI надає надійний API, який дозволяє розробникам взаємодіяти з такими моделями, як GPT (Generative Pre-trained Transformer) для різноманітних завдань. У цьому розділі розділу про впровадження буде досліджено про процес налаштування та безпечної взаємодії з OpenAI API, забезпечуючи плавну інтеграцію.

4.3.1. Налаштування середовища OpenAI API

Перш ніж занурюватися в код, важливо налаштувати належне середовище розробки, яке буде взаємодіяти з OpenAI API. Це передбачає отримання ключів API, встановлення необхідних бібліотек і налаштування простору розробки.

Спочатку необхідно отримати ключ API з платформи OpenAI. Цей унікальний ідентифікатор буде шлюзом для запитів до служб OpenAI.

Маючи під рукою ключ API, можемо продовжити налаштування свого середовища (Рис. 4.1):

```
// Install the OpenAI Node.js package
const { Configuration, OpenAIApi } = require("openai");

const configuration = new Configuration({
  apiKey: process.env.OPENAI_API_KEY,
});
const openai = new OpenAIApi(configuration);
```

Рис. 4.1. Налаштування API

Необхідно зберігати ключ API у змінній середовища, а не жорстко кодувати його у своїй програмі. Це фундаментальна практика безпеки для захисту облікових даних.

4.3.2. Найкращі методи автентифікації та безпеки

Під час роботи з API, особливо тими, які надають доступ до потужних моделей машинного навчання, безпека має бути на першому місці у стратегії впровадження.

Автентифікація в API OpenAI здійснюється за допомогою ключа API, який було отримано раніше (Рис. 4.2):

```
// Setup environment variable in your shell or environment configuration
process.env.OPENAI_API_KEY = 'your-api-key';
```

Рис. 4.2. Аутентифікація за допомогою API ключа

Завжди необхідно використовувати змінні середовища та ніколи не закріплювати ключі API в системах контролю версій, таких як Git.

Крім того, треба дотримуватися цих найкращих практик безпеки:

Обмежити доступ до ключа API: лише необхідні частини програми повинні мати доступ до вашого ключа OpenAI API.

Використовуйте HTTPS: переконайтеся, що всі запити до OpenAI API надходять через HTTPS, щоб запобігти перехопленню.

Сумлінно обробляйте помилки: переконайтеся, що будь-які помилки під час зв'язку з API не розкривають конфіденційну інформацію.

4.3.3. Обробка запитів і відповідей API

Взаємодія з API OpenAI передбачає створення HTTP-запитів і обробку відповідей. Пакет Node.js, наданий OpenAI, спрощує цей процес. Приклад створення запиту на створення тексту (Рис. 4.3):

```
async function generateText(promptText) {
  try {
    const response = await openai.createCompletion({
      model: "text-davinci-003", // Specify the model
      prompt: promptText, // Your prompt
      max_tokens: 150, // Maximum number of tokens to generate
    });

    console.log(response.data.choices[0].text); // Output the response
    return response.data.choices[0].text;
  } catch (error) {
    console.error(error.response ? error.response.data : error.message);
  }
}

// Use the function
generateText("Translate the following English text to French: 'Hello, how a
```

Рис. 4.3. Взаємодія з OpenAI API

У функції `generateText` ми створюємо запит на завершення для вказаної моделі (наприклад, "text-davinci-003"). Функція отримує підказку, надсилає запит і записує згенерований текст. Обробка помилок має вирішальне значення; будь-які проблеми

в процесі надсилання запитів будуть виявлені та зареєстровані, забезпечуючи надійність і надійність вашої програми.

Завдяки цьому керованому налаштуванню розробники можуть ініціювати безпечно та ефективно спілкування з API OpenAI, використовуючи можливості LLM для вдосконалення своїх програм. Маючи фундаментальні знання про конфігурацію середовища, найкращі практики автентифікації та обробку запитів, ви тепер готові досліджувати широкі можливості, які надають моделі OpenAI.

4.4. Розробка бекенда за допомогою NestJS

У сфері розробки на стороні сервера Node.js зарекомендував себе як надійна платформа для створення масштабованих мережових програм. NestJS, платформа для створення ефективних, надійних і масштабованих програм на стороні сервера, набула популярності завдяки простоті використання та інтеграції найкращих концепцій об'єктно-орієнтованого програмування, функціонального програмування та функціонального реактивного програмування.

4.4.1. Основи NestJS і налаштування проекту

Перш ніж занурюватися в код, важливо розуміти, що NestJS використовує TypeScript за замовчуванням, привносячи статичний тип у Node.js. Він також використовує декоратори для легкої анотації та логічного розподілу проблем, що робить кодову базу чистою та зручною для обслуговування.

Щоб почати новий проект, ми використовуємо Nest CLI (Рис. 4.4):

```
npm i -g @nestjs/cli  
nest new project-name
```

Рис. 4.4. Створення нового проекту

Це встановлює новий проект NestJS зі структурою за замовчуванням. можемо перейти до каталогу проекту та запустити сервер за допомогою (Рис. 4.5):

```
cd project-name
npm run start
```

Рис. 4.5. Запуск проекту

4.4.2. Створення RESTful API за допомогою NestJS

Створення кінцевої точки RESTful API у NestJS є простим. Розглянемо простий контролер, який обробляє вибірку даних користувача (Рис. 4.6):

```
import { Controller, Get, Param } from '@nestjs/common';

@Controller('users')
export class UsersController {
  @Get(':id')
  findOne(@Param('id') id: string): string {
    // Placeholder for service call to fetch user by id
    return `This action returns a user with id ${id}`;
  }
}
```

Рис. 4.6. Створення кінцевої точки RESTful API

У наведеному вище фрагменті коду `@Controller('users')` прикрашає (decorates) клас `UsersController`, зіставляючи його з маршрутом `'users'`. Декоратор `@Get(':id')` створює маршрут для запитів HTTP GET, де `'id'` є параметром маршруту.

4.4.3. Інтеграція OpenAI API у NestJS Framework

Щоб інтегрувати API OpenAI, спочатку потрібно встановити необхідну клієнтську бібліотеку (Рис. 4.7):

```
npm install openai
```

Рис. 4.7. Інтеграція OpenAI API

Після цього можемо налаштувати службу в NestJS, яка використовує OpenAI API (Рис. 4.8):

```
import { Injectable } from '@nestjs/common';
import { Configuration, OpenAIApi } from "openai";

@Injectable()
export class OpenAIService {
  private openai;

  constructor() {
    const configuration = new Configuration({
      apiKey: process.env.OPENAI_API_KEY,
    });
    this.openai = new OpenAIApi(configuration);
  }

  async generateResponse(prompt: string) {
    const response = await this.openai.createCompletion("text-davinci-003",
      prompt,
      max_tokens: 150,
    );
    return response.data.choices[0].text.trim();
  }
}
```

Рис. 4.8. Налаштування служби NestJS

У цьому сервісі ми налаштуємо клієнт OpenAI за допомогою ключа API та створюємо метод для генерації текстової відповіді на запит.

4.4.4. Обробка та журналювання помилок

Правильна обробка помилок і ведення журналу мають вирішальне значення для налагодження та підтримки якості програми. NestJS надає готові фільтри винятків і службу журналювання.

Маємо фільтр глобальних винятків (Рис. 4.9):

```
import { ExceptionFilter, Catch, ArgumentsHost, HttpException, Logger } from '@nestjs/core';

@Catch(HttpException)
export class HttpErrorFilter implements ExceptionFilter {
  catch(exception: HttpException, host: ArgumentsHost) {
    const ctx = host.switchToHttp();
    const request = ctx.getRequest();
    const response = ctx.getResponse();
    const status = exception.getStatus();

    const errorResponse = {
      statusCode: status,
      timestamp: new Date().toISOString(),
      path: request.url,
      method: request.method,
      message: exception.getResponse()['message'],
    };

    Logger.error(
      `${request.method} ${request.url}`,
      JSON.stringify(errorResponse),
      'ExceptionFilter',
    );

    response.status(status).json(errorResponse);
  }
}
```

Рис. 4.9. Фільтр глобальних винятків

І налаштуємо цей фільтр глобально у файлі main.ts (Рис. 4.10):

```
import { NestFactory } from '@nestjs/core';
import { AppModule } from './app.module';
import { HttpErrorFilter } from './shared/http-error.filter';

async function bootstrap() {
  const app = await NestFactory.create(AppModule);
  app.useGlobalFilters(new HttpErrorFilter());
  await app.listen(3000);
}
bootstrap();
```

Рис. 4.10. Налаштування глобального фільтру.

Рядок `Logger.error` у фільтрі використовує вбудований реєстратор NestJS для реєстрації помилок. Цей фільтр фіксує всі винятки HTTP та форматує стандартну відповідь про помилку.

У цих підрозділах було окреслено базові кроки для налаштування проекту NestJS, створення RESTful API, інтеграції з OpenAI API та забезпечення надійної обробки помилок і журналювання, надаючи план для розробки серверної частини за допомогою NestJS.

4.5. Розробка зовнішнього інтерфейсу за допомогою JavaScript

У контексті впровадження великих мовних моделей (LLM) розробка інтерфейсу відіграє вирішальну роль у формуванні взаємодії з користувачем. Використання JavaScript як обраної мови для інтерфейсу дозволяє створювати динамічні та адаптивні інтерфейси, які можуть бездоганно взаємодіяти з серверними службами. Цей розділ має на меті окреслити найкращі методи розробки інтерфейсів, які використовують функції LLM, ефективно керують асинхронним зв'язком із серверною частиною та розглядають питання безпеки, важливі для захисту даних користувача та взаємодії.

4.5.1. Розробка зручних інтерфейсів для функцій LLM

Щоб створити привабливий та інтуїтивно зрозумілий інтерфейс користувача, який використовує потужність LLM, розробники повинні зосередитися на чіткості, оперативності та зворотному зв'язку.

```
// Example: Input field for text analysis with LLM
const userInput = document.getElementById('user-input');
const submitButton = document.getElementById('submit-button');
const resultDisplay = document.getElementById('result-display');

submitButton.addEventListener('click', () => {
  const textToAnalyze = userInput.value;
  if (textToAnalyze) {
    // Display a loading message or animation
    resultDisplay.textContent = 'Analyzing text...';
    // Function to handle sending text to the backend will go here
  }
});
```

Рис. 4.11. Механізм введення інформації користувачем.

Цей фрагмент (Рис. 4.11) ілюструє основну структуру механізму введення користувача. Важливо надати негайний зворотний зв'язок, наприклад стан завантаження, щоб повідомити користувача, що його запит обробляється.

4.5.2. Асинхронний зв'язок із серверною частиною

Управління асинхронними операціями є основоположним у розробці інтерфейсу, особливо коли мова йде про затримку мережевих запитів до серверних служб, таких як LLM API.

```

// Example: Function to send text to the backend and receive analysis result
async function analyzeText(text) {
  try {
    const response = await fetch('/api/analyze', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({ text }),
    });
    if (!response.ok) {
      throw new Error(`HTTP error! status: ${response.status}`);
    }
    const result = await response.json();
    // Display results
    resultDisplay.textContent = result.analysis;
  } catch (error) {
    // Handle errors, such as displaying an alert to the user
    console.error('There was a problem with the fetch operation:', error);
  }
}

```

Рис. 4.12. Надсилання тексту серверній частині

Функція `analyzeText` (Рис. 4.12) інкапсулює логіку для надсилання тексту до серверної частини та оновлення інтерфейсу користувача результатами, обробляючи стани успіху та помилки.

4.5.3. Міркування безпеки зовнішнього інтерфейсу

Безпека на інтерфейсі включає перевірку введених даних, кодування даних для запобігання атакам XSS і належне керування секретами.

```
// Example: Encoding user input to prevent XSS when displaying results
function encodeHTML(str){
    return str.replace(/[\&<>'"]/g,
        tag => ({'&': '&amp;', '<': '&lt;', '>': '&gt;', '"': '&quot;', "'":
    }
}

// Safe display of results
resultDisplay.innerHTML = encodeHTML(result.analysis);
```

Рис. 4.13. Кодування спец символів.

Ця функція (Рис. 4.13) забезпечує кодування будь-яких спеціальних символів HTML перед відображенням, запобігаючи впровадженню шкідливих сценаріїв. Крім того, розробники повинні переконатися, що ключі API та конфіденційні дані не розкриваються в кодї інтерфейсу. Такі секрети мають зберігатися у змінних середовища на стороні сервера, а доступ до них здійснюється через захищені серверні кінцеві точки.

Завдяки ретельному дизайну, ефективній асинхронній комунікації та сильному акценту на безпеці інтерфейс може служити потужним каналом для функцій LLM, забезпечуючи користувачам безперебійну та безпечну роботу.

4.6. Включення векторної бази даних

Включення векторних баз даних у сферу великих мовних моделей (LLM) означає ключовий крок в управлінні та отриманні даних великого розміру, що є символом завдань обробки природної мови. У цьому розділі розглядаються нюанси векторних баз даних, окреслюються критерії вибору відповідної векторної бази даних і пояснюється процес інтеграції таких баз даних у структуру NestJS.

4.6.1. Вступ до векторних баз даних та їх значення

Поява векторних баз даних стала відповіддю на унікальні виклики, пов'язані з сучасними структурами даних, особливо тими, що створені LLM. На відміну від традиційних баз даних, які розроблені для скалярних значень (числових або текстових даних у дискретних одиницях), векторні бази даних створені для ефективної обробки та запитів до даних великого векторного простору.

Ключові переваги:

Продуктивність: векторні бази даних чудово підходять для пошуку та отримання даних у просторі, де зв'язки між об'єктами визначаються метрикою відстані, наприклад косинусною подібністю, а не точними збігами.

Релевантність: вони підтримують пошук приблизного найближчого сусіда (ANN), який є критично важливим у таких програмах, як семантичний пошук, системи рекомендацій і кластеризація даних.

Масштабованість: розроблені для обробки великих обсягів даних без значного погіршення продуктивності, векторні бази даних ідеально підходять для величезних наборів даних, створених LLM.

Актуальність векторних баз даних у контексті LLM є безперечною. Вони незамінні для таких завдань, як пошук подібного текстового вмісту через вбудовані файли, кластеризація документів на основі семантичної подібності або навіть забезпечення серверної частини для складних систем автозаповнення, які розуміють контекст, окрім простого зіставлення рядків.

4.6.2. Вибір правильної векторної бази даних

Коли справа доходить до вибору векторної бази даних, необхідно взяти до уваги кілька міркувань, щоб забезпечити відповідність вимогам проекту:

Фактори, які слід враховувати:

Затримка запиту: час, потрібний для виконання пошукового запиту, може бути критичним, особливо для призначених для користувача програм, які потребують відповідей у реальному часі.

Масштабованість: важливо оцінити, наскільки добре база даних масштабується зі збільшенням обсягу даних і навантаження на запити.

Сумісність даних: переконайтеся, що база даних може обробляти конкретні типи даних і векторні розміри, створені вашим LLM.

Операційна складність: оцініть простоту налаштування, обслуговування та інтеграції в існуючі системи.

Спільнота та підтримка: сильна спільнота та професійна підтримка можуть бути життєво важливими для довгострокового успіху.

Ретельно оцінюючи ці фактори, організації можуть визначити найбільш підходящу векторну базу даних, яка не тільки відповідає безпосереднім потребам їхніх додатків LLM, але й пристосовується до майбутнього зростання та складності.

4.6.3. Інтеграція векторної бази даних із NestJS

Інтеграція векторної бази даних із NestJS, прогресивним фреймворком Node.js, потребує систематичного підходу, який забезпечує безперебійну взаємодію між системою керування багатовимірними даними та логікою програми на стороні сервера.

Кроки для інтеграції:

Початкове налаштування: інсталювати необхідні пакети векторної бази даних або клієнти, надані для середовищ Node.js.

Конфігурація: налаштувати параметри підключення в NestJS, щоб встановити безпечне та ефективне посилання на векторну базу даних.

Створення служби: розробити службу в NestJS, яка абстрагує операції з базою даних, надаючи чистий інтерфейс для вставки даних, запитів і керування (Рис. 4.14).

```

@Module({})
export class PineconeModule {
  static forRootAsync({
    inject,
    useFactory,
  }: ForRootAsyncModuleOptions<PineconeModuleOptions>): DynamicModule {
    return {
      module: PineconeModule,
      global: true,
      providers: [
        {
          provide: PineconeClient,
          inject,
          useFactory: async (...args) => {
            const options = await useFactory(...args);

            const client = new PineconeClient();

            await client.init({
              apiKey: options.apiKey,
              environment: options.environment,
            });

            return client;
          },
        },
      ],
      exports: [PineconeClient],
    };
  }

  static forRoot(options: PineconeModuleOptions) {
    return this.forRootAsync({ useFactory: () => options });
  }
}

```

Рис. 4.14. Служба взаємодії з векторною базою даних.

Обробка запитів: реалізувати логіку в службі для обробки векторних запитів, використовуючи можливості бази даних (Рис. 4.15).

```

public async query(prompt: string, chunksToIncludeNumber: number) {
  const [promptEmbedding] = await this.createEmbeddings([prompt]);
  const queryResponse = await this.getPineconeIndex().query({
    queryRequest: {
      topK: chunksToIncludeNumber,
      includeMetadata: true,
      vector: promptEmbedding.embedding,
      namespace: PINECONE_INDEX_NAME,
    },
  });

  if (queryResponse.matches.length === 0) {
    throw new Error('The index is empty.');
```

Рис. 4.15 Логіка взаємодії з векторною базою даних.

Кінцеві точки API: відкрити кінцеві точки API RESTful, які дозволяють інтерфейсній програмі взаємодіяти з векторною базою даних через серверну частину NestJS.

Дотримуючись цих кроків, розробники можуть створювати надійні програми NestJS, які використовують унікальні можливості векторних баз даних, таким чином забезпечуючи складні функції пошуку даних, необхідні для програм на базі LLM.

Загалом, інтеграція векторних баз даних є фундаментальним компонентом у використанні повного потенціалу LLM у програмі. У цьому розділі закладено основу для розуміння, вибору та впровадження векторних баз даних спільно з NestJS, створюючи схему, на якій розробники можуть спиратися під час розробки наступного покоління інтелектуальних програм.

4.7. Повна інтеграція стека

У ландшафті сучасної веб-розробки повна інтеграція стека формує основу, на якій будуються надійні програми. Ця інтеграція є синтезом зовнішніх і внутрішніх

систем, організованих для бездоганної роботи як єдиного цілого. Саме тут зустрічаються користувальницький інтерфейс і серверна логіка, гарантуючи, що користувачам надається оперативний, надійний та інтуїтивно зрозумілий досвід.

У цьому розділі розглядаються основні практики та методології для досягнення гармонійної повної інтеграції стека, з особливим акцентом на ролі великих мовних моделей (LLM) у цій структурі. Розглядаються прагматичні аспекти створення плавного каналу зв'язку між передньою частиною, яка стикається з користувачем, і задньою частиною обробки даних, підтримуючи цілісність даних протягом усього робочого процесу та дотримуючись суворого режиму тестування та гарантії якості.

4.7.1. Поєднання між інтерфейсом і сервером: найкращі методи

Створення симбіотичного зв'язку між інтерфейсом і бекендом вимагає ретельного підходу до проектування та впровадження. Ось деякі з найкращих практик:

Визначення контракту API: Встановити чіткий контракт API, використовуючи такі стандарти, як OpenAPI/Swagger. Це гарантує, що і зовнішні, і бекенд-розробники мають спільне розуміння кінцевих точок API, структур даних і очікуваної поведінки.

Управління станом. Використовувати надійні стратегії керування станом на інтерфейсі, щоб керувати динамікою взаємодії користувачів і змінами даних, забезпечуючи безперебійну роботу користувача.

Архітектура мікросервісів: розглянути можливість прийняття архітектури мікросервісів, щоб відокремити компоненти, зробити систему більш масштабованою, зручною для обслуговування та дозволити командам інтерфейсу та серверної частини працювати незалежно та ефективно.

Автентифікація та авторизація: реалізувати безпечні механізми автентифікації та авторизації, щоб забезпечити безпечний обмін даними між інтерфейсом і сервером, як правило, за допомогою маркерів, таких як JWT.

Безперервна інтеграція/безперервне розгортання (CI/CD): налаштувати конвеєр CI/CD для автоматизації тестування та розгортання, що допомагає завчасно виявляти проблеми інтеграції та оптимізувати процес доставки.

4.7.2. Забезпечення потоку даних і цілісності між компонентами

Цілісність даних, які проходять через рівні системи, має першочергове значення. Щоб зберегти цю цілісність:

Перевірка та санітарна обробка: виконати перевірку та санітарну обробку даних на обох кінцях системи. Інтерфейс має перевіряти введені користувачем дані, щоб забезпечити негайний зворотний зв'язок, тоді як сервер має дезінфікувати дані, щоб запобігти ін'єкції SQL та іншим зловмисним діям.

Синхронізація даних: реалізувати механізми для синхронізації даних між клієнтом і сервером, такі як WebSockets або Server-Sent Events (SSE), щоб полегшити оновлення даних у реальному часі.

Обробка помилок: розробити надійні процедури обробки помилок, щоб фіксувати винятки чи аномалії в потоці даних і керувати ними, надаючи чіткі повідомлення для зовнішнього інтерфейсу, одночасно реєструючи проблеми для внутрішнього дослідження.

Управління транзакціями: якщо це можливо, використовувати транзакції, щоб забезпечити повне завершення внутрішніх операцій або відкат у разі помилки, зберігаючи узгодженість даних.

Стратегії кешування: використовувати кешування, де це необхідно, щоб зменшити навантаження на базу даних і підвищити продуктивність, гарантуючи, що стратегії анулювання кешу запобігають представленню застарілих даних.

4.7.3. Тестування та забезпечення якості

Тестування та гарантія якості мають вирішальне значення для забезпечення належного функціонування повної інтеграції стека. Це охоплює:

Модульне тестування: і фронтенд, і бекенд повинні мати набір модульних тестів для перевірки функціональності окремих компонентів або функцій.

Тестування інтеграції: Провести тести інтеграції, щоб перевірити взаємодію між різними компонентами, включаючи зовнішні служби, такі як OpenAI API.

Наскрізне тестування: запровадити наскрізні тестові фреймворки, такі як Selenium або Cypress, щоб імітувати сценарії реального користувача та перевірити потік програми від початку до кінця.

Тестування продуктивності: регулярно перевіряти продуктивність програми, щоб виявити вузькі місця або погіршення, які можуть вплинути на взаємодію з користувачем.

Аудити безпеки: Виконувати перевірки безпеки та сканування вразливостей, щоб виявити та пом'якшити потенційні загрози безпеці.

Використовуючи ці стратегії та підкреслюючи важливість ретельного процесу повної інтеграції стека, розробники можуть створити надійну програму, яка максимально використовує можливості LLM, забезпечуючи тим самим ефективний, безпечний і зручний продукт.

4.8. Розгортання та масштабування

Розгортання та масштабування великих мовних моделей (LLM) в інфраструктурі програми є ключовими етапами, на яких розроблене рішення перетворюється на надійний продукт, призначений для користувачів. У цьому розділі розглядаються основні стратегії, методи моніторингу та методи масштабування, необхідні для того, щоб програми LLM не тільки розгорталися ефективно, але й здатні підтримувати продуктивність і надійність у масштабі.

4.8.1. Стратегії розгортання програм LLM

Розгортання програм LLM передбачає більше, ніж просто перенесення програмного забезпечення із середовища розробки на робочі сервери. Це критично

важливий процес, який вимагає ретельного планування та виконання для забезпечення надійності, доступності та продуктивності програми.

Хмара проти локального серверу: рішення про розгортання в хмарі чи локальному сервері є фундаментальним. Хмарні розгортання можуть запропонувати масштабованість і скоротити технічне обслуговування, але можуть створити проблеми щодо суверенітету даних і передбачуваності витрат. Локальне розгортання забезпечує контроль і локальність даних за рахунок попередніх інвестицій і проблем з масштабованістю.

Контейнеризація: використання контейнерної технології, такої як Docker, інкапсулює середовище LLM, забезпечуючи узгодженість у середовищах розробки, тестування та виробництва, сприяючи безперервній інтеграції та конвеєрам безперервного розгортання (CI/CD).

Конвеєри CI/CD: створення конвеєрів CI/CD дає змогу автоматизовано тестувати та розгортати, зменшуючи ймовірність людської помилки та прискорюючи процес випуску.

Синьо-зелені розгортання: ця стратегія передбачає запуск двох ідентичних виробничих середовищ. Лише одне обслуговує живий трафік, а інше є зоною зупинки. Це забезпечує плавний відкат у разі проблем і мінімальний час простою під час оновлень.

Випуски Canary: поступове впровадження змін для невеликої підгрупи користувачів перед повномасштабним розгортанням допомагає завчасно виявити потенційні проблеми, не впливаючи на всіх кінцевих користувачів.

4.8.2. Моніторинг і налаштування продуктивності

Після розгортання вкрай важливо мати комплексний моніторинг і механізми налаштування продуктивності, щоб переконатися, що додаток підтримує оптимальний стан.

Моніторинг у реальному часі: впровадження інструментів моніторингу в реальному часі для відстеження продуктивності програми, дій користувачів і

працездатності системи, гарантуючи, що будь-які проблеми можна буде виявити та оперативно вирішити.

Аналітика журналів: збір і аналіз журналів може запропонувати уявлення про поведінку користувачів, помилки програми та проблеми з продуктивністю системи, повідомляючи про необхідні коригування.

Показники продуктивності: Встановлення ключових показників ефективності (KPI), специфічних для програм LLM, таких як час відгуку, пропускна здатність і рівень помилок, є важливим для налаштування продуктивності.

Автоматичне масштабування: використання функцій автоматичного масштабування, наданих хмарними службами або інструментами оркестровки, гарантує, що програма може обробляти змінні навантаження шляхом автоматичного налаштування обчислювальних ресурсів.

4.8.3. Масштабування програм LLM для високої доступності та навантаження

Масштабування — це сприяння зростанню без шкоди для взаємодії з користувачем і без непомірних витрат.

Горизонтальне та вертикальне масштабування: горизонтальне масштабування (додавання додаткових машин) часто є більш гнучким і надійним, ніж вертикальне масштабування (додавання більшої потужності існуючим машинам), особливо для додатків LLM, у яких можуть виникнути непередбачувані стрибки використання.

Балансування навантаження: розподіл робочого навантаження між декількома серверами, щоб гарантувати, що жоден сервер не несе надмірного навантаження, запобігаючи вузьким місцям продуктивності.

Масштабування бази даних: використання таких методів, як шардинг або використання баз даних NoSQL, щоб гарантувати, що рівень даних може обробляти збільшені навантаження та складні запити, не перетворюючись на вузьке місце.

Стратегії кешування: Реалізація кешування для тимчасового зберігання копій часто використовуваних точок даних або результатів обчислень, що значно зменшує необхідність повторного обчислення або отримання даних із бази даних для кожного запиту користувача.

Завдяки стратегічному застосуванню цих методологій розгортання та масштабування програми LLM можуть повністю розкрити свій потенціал, надаючи широкій аудиторії складні можливості обробки мови з надійністю та продуктивністю, очікуваними від сучасних програмних рішень.

4.9. Технічне обслуговування та постійне вдосконалення

У життєвому циклі програмної системи фаза після розгортання не є ні пасивною, ні статичною. Щоб забезпечити довговічність і релевантність великих мовних моделей (LLM) у робочому процесі організації, постійне обслуговування та безперервне вдосконалення є критично важливими. У цьому розділі пояснюється стратегія та методологія підтримки та підвищення продуктивності LLM після розгортання. Він охоплює створення надійних циклів зворотного зв'язку, вирішення явища дрейфу моделі та створення основи для довгострокової підтримки та оновлень. Буде досліджено, як ці елементи формують згуртований підхід до підтримки життєздатності магистратури в динамічному бізнесі та технологічному середовищі.

4.9.1. Впровадження циклів зворотного зв'язку для безперервного навчання

Корисність LLM глибоко переплітається з їх здатністю навчатися та адаптуватися. Цикли зворотного зв'язку необхідні для безперервного навчання, дозволяючи моделям розвиватися та залишатися налаштованими на нові дані та взаємодії з користувачем.

Інтеграція відгуків користувачів: розглядаються методи фіксації відгуків користувачів, як явних, так і неявних, для інформування про коригування моделі.

Механізми повторного введення даних: дизайн систем, які повторно вводять результати реальних даних у модель для вдосконалення її прогнозів і розуміння природної мови.

Конвеєри безперервного навчання: технічна реалізація систем безперервного навчання, що гарантує, що моделі отримують переваги від свіжих даних без погіршення продуктивності під час оновлення.

4.9.2. Оновлення моделей і робота з зміщенням

Зміщення моделі відбувається, коли продуктивність LLM з часом погіршується через зміни в базовому розподілі даних або у зовнішньому середовищі. Важливо підтримувати актуальність моделі.

Ідентифікація зміщення: методи моніторингу та ідентифікації, включаючи показники продуктивності та виявлення аномалій.

Стратегії перенавчання моделей: необхідно ефективно перенавчати моделі, врівноважуючи потребу в сучасній точності з обмеженнями обчислювальних ресурсів.

Контроль версій і процедури відкату: у центрі уваги є підтримка версії моделі, забезпечення плавного переходу між оновленнями моделі та наявність стратегій відкату для стабільності.

4.9.3. Довгострокова підтримка та оновлення екосистеми LLM

Забезпечення того, щоб LLM залишалися ефективними з часом, вимагає екосистемного підходу, коли всі компоненти впровадження переглядаються та оновлюються за необхідності.

Оновлення інфраструктури та залежностей: важливою є підтримка актуальності допоміжної інфраструктури та залежностей за допомогою виправлень безпеки та покращень продуктивності.

Адаптація до розвитку мови: Мова змінна, і LLM повинні адаптуватися до змін у народній мові та вживанні.

Розрахованість на майбутнє та масштабованість: необхідно побудувати стратегії для перспективних реалізацій LLM, гарантуючи, що вони можуть масштабуватися та адаптуватися до майбутніх технологічних досягнень.

4.10. Тематичні дослідження та реальні приклади

У цьому розділі ми заглибимося в тематичні дослідження, виклики та майбутні перспективи, пов'язані з впровадженням великих мовних моделей (LLM) для аналізу тексту.

4.10.1. Аналіз успішних реалізацій LLM

Універсальність LLM призвела до їх розгортання в різних секторах. Ось кілька помітних реалізацій:

Фінансовий сектор: провідна інвестиційна компанія використала LLM для аналізу ринкових тенденцій і прогнозування цін на акції, перевершивши точність експертів і досягнувши значних фінансових прибутків.

Журналістика: велика новинна організація інтегрувала LLM у свою редакцію, щоб допомогти у створенні контенту, покращуючи результати без шкоди для якості.

Індустрія розваг: кінокомпанія використовувала LLM для створення ідей сценарію, значно скорочуючи час мозкового штурму та покращуючи процес оповідання.

Аналіз медичного тексту: LLM використовуються для діагностики захворювань і сортування випадків у відділеннях невідкладної допомоги шляхом аналізу медичних текстів, що демонструє їхній потенціал у сфері охорони здоров'я.

Аналіз юридичного тексту: в емпіричних юридичних дослідженнях LLM полегшує тематичний аналіз, демонструючи новий підхід до якісного аналізу в юридичних сферах.

4.10.2. Висновки, отримані зі складних розгортань

Розгортання LLM не позбавлене проблем, деякі з яких включають:

Безпека моделі та неправильне використання: Розуміння та вирішення проблем безпеки та неправильного використання розгорнутих моделей є надзвичайно важливими, як підкреслює OpenAI.

Інженерні витрати: розгортання може потребувати складних інженерних зусиль, особливо в безпечних середовищах.

Доступність інфраструктури: доступність необхідної інфраструктури, як-от графічних процесорів, може блокувати поряд із високими витратами на логічні висновки зі збільшенням масштабу.

Виявлення викликів: Швидкий розвиток магістрів LLM ускладнює визначення проблем, що залишилися, і плідних сфер застосування.

Найкращі практики розгортання: Створення чітких інструкцій, окреслення бажаного результату та використання короткочасних підказок є одними з найкращих практик для успішного розгортання LLM.

4.10.3. Майбутні тенденції та нові практики впровадження LLM

Майбутнє великих мовних моделей (LLM) багатообіцяюче завдяки постійному розвитку технологій і зростаючому інтересу з боку різних секторів. Ось деякі з майбутніх тенденцій і нових практик впровадження LLM:

Подальші досягнення в LLM:

Випуск OpenAI GPT-4 знаменує собою важливу віху завдяки його мультимодальним можливостям, що обробляють як текст, так і зображення, демонструючи тенденцію до більш складних і універсальних LLM.

Підвищений корпоративний інтерес:

Успіх таких моделей, як ChatGPT, викликав значний корпоративний інтерес. Бюджетні асигнування на технологію LLM досягають мільйонів, що вказує на майбутнє, де LLM є невід'ємною частиною різноманітних організаційних операцій, починаючи від управління знаннями до підтримки клієнтів і маркетингу.

Трансформаційний потенціал у різних галузях:

LLM вважаються трансформаційними, особливо в галузях, які зазнають цифрового прискорення. Наприклад, віце-президент General Motors зазначив, що «ChatGPT буде в усьому», підкресливши широке застосування LLM у найближчому майбутньому.

Розробка нових моделей та інструментів:

Після випуску ChatGPT стався сплеск у розробці та випуску нових LLM та інструментів. Наприклад, GPT-4 тепер перевищує прохідний бал на кожному етапі іспиту з медичної ліцензії США (USMLE), що вказує на потенціал для LLM в оцінці професійної компетентності. Також з'явилися інші моделі та інструменти, такі як PaLM, ChatBot BARD, Llama та Llama-2, які розширюють екосистему LLM. Ці розробки супроводжуються випуском API, що дозволяє легше інтегрувати ці моделі в незалежне програмне забезпечення, розширюючи сферу можливого застосування.

Розширення можливостей:

Наступне покоління LLM від різних компаній, таких як Google, Meta, Nvidia/Microsoft і AI21 Labs, розширює можливості та, можливо, включає візуальне введення та інші мультимодальні функції, які, як очікується, експоненціально розширять список можливих програм.

Плани розгортання виробництва:

Опитування показало, що більше половини науковців та інженерів з обробки даних планують розгорнути додатки LLM у виробництві найближчим часом, що вказує на суттєвий зсув у бік впровадження технології LLM.

Найкращі методи розгортання:

Зі зростанням розгортання LLM зростає важливість встановлення найкращих практик розгортання для забезпечення ефективного та етичного використання цих

моделей. Це включає в себе написання чітких і конкретних інструкцій для підказок, використання роздільників для позначення окремих частин підказки, окреслення структури бажаного результату та використання невеликих підказок для надання успішних прикладів виконання завдання.

Ці тенденції та практики підкреслюють ландшафт LLM, що розвивається, з чіткою траєкторією до більш інтегрованих, універсальних та відповідальних реалізацій.

РОЗДІЛ 5. ВИКЛИКИ ТА ЗАСОБИ ЇХ ПОДОЛАННЯ

Поява великих мовних моделей (LLM) відкрила нову еру в царині текстового аналізу, автоматизації та безлічі інших програм, змінюючи традиційні парадигми та обіцяючи трансформаційні результати. Однак, як і будь-який технологічний стрибок, ця еволюція несе з собою унікальний набір проблем, які необхідно визнати, зрозуміти та вирішити. У цьому розділі розглядаються багатогранні перешкоди, які виникають після впровадження LLM. Окрім простої ідентифікації, також буде досліджено інноваційні стратегії, практики та рішення, які лідери галузі та дослідники використовують для подолання цих перешкод. Вивчаючи як перешкоди, так і шляхи їх вирішення, є прагнення представити збалансовану, цілісну перспективу на шляху інтеграції LLM у різноманітні сфери, наголошуючи на тому, що хоча цей шлях може бути сповнений складнощів, за допомогою продуманих підходів ці виклики можна змінити. у можливості для зростання та вдосконалення.

5.1. Етичні вимоги до генерації тексту

Сфера створення тексту, хоч і зазнала революції завдяки можливостям великих мовних моделей (LLM), опинилася на критичному етапі, коли здатність генерувати інформацію є її найбільшою силою та потенційним етичним мінним полем. Здатність LLM створювати високоякісний, послідовний і контекстуально релевантний текст у великому масштабі не має собі рівних. Однак ця майстерність висуває на перший план важливі етичні міркування, які не можна нехтувати. У цьому підрозділі буде досліджено етичні вимоги до створення тексту за допомогою LLM, підкреслюючи принципи, які повинні керуватися відповідальним використанням та можливість неправильного використання.

Кафедра КІТ (47)				НАУ 23 28 98 000 ПЗ			
Виконав	Драган В.В.			МОВНІ МОДЕЛІ ДЛЯ АНАЛІЗУ ВЕЛИКИХ ОБ'ЄМІВ ТЕКСТУ	Літера	аркуш	аркушів
Керівник	Климова А.С.					91	7
Консульт.					УС-212М		122
Н. контроль	Райчев І. Е.						

5.1.1. Принципи створення етичних текстів

В основі створення етичного тексту лежать основні принципи, яких необхідно дотримуватися:

Точність: контент, створений LLM, повинен відповідати фактичній достовірності, уникаючи поширення дезінформації, яка може мати далекосяжні наслідки.

Справедливість: результати LLM не повинні бути упередженими, забезпечуючи справедливість уявлень і не закріплюючи стереотипів чи упереджених настроїв.

Прозорість: повинна бути чітка інформація про участь LLM у створенні контенту, гарантуючи, що кінцеві користувачі знають, коли вони читають або взаємодіють із машинним контентом.

5.1.2. Можливість неправильного використання

З великою потужністю виникає потенціал для значного зловживання, і LLM не є винятком:

Дезінформація: майстерність LLM у створенні реалістичного контенту може бути використана для створення та поширення дезінформації в великих масштабах.

Плагіат: здатність LLM асимілювати величезні джерела тексту викликає занепокоєння щодо оригінальності створеного контенту та прав інтелектуальної власності.

Емоційна маніпуляція: LLM можуть створювати контент, націлений на емоційні реакції, які можуть бути використані для маніпуляційних або оманливих цілей.

5.1.3. Захист етичної чесності

У відповідь на ці виклики інституціоналізовано декілька заходів:

Політика використання. Суб'єкти, які розгортають LLM, мають встановити суворі вказівки щодо їх використання, зокрема щодо характеру вмісту, який може бути створений, і перевірок, які застосовуються для дотримання цих правил.

Журнали аудиту: Ведення детальних записів вмісту, створеного LLM, включаючи вихідні дані, модифікації та контекст використання, підтримувати підзвітність.

Комітети з питань етики: створення спеціалізованих органів для нагляду та оцінки діяльності LLM, до складу яких входять люди з різним досвідом, забезпечує різноманітність точок зору в оцінці дотримання етичних норм.

5.1.4. Роль нормативно-правової бази

Визнаючи суспільний вплив машинно-генерованого контенту, зростає вимога зовнішнього регулювання:

Законодавчі заходи: Уряди та міжнародні організації вивчають правила, які регулюють цифровий вміст, поширюючись на машинно-генерований текст.

Стандартизація: розробляються загальногалузеві стандарти для етичного використання штучного інтелекту, що сприяє єдиному підходу до етичних проблем.

Громадський контроль: заохочення публічного дискурсу та ретельного вивчення контенту, створеного LLM, забезпечує відображення суспільних цінностей в етичних міркуваннях.

Підсумовуючи, оскільки ми стоїмо на порозі нової ери генерації інформації, забезпечення етичної цілісності створення тексту з підтримкою LLM є першорядним. У цьому підрозділі наголошується на необхідності пильності, відповідальності та колективної підзвітності під час орієнтування в етичних умовах цієї потужної технології.

5.2. Впорядкування упереджень та справедливості

У цьому підрозділі фокус зміщується в бік законодавчих і саморегулюючих зусиль, спрямованих на стримування упереджень і забезпечення справедливості в контексті великих мовних моделей (LLM).

5.2.1. Поширеність упередженості в LLM

Поява LLM викликала занепокоєння щодо поширення суспільних упереджень через ці моделі. Ці упередження часто виникають через навчальні дані, які можуть містити притаманні упередження, присутні в суспільних дискурсах, з яких дані були взяті. Було задокументовано обширне дослідження методів оцінки упередженості та їх зменшення для LLM, спрямоване на формалізацію та розширення понять соціальної упередженості та справедливості в обробці природної мови. Примітно, що упередження в LLM не лише обмежуються генерацією тексту, але також впливають на наступні завдання, де розгортаються ці моделі.

5.2.2. Регуляторні підходи

Із загостренням дискурсу навколо упереджень у LLM зростає і заклик до надійної нормативної бази. Документ «Frontier AI Regulation: Managing Emerging Risks to Public Safety» пропонує потрібну регуляторну структуру: процеси встановлення стандартів, вимоги до реєстрації та звітності та механізми для забезпечення дотримання стандартів безпеки. Це підкреслює необхідність державного втручання разом із галузевим саморегулюванням, щоб збалансувати ризики громадської безпеки та переваги інновацій штучного інтелекту. Інші обговорення зосереджені навколо забезпечення дотримання політики в контексті LLM, з пропозиціями, включаючи режими ліцензування для передових моделей штучного інтелекту, надання правозастосовних повноважень органам нагляду та

сприяння централізованому регулюванню в поєднанні з механізмами безпеки, розробленими спільнотою.

5.2.3. Зменшення упередженості та сприяння справедливості

Докладаються зусилля щоб консолідувати, формалізувати та розширити розуміння соціальної упередженості та справедливості в сфері LLM. Ці ініціативи спрямовані на запровадження необхідності операціоналізувати справедливість для LLM та дослідити існуючі показники справедливості для попередньо навчених мовних моделей. Крім того, дискусії щодо зменшення упередженості підкреслюють важливість проведення оцінки ризиків перед розгортанням, залучення зовнішніх експертів для незалежного контролю та дотримання стандартизованих протоколів для розгортання передових моделей штучного інтелекту на основі оціненого ризику.

5.2.4. Майбутні напрямки

Динамічний ландшафт регулювання штучного інтелекту вимагає постійної еволюції нормативно-правової бази, щоб йти в ногу зі швидким прогресом технології LLM. «Загальна картина регулювання LLM» охоплює різні аспекти, включаючи моделі нагляду за штучним інтелектом, перетин централізованого регулювання та краудсорсингових механізмів безпеки, а також активний інноваційний підхід до регулювання штучного інтелекту в різних юрисдикціях. Метою залишається сприяння створенню середовища, у якому LLM можна розвивати та використовувати відповідально, без шкоди для етичної чесності та справедливості.

Підрозділ пояснює критичну потребу в регулятивних заходах для пом'якшення упереджень і підтримки справедливості під час розгортання LLM. Завдяки поєднанню державного нагляду, галузевого саморегулювання та механізмів безпеки, керованих громадою, передбачається шлях до відповідального використання LLM.

5.3. Забезпечення конфіденційності та безпеки даних

У множині викликів, пов'язаних з великими мовними моделями (LLM), недоторканість конфіденційності та безпеки даних займає ключову позицію. Оскільки LLM обробляють і генерують величезні обсяги текстової інформації, неминуче виникають проблеми щодо захисту, зберігання та можливого зловживання цими даними. Ці занепокоєння мають не лише технічне значення — вони стосуються самої суті прав особи на конфіденційність, організаційної цілісності та екосистеми довіри, що лежить в основі цифрових взаємодій. У цьому підрозділі розглядається багатогранна сфера конфіденційності та безпеки даних у контексті LLM. Метою є розкрити вразливі місця, серйозність потенційних порушень і цілісні заходи, які розробляються та вживаються, щоб гарантувати, що дані, на яких базуються ці моделі, залишаються безкомпромісними.

5.3.1. Ландшафт загроз

З експоненційним зростанням LLM занепокоєння щодо конфіденційності та безпеки даних загострилося. LLM за своєю конструкцією навчаються з величезних наборів даних, які можуть містити конфіденційну або особисту інформацію. Здатність випадково запам'ятовувати та вигадувати цю інформацію становить значний ризик для конфіденційності. Крім того, зловмисники можуть використовувати LLM для допомоги в кіберзлочинності, ще більше загострюючи проблеми безпеки.

5.3.2. Методи збереження конфіденційності

Були запропоновані різні методи для зменшення ризиків конфіденційності, пов'язаних із LLM. Наприклад, диференціальна конфіденційність забезпечує математичну структуру для кількісної оцінки витоку конфіденційності, хоча її застосування до завдань генерування тексту залишається складним через потенційне

семантичне упередження. Також були введені методи висновку гомоморфного шифрування, щоб забезпечити умови для збереження конфіденційності в LLM як в чорній скриньці.

5.3.3. Регуляторні наслідки

Регуляторний ландшафт розвивається, щоб вирішити проблеми конфіденційності та безпеки даних, пов'язані з LLM. Дотримання встановлених законів про захист даних, таких як Загальний регламент захисту даних (GDPR), має вирішальне значення для розробників і операторів LLM. Крім того, розробляються спеціальні галузеві вказівки та стандарти, щоб забезпечити відповідальне поводження з даними та створити основу для оцінки та пом'якшення ризиків конфіденційності та безпеки, пов'язаних із LLM.

5.3.4. Майбутні напрямки

Стаття під назвою «Конфіденційність у великих мовних моделях: атаки, захист і майбутні напрямки» представляє всебічне обговорення майбутнього збереження конфіденційності в LLM. Вона досліджує потенційні атаки, засоби захисту та окреслює напрямок майбутніх досліджень для забезпечення конфіденційності та безпеки даних у мінливому ландшафті LLM.

Підрозділ підкреслює першочергову важливість забезпечення конфіденційності та безпеки даних у магістратурі права, нові методи збереження конфіденційності та нормативні рамки, створені для захисту конфіденційності даних. Завдяки поєднанню технологічних досягнень, відповідності нормативним вимогам і найкращих галузевих практик передбачається шлях до безпечного та конфіденційного використання LLM.

РОЗДІЛ 6. РЕЗУЛЬТАТИ ТА ОБГОВОРЕННЯ

Дослідження великих мовних моделей (LLM) і безлічі їх застосувань висвітлює технологічний ландшафт, де обробка величезних текстових даних була позитивно переосмислена. Перетворююча сила LLM не обмежується простими теоретичними положеннями. Конкретні результати, як якісні, так і кількісні, підкреслюють відчутний вплив, який вони мають в різних секторах і сферах застосування. Вплітаючи ці моделі в операційну структуру організацій, компанії стали свідками безпрецедентного прискорення своїх робочих процесів. Колись складне завдання просіювання об'ємного текстового вмісту, будь то для вирішення складних технічних проблем або дистиляції складної документації у зрозумілі резюме, було значно спрощено. У цій главі детально розглядаються ці відчутні результати, представлено об'єднання результатів з різних секторів і забезпечено серйозне обговорення їх наслідків. Маємо на меті проаналізувати успіхи, оцінити виклики та запропонувати уявлення про трансформаційний потенціал впровадження LLM у сучасні робочі процеси, підкреслюючи зростаючий ландшафт, де людський досвід посилюється точністю та ефективністю машин.

6.1. Оцінка продуктивності мовних моделей

Мовні моделі, особливо великі мовні моделі (LLM), переосмислили горизонти обробки тексту. Проте, як і з будь-якою технологією, важливо періодично оцінювати їх продуктивність, щоб переконатися, що їхні можливості відповідають вимогам сучасних підприємств, що постійно розвиваються. Ця оцінка стає ще важливішою, коли ці моделі інтегруються в проекти, які вимагають точності, швидкості та адаптивності.

Кафедра КІТ (47)				НАУ 23 28 98 000 ПЗ			
Виконав	Драган В.В.			МОВНІ МОДЕЛІ ДЛЯ АНАЛІЗУ ВЕЛИКИХ ОБ'ЄМІВ ТЕКСТУ	Літера	аркуш	аркушів
Керівник	Климова А.С.					98	8
Консульт.					УС-212М 122		
Н. контроль	Райчев І. Е.						

Один із таких важливих проектів, який є свідченням майстерності LLM, спрямований на оптимізацію громіздкого процесу навігації через складну документацію. Основна мета цього проекту полягає в створенні коротких, зрозумілих і точних підсумків, які дозволили розробникам, технічним фахівцям та іншим професіоналам швидко шукати та отримувати доступ до необхідної інформації без необхідності пробиратися через щільний текстовий вміст.

Практичні досягнення:

Ефективне підбиття підсумків: використовуючи можливості глибокого навчання LLM, проект досяг успіху в перетворенні складних деталей із великих документів у стислі та послідовні резюме.

Швидкий пошук: зведення, точно проіндексовані, дозволяли користувачам швидко визначати відповідні розділи чи теми у величезному просторі документації.

Контекстуальна точність: LLM гарантує, що, незважаючи на стислість резюме, суть і контекстуальна значущість вмісту залишаються недоторканими.

Важливим аспектом цього проекту була його адаптивність. У той час як основним напрямком роботи була технічна документація, основна структура, що базується на LLM, була розроблена так, щоб бути універсальною. Це означає, що за допомогою мінімальних налаштувань проект може бути перепрофільований для допомоги в різноманітних завданнях обробки тексту в різних секторах. Незалежно від того, чи йдеться про юридичні тексти, академічні дослідницькі статті чи бізнес-звіти, ця адаптивність позиціонує проект як цінний інструмент, який може багаторазово підвищити ефективність роботи.

Показники продуктивності:

Швидкість: час, витрачений LLM на обробку та узагальнення вмісту, порівнювався з традиційними методами, що виявило значне скорочення часу обробки.

Точність: Випадкові зразки резюме оцінювалися на відповідність оригінальному вмісту, гарантуючи, що жодна важлива інформація не була втрачена чи спотворена, що продемонстровано на рис. 6.1.

Відгуки користувачів. Кінцеві користувачі, насамперед розробники та технічні спеціалісти, були опитані, щоб оцінити їхнє задоволення та будь-які покращення ефективності робочого процесу.

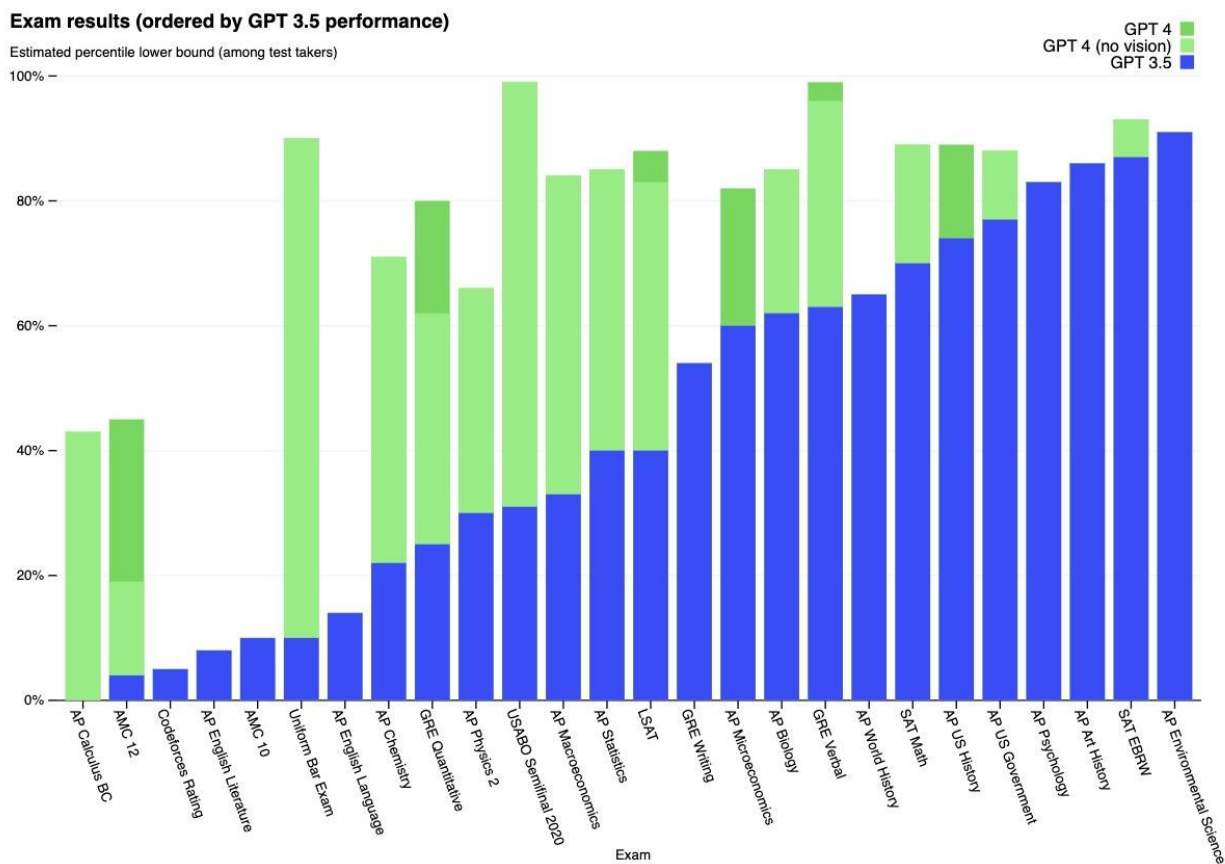


Рис. 6.1. стрибок успішності іспитів між GPT-3.5 і GPT-4.

Результати та наслідки:

Результати були надзвичайно позитивними. Проект не тільки прискорив процес пошуку інформації, але й зменшив когнітивне навантаження на професіоналів, дозволивши їм зосередитися на основних завданнях, а не загрузнути у великій документації. Крім того, успіх проекту підкреслив основне розуміння: включення LLM в організаційні робочі процеси може призвести до трансформаційних переваг у ефективності, точності та задоволеності користувачів.

На завершення оцінка ефективності цього проекту підкреслює величезний потенціал LLM у зміні середовища обробки тексту. Якщо такі проекти розроблені та зважено розгорнуті, вони не тільки задовольняють свої основні цілі, але й

відкривають двері для безлічі інших програм, підвищуючи операційну продуктивність у всіх сферах.

6.2. Реальний вплив та ефективність

Поява великих мовних моделей (LLM) започаткувала нову еру ефективності, що особливо помітно в робочих процесах розробників і технічних спеціалістів. Незважаючи на те, що теоретичні основи LLM широко обговорювалися, реальні наслідки цих моделей на місцях свідчать про їх трансформаційну силу. У цьому підрозділі буде заглиблено у відчутний вплив LLM-ів у сфері розробки програмного забезпечення та надано практичні приклади, що підкреслюють їх ефективність.

1. Документація та розуміння коду:

До LLM: розробники часто витрачали години, якщо не дні, намагаючись зрозуміти застарілий код або погано задокументовані модулі. Процес був виснажливим і спричиняв помилки, що призводило до неефективності.

З LLM: тепер розробники можуть вводити розділи коду в мовні моделі та отримувати чіткі, стислі пояснення чи анотації (рис. 6.2.). Це не тільки прискорює розуміння, але й гарантує, що розробники можуть швидко приймати обґрунтовані рішення.

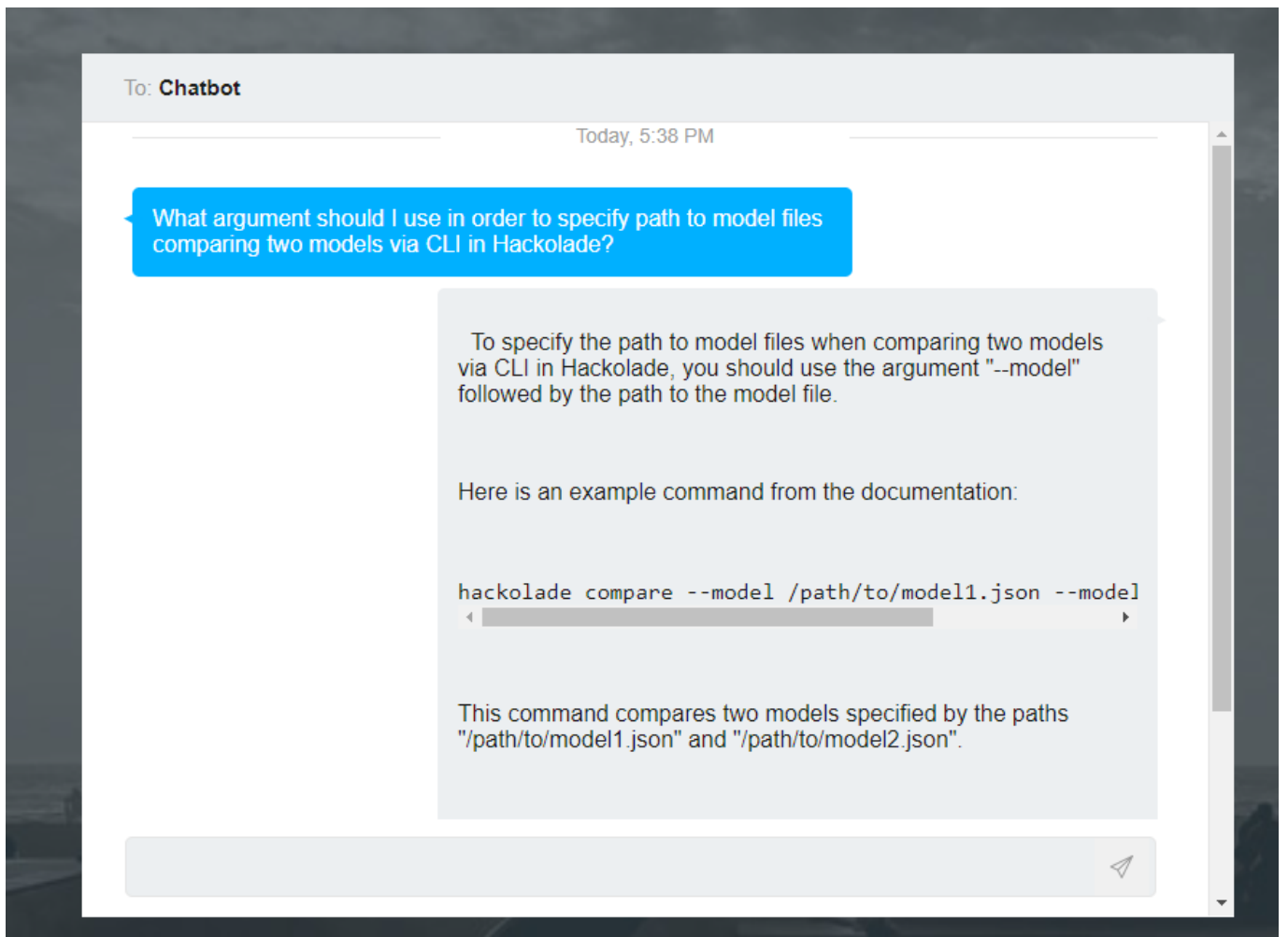


Рис. 6.2. Модель відповідає на запитання посилаючись на документацію

2. Виявлення помилок і усунення несправностей:

До LLM: діагностика помилок або проблем вимагала від розробників вручну переглядати рядки коду або покладатися на обмежені інструменти, які могли пропустити нюанси проблем.

За допомогою LLM: ці моделі можна навчити виявляти шаблони, пов'язані з поширеними помилками чи невідповідностями, пропонуючи пропозиції щодо можливих виправлень. Наприклад, розробник, що бореться з певним повідомленням про помилку, може ввести його в LLM, який потім надає потенційні рішення на основі подібних сценаріїв (Рис. 6.3, Рис. 6.4).

S

You

given two tensors:

```
tensor1 = tf.Tensor([], shape=(2, 0), dtype=float32)
```

```
tensor2 = tf.Tensor([[0.]], shape=(1, 1), dtype=float32)
```

an error occurs when executing `tf.concat([tensor1, tensor2], 0)`: `ERROR: ConcatOp : Dimensions of inputs should match: shape[0] = [2,0] vs. shape[1] = [1,1] [Op:ConcatV2] name: concat`

How can this error be resolved?

Рис. 6.3. Запит на вирішення проблеми розробником



ChatGPT

The error you're seeing is due to the mismatch in the dimensions of `tensor1` and `tensor2`. Specifically, while you are trying to concatenate along axis 0 (i.e., vertically), their second dimensions do not match.

Let's break it down:

- `tensor1` has a shape of `(2, 0)`. This means it has 2 rows and 0 columns.
- `tensor2` has a shape of `(1, 1)`. This means it has 1 row and 1 column.

When you try to concatenate these two tensors vertically (i.e., along axis 0), the number of columns should be the same in both tensors. But as you can see, `tensor1` has 0 columns and `tensor2` has 1 column, causing the mismatch error.

To resolve the error, you need to ensure that the tensors you're concatenating have the same number of columns. Here's one approach to resolve the issue:

1. You can add a dummy column of zeros (or any other value) to `tensor1` to make its shape `(2, 1)` and then concatenate.
2. After concatenation, if the dummy column isn't required, you can slice it out.

Here's a step-by-step solution using TensorFlow:

python

Copy code

```
import tensorflow as tf

# Given tensors
tensor1 = tf.Tensor([], shape=(2, 0), dtype=float32)
tensor2 = tf.Tensor([[0.]], shape=(1, 1), dtype=float32)

# Add a dummy column of zeros to tensor1
tensor1_with_dummy = tf.zeros((2, 1))

# Now concatenate tensor1_with_dummy and tensor2 along axis 0
result = tf.concat([tensor1_with_dummy, tensor2], axis=0)

# If you don't need the dummy values, slice out the required columns.
# But in this case, the dummy column is essentially the result since tensor1 had 0 co

print(result)
```

Рис. 6.4. Мовна модель ChatGPT автоматично вирішує проблему

3. Автоматичний пошук рішень:

До LLM: Коли розробники стикаються з проблемами, часто звертаються до форумів, таких як Stack Overflow. Однак пошук правильного рішення означав просіювання численних тем і відповідей, забираючи дорогоцінний час.

З LLM: тепер розробники можуть коротко описати свою проблему LLM, який потім сканує величезні сховища форумів розробників і повертає найбільш відповідні рішення, спрощуючи процес усунення несправностей (Рис. 24).

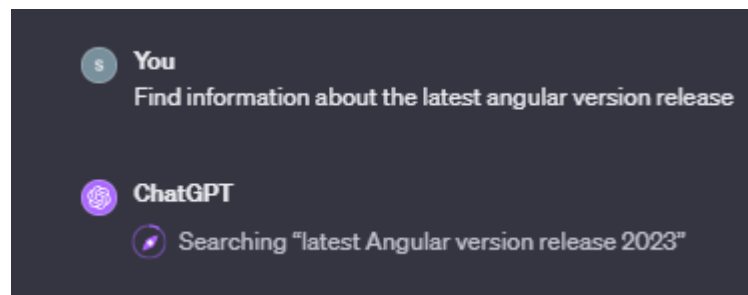


Рис. 6.5. Мовна модель сканує мережу за для пошуку відповідей

4. Автоматична генерація фрагментів коду:

До LLM: для стандартних завдань розробники або вручну писали повторюваний код, або витрачали час на пошук бібліотек або пакетів.

З LLM: за допомогою опису завдання LLM може генерувати фрагменти коду або пропонувати оптимальні бібліотеки. Наприклад, розробник, який хоче реалізувати алгоритм сортування, може просто запитати LLM, який потім створить спеціалізований фрагмент коду, який підходить для цього завдання.

5. Покращення співпраці та спілкування:

До LLM: передача технічних концепцій членам команди, які не є розробниками, часто призводила до непорозумінь або тривалих пояснень.

З LLM: розробники можуть використовувати LLM для перекладу технічного жаргону на зрозумілу мову, сприяючи чіткішому спілкуванню між командами та подолаючи розрив між технічними та нетехнічними зацікавленими сторонами.

Підсумовуючи, інтеграція великих мовних моделей в екосистему розробників стосувалася не лише автоматизації, а й розширення. Перекладаючи певні завдання на LLM, розробники можуть зосередитися на інноваційних, орієнтованих на цінності починаннях. Крім того, справжній вплив і ефективність цих моделей полягає в їх здатності інтуїтивно з'єднуватися з людськими робочими процесами, каталізуючи синергетичний зв'язок, який покращує весь процес розробки.

РОЗДІЛ 7. ПЕРСПЕКТИВИ МАЙБУТНЬОГО РОЗВИТКУ

Сфера великих мовних моделей (LLM) — це захоплюючий ландшафт, готовий до безлічі досягнень і проривів, які обіцяють розширити межі можливостей і програм ШІ. Ця траєкторія спрямована не лише на вдосконалення основних функціональних можливостей, але й на вирішення переважаючих проблем, прокладаючи таким чином шлях до покоління більш універсальних, практичних і ефективних LLM.

7.1. Дослідження та досягнення

Очікується, що майбутня хвиля LLM буде наповнена захоплюючими функціями, такими як покращена точність, розширена підтримка мови та більша обчислювальна ефективність, і все це спрямовано на те, щоб зробити ці моделі ще надійнішими та зручнішими, ніж їхні попередники. Ключові підходи, як-от самопідготовка, перевірка фактів і обмежений досвід, досліджуються, щоб згладити існуючі проблеми та розкрити повний потенціал LLM.

7.1.1. Покращена взаємодія людини з комп'ютером

Нещодавні досягнення в LLM започаткували фазу трансформації в ШІ, зокрема покращивши діалог між людиною та комп'ютером. LLM, такі як ChatGPT, продемонстрували вміння імітувати людські взаємодії, таким чином задовольняючи зростаючий попит на різноманітні послуги. Ця трансформація натякає на майбутнє, де LLM зможуть запропонувати більш природні та інтуїтивно зрозумілі інтерфейси для низки програм, від обслуговування клієнтів до персональних цифрових помічників.

Кафедра КІТ (47)				НАУ 23 28 98 000 ПЗ			
Виконав	Драган В.В.			МОВНІ МОДЕЛІ ДЛЯ АНАЛІЗУ ВЕЛИКИХ ОБ'ЄМІВ ТЕКСТУ	Літера	аркуш	аркушів
Керівник	Климова А.С.					106	7
Консульт.					УС-212М 122		
Н. контроль	Райчев І. Е.						

7.1.2. Мультимодальні можливості

Помітною еволюцією є інтеграція мультимодальних можливостей у LLM. OpenAI GPT-4, наприклад, був навчений як мультимодальна модель, яка поєднує аналіз тексту з сенсорним введенням, яке може бути представленим у вигляді зображень. Ця еволюція має фундаментальне значення для покращення розуміння і взаємодії LLM з реальним світом, створення більш цілісного штучного інтелекту, який може обробляти та інтерпретувати кілька типів даних одночасно.

7.1.3. Вирішення питання фактичної надійності

Прагнення до фактичної надійності є наріжним каменем поточних досягнень LLM. Інновації, такі як пошук зовнішньої інформації та надання довідок, були запроваджені, щоб згладити притаманну фактичну ненадійність LLM. Наприклад, WebGPT від OpenAI і Sparrow від DeepMind мають можливість переглядати в Інтернеті актуальну інформацію, таким чином підвищуючи точність і достовірність своїх відповідей. Ці розробки є ключовими для підготовки LLM до розгортання в реальному світі, оскільки вони сприяють підвищенню прозорості та довіри користувачів.

7.1.4. Розріджені експертні моделі

Дослідження розріджених експертних моделей пропонує свіжий погляд на архітектуру LLM, що відходить від звичайних щільних моделей, таких як GPT-3. Розріджені моделі активують лише підмножину параметрів, що мають відношення до даного вхідного сигналу, тому є більшими та менш вимогливими до обчислень. Такі моделі, як Google GLaM, є прикладом цього підходу, демонструючи значні переваги з точки зору розміру, енергії та обчислювальної ефективності. Розріджені експертні моделі натякають на зміну парадигми, що потенційно призведе до більш

ефективних і потужних LLM, здатних обробляти ширший спектр завдань без пропорційного збільшення обчислювальних вимог.

7.1.5. Інвестиції та залучення громади

У 2023 році відбулося стрімке зростання інвестицій у LLM, що свідчить про великий інтерес і віру в їхній потенціал. Очікується, що цей фінансовий приплив сприятиме подальшим дослідженням, розробці та розгортанню LLM у різних сферах, прискорюючи інноваційний цикл і швидше виводячи на ринок ефективніші моделі.

7.1.6. Пристосування до мінливих світових знань

Підтримка актуального статусу LLM після розгортання є актуальною проблемою. Останні досягнення зосереджені на узгодженні LLM зі світовими знаннями, що постійно змінюються, без необхідності повторного навчання з нуля, що є кроком до того, щоб LLM залишалися актуальними та точними з часом⁵.

Підрозділ узагальнює поточні дослідження та досягнення в галузі LLMs, підкреслюючи багатогранні зусилля з подолання існуючих проблем і розкриття нових потенціалів. Завдяки поєднанню технологічних інновацій, участі спільноти та значних інвестицій, траєкторія LLM спрямована на багатообіцяюче майбутнє, яке характеризується розширеними можливостями, надійністю та більш широкою сферою застосування.

7.2. Трансформуючий вплив

LLM стоять на порозі суттєвої трансформації різних сфер, включаючи науку, суспільство та взаємодію між ШІ та людьми. Поєднання можливостей лінгвістики та машинного навчання в LLM розглядається як каталізатор суттєвих змін у цих сферах. Випуск таких моделей, як ChatGPT, очевидно, викликав інтерес до LLM,

вказуючи на світле майбутнє, наповнене подальшими вдосконаленнями та покращеннями, які відповідають очікуванням користувачів, особливо в сфері великих мовних моделей

7.2.1. Покращена взаємодія людини з комп'ютером

LLM, такі як ChatGPT і Bard, плавно влилися в наше повсякденне життя, спрощуючи численні завдання завдяки своїй здатності генерувати текст, схожий на людський. Вони широко застосовуються в різних галузях завдяки своїй універсальності, зрозумілості та швидкому відгуку. Їхня масштабованість дозволяє обробляти кілька взаємодій паралельно, таким чином автоматизуючи рутинні завдання та мінімізуючи використання людських ресурсів.

7.2.2. Динаміка суспільства

Використання LLM може суттєво змінити соціальні взаємодії та динаміку довіри. Дослідження Hohenstein et al. вказали, що ефективність розмов, керованих ШІ, значною мірою залежить від того, чи знають люди, що вони взаємодіють з алгоритмом. Також було відзначено, що інтеграція LLM у соціальні медіа-платформи та інші додатки забезпечує форму спілкування, зменшуючи відчуття самотності серед користувачів усіх вікових груп.

7.2.3. Вплив на охорону здоров'я

Сектор охорони здоров'я також очікує трансформаційної сили LLM. Згідно з опитуванням Gartner, 55% IT-директорів і технологічних лідерів платників медичних послуг вважають, що магістерські програми матимуть трансформаційний або руйнівний вплив на галузь охорони здоров'я. Ранні впровадження LLM в охорону здоров'я охоплюють різні випадки використання, підкреслюючи їхній потенціал революції в наданні медичних послуг.

7.2.4. Наукові дослідження та написання

LLM пробиваються в науковому письмі, особливо в медичній літературі, де вони допомагають у синтезі інформації, лінгвістичному вдосконаленні та глобальному поширенні знань. Інтеграція LLM та штучного інтелекту в наукову літературу створює як безпрецедентні можливості, так і виклики, пропонуючи новий шлях для прискорення досліджень і обміну знаннями.

7.2.5. До узагальнення та універсальності

LLM є кроком ближче до досягнення надзвичайної універсальності людської мови. Це попередньо підготовлені базові моделі, які можна адаптувати до широкого спектру завдань природної мови, кожна з яких раніше вимагала окремої моделі мережі. Ця розробка підкреслює трансформаційну природу LLM, наближаючись до більш узагальненої моделі, яка може вирішувати ширший спектр завдань.

Підрозділ пояснює багатогранний трансформаційний вплив магістерських програм, підкреслюючи, як вони переосмислюють взаємодію людини та комп'ютера, змінюють суспільну динаміку та роблять значний внесок у охорону здоров'я, наукові дослідження та багато інших сфер. Завдяки безперервному вдосконаленню магістратури готові ще більше подолати розрив між машинним розумінням і людською мовою, відкриваючи нову еру рішень на основі ШІ, які можуть вирішувати ширший спектр реальних проблем.

7.3. Інтеграція з новими технологіями

Очікується, що синергія між LLM та іншими новими технологіями стане ключовою темою майбутніх розробок. Інтеграція розглядається як провідник для створення більш цілісних і ефективних рішень, які вирішують ширший спектр реальних проблем.

7.3.1. Багатомодельна співпраця:

Дослідники з Массачусетського технологічного інституту оприлюднили новаторський підхід, у якому кілька моделей штучного інтелекту, включно з магістрами, співпрацюють, обговорюють і вдосконалюють свої здібності міркувати, щоб генерувати точніші та фактично послідовні відповіді. Ця спільна структура втілює прислів'я «Дві голови краще, ніж одна», використовуючи колективний інтелект кількох агентів штучного інтелекту для отримання більш витонченої та точної відповіді на поставлене запитання¹.

7.3.2. Охорона здоров'я:

Особливо помітною є інтеграція LLMs у секторі охорони здоров'я. Наприклад, LLMs використовуються для обробки великих обсягів текстових даних для отримання цінної інформації, яка може мати вирішальне значення для медичної діагностики та досліджень.

7.3.3. Технологічні платформи:

LaMDA від Google: Google розробив LaMDA, велику мовну модель, спрямовану на участь у більш тонких і зв'язних розмовах з користувачами за допомогою свого інструменту Bard. Хоча спочатку працював на базі LaMDA, Bard пізніше перейшов на більш складну модель, PaLM 2, підкреслюючи безперервний розвиток та інтеграцію LLM з пошуковими технологіями.

ChatGPT від OpenAI: у січні 2023 року ChatGPT зібрав колосальні 100 мільйонів активних користувачів щомісяця, демонструючи високий попит і різноманітні програми LLM, як-от генерування тексту, аналіз настроїв і створення контенту.

7.3.4. Міжгалузевий вплив:

Розгортання LLM у різних галузях, таких як охорона здоров'я, фінанси, розваги та освіта, є свідченням їх інтеграції з різними сучасними технологічними рамками. Реальні застосування LLM розширюються, оскільки вони адаптуються до конкретних потреб різних областей.

7.3.5. Розширені можливості:

Запуск GPT-4 компанією OpenAI у березні 2023 року став важливою віхою в спільноті ШІ. GPT-4 продемонстрував розширені можливості, такі як складне міркування, кодування та майстерність у академічних іспитах, підкреслюючи синергію між LLM та іншими сучасними технологічними досягненнями для досягнення людської продуктивності.

7.3.6. Розробка програмного забезпечення:

Інтеграція LLM з платформами розробки програмного забезпечення є ще однією помітною тенденцією. Наприклад, Copilot від GitHub працює на базі OpenAI GPT-3 і допомагає розробникам писати новий код, розуміти існуючий код і працювати з ним.

Переплетення LLM з іншими сучасними технологіями не тільки покращує функціональні можливості, але й прокладає шлях до цілісних рішень реальних проблем. Ця інтеграція відіграє вирішальну роль у подоланні традиційної розрізненості технологічних функцій, таким чином сприяючи створенню сприятливого середовища для інноваційних рішень і досягнень у різних сферах.

ВИСНОВКИ

У цій роботі було досліджено трансформаційний потенціал великих мовних моделей для аналізу тексту, починаючи з вивчення еволюції цих моделей, простеживши їх розвиток від традиційних систем, заснованих на правилах, до появи глибокого навчання та появи таких моделей, як GPT-3 та GPT-4. Ці моделі з їх величезними параметрами та механізмами уваги зробили революцію в області обробки природної мови.

Було розглянуто різноманітні методи аналізу тексту, від токенізації та аналізу настроїв до машинного перекладу та систем відповідей на запитання. Ці методи служать будівельними блоками для отримання інформації з текстових даних, демонструючи універсальність великих мовних моделей.

Було детально описано технічний стек для аналізу тексту, наголошуючи на використанні JavaScript як для розробки серверної, так і зовнішньої частини. Було підкреслено універсальність і ефективність JavaScript, зокрема в інтерфейсі з OpenAI API та управлінні векторною базою даних, розміщеною в хмарі. Ця інтеграція забезпечує зручну взаємодію з користувачем і дозволяє програмі ефективно взаємодіяти з великими мовними моделями.

Векторна база даних стала ключовим компонентом, який служить динамічною системою зберігання документації. Інкапсулюючи текстові представлення у векторні вбудовування, база даних забезпечує ефективний аналіз подібності, кластеризацію та контекстний аналіз. Він також підтримує навчання та налагодження моделей, сприяючи створенню середовища для співпраці та багатих знаннями.

Крім того, було досліджено значення хмарних рішень для розгортання великих мовних моделей. Хмарні платформи пропонують масштабованість, розподілені обчислення, керовані послуги хостингу, спеціалізоване обладнання, оптимізацію витрат, глобальну доступність і надійні заходи безпеки. Ці функції спільно вирішують обчислювальні проблеми, пов'язані з великими мовними моделями, дозволяючи організаціям ефективно використовувати їхній потенціал.

Підсумовуючи, великі мовні моделі вийшли за межі звичайного аналізу тексту, пропонуючи безпрецедентні можливості розуміння та генерування тексту. Їх застосування охоплює різноманітні галузі, від підтримки клієнтів до створення контенту, і їхній вплив на такі сфери, як охорона здоров'я та юридичний аналіз, глибокий. Завдяки правильному технічному стеку та хмарній інфраструктурі організації можуть використовувати ці моделі для отримання значущої інформації, кардинально змінивши спосіб обробки та розуміння текстових даних. Подорож великих мовних моделей продовжує розгортатися, обіцяючи ще більший прогрес у сфері обробки природної мови.

Внесок у галузь

У епоху, коли інформація є одночасно і валютою, і каталізатором, еволюція великих мовних моделей (LLM) являє собою значний стрибок у тому, як галузі обробляють, розуміють і використовують текстові дані. Коли ми розмірковуємо про багатогранні використання LLM, описані в попередніх розділах, вкрай важливо підкреслити їхній трансформаційний внесок у ширший промисловий ландшафт. У цьому підрозділі пояснюється зміна парадигми, започаткована LLM, і те, як вони змінили галузеві стандарти та очікування.

1. Перевизначення ефективності:

Розгортання LLM значно скоротило час, необхідний для таких завдань, як обробка даних, аналіз тексту та створення вмісту. Галузі, які колись покладалися на величезні людські ресурси для інтерпретації та обробки даних, тепер можуть досягти таких самих, якщо не кращих, результатів за частку часу. Ця прискорена ефективність проклала шлях до швидшого прийняття рішень і більш гнучкої бізнес-операцій.

2. Експертиза демократизації:

LLM зробили спеціальні знання доступними. Будь то юридичний жаргон, технічна документація чи наукова література, ці моделі можуть перетворювати складну інформацію на зрозумілі ідеї. Ця демократизація надала можливість навіть

тим, хто не має спеціального досвіду, приймати обґрунтовані рішення, вирівнюючи умови гри в багатьох секторах.

3. Підвищення креативності та інновацій:

Виконуючи повторювані та повсякденні завдання, LLM звільнили людський капітал, щоб зосередитися на більш творчих і стратегічних починаннях. У таких сферах, як маркетинг, дизайн і дослідження, професіонали тепер можуть приділяти більше часу ідеям та інноваціям, підкріплюючись ідеями та даними, які пропонують LLM.

4. Каталізуючі налаштування:

Адаптивний характер LLM дозволяє галузям пристосовувати свої функції до конкретних потреб. Цей рівень персоналізації призвів до більш персоналізованого досвіду клієнтів, розробки продуктів і пропозицій послуг, дозволяючи компаніям точно обслуговувати ринкові ніші та нові тенденції.

5. Встановлення нових етичних стандартів і стандартів управління:

Становлення LLM також спонукало галузі переглянути та переглянути свої етичні принципи та структури управління. З огляду на занепокоєння щодо конфіденційності даних, неправильного використання та упередженості, галузі зараз знаходяться в авангарді встановлення суворих протоколів, щоб гарантувати, що розгортання LLM відповідає суспільним цінностям і нормам.

6. Сприяння міжгалузевому співробітництву:

Універсальність LLM сприяла співпраці між традиційно різнорідними секторами. Наприклад, галузь охорони здоров'я та технологічна індустрія об'єднуються навколо LLM для відкриття ліків, догляду за пацієнтами та медичних досліджень, проголошуючи нову еру міждисциплінарних інновацій.

По суті, внесок великих мовних моделей у галузь виходить за межі простого технологічного прогресу. Вони представляють собою злиття ефективності, справедливості, інновацій та відповідальності. На даному етапі стає очевидним, що LLM є не просто інструментами сьогодення, а й архітекторами майбутнього, які постійно формуються та покращуються під впливом постійних змін промислової екосистеми.

Вплив на промисловість та науку

Поява та швидка еволюція великих мовних моделей (LLM) накреслила безпрецедентну траєкторію впливу не лише в галузях промисловості, але й у священних коридорах наукових досліджень. Коли підходимо до завершення цього дискурсу, вкрай важливо окреслити широкий спектр впливу LLM, висвітлюючи, як вони стали невід’ємним стрижнем як у торгівлі, так і в академічних колах. Цей підрозділ намагається охопити трансформаційні наслідки LLM для промисловості та науки.

1. Відновлення промислових процесів:

Оптимізовані операції: LLM здійснили революцію в галузях, автоматизувавши складні текстові процеси, від аналізу контрактів у юридичному секторі до взаємодії з клієнтами в роздрібній торгівлі. Ця автоматизація забезпечила операційну ефективність, яка раніше вважалася недосяжною.

Обґрунтоване прийняття рішень. Переробляючи величезні резервуари даних у практичні ідеї, LLM дають можливість компаніям приймати рішення на основі даних, зміцнюючи як стратегічний напрямок, так і тактичні впровадження.

2. Сприяння інноваціям у розробці продукту:

Швидке створення прототипів: маючи можливість швидко генерувати описовий і технічний зміст, магістри LLM пришвидшили етап створення прототипів у розробці продукту, дозволяючи швидше проходити перевірку на ринку.

Персоналізовані пропозиції: LLM, завдяки детальному розумінню вподобань користувачів і відгукам, допомагають галузям у розробці індивідуальних продуктів і послуг, підвищуючи задоволеність клієнтів і утримуючи їх.

3. Каталізатор наукових відкриттів:

Аналіз даних: Наукові дослідження часто мають справу з об’ємними наборами даних. LLM допомагають дослідникам аналізувати ці дані, виявляючи закономірності та кореляції, які могли б бути пропущені лише людським аналізом.

Огляди літератури: LLM можуть швидко аналізувати величезні сховища наукових робіт, надаючи дослідникам вичерпні огляди та резюме, гарантуючи, що наукові зусилля добре поінформовані та ґрунтуються на існуючих знаннях.

4. Сприяння міждисциплінарній співпраці:

Подолання прогалів у знаннях: LLM, завдяки своїй здатності інтерпретувати та перекладати складну термінологію в різних галузях, сприяли співпраці між різними науковими дисциплінами, що призвело до цілісних результатів досліджень.

Синергія промисловості та науки: діючи як посередники, LLM також активізували співпрацю між промисловими науково-дослідними, дослідно-конструкторськими роботами та академічними дослідженнями, гарантуючи, що наукові відкриття знаходять прагматичне застосування в реальних сценаріях.

5. Демократизація доступу до знань:

Співпраця з відкритим вихідним кодом: LLM відіграють ключову роль у проектах з відкритим кодом, допомагаючи розробникам із пропозиціями коду, виявленням помилок і документацією, тим самим підвищуючи загальну якість і охоплення ініціатив з відкритим кодом.

6. Удосконалення освіти

у сфері освіти LLM є безцінними інструментами, надаючи студентам спеціалізований контент, допомагаючи викладачам у розробці навчальних програм і дозволяючи дослідникам бути в курсі світових досягнень.

7. Встановлення етичних орієнтирів у дослідженнях:

Виявлення упереджень: оскільки наука прагне до об'єктивності, LLM допомагають дослідникам виявляти та виправляти ненавмисні упередження в їхніх дослідженнях, забезпечуючи збалансованість та справедливість результатів досліджень.

Цілісність даних: LLM, перехресно посилаючись на величезні джерела даних, допомагають перевірити цілісність наукових даних, зміцнюючи достовірність результатів досліджень.

Підсумовуючи, злиття великих мовних моделей з промисловістю та наукою символізує трансформаційну синергію (Рис. 8.1).

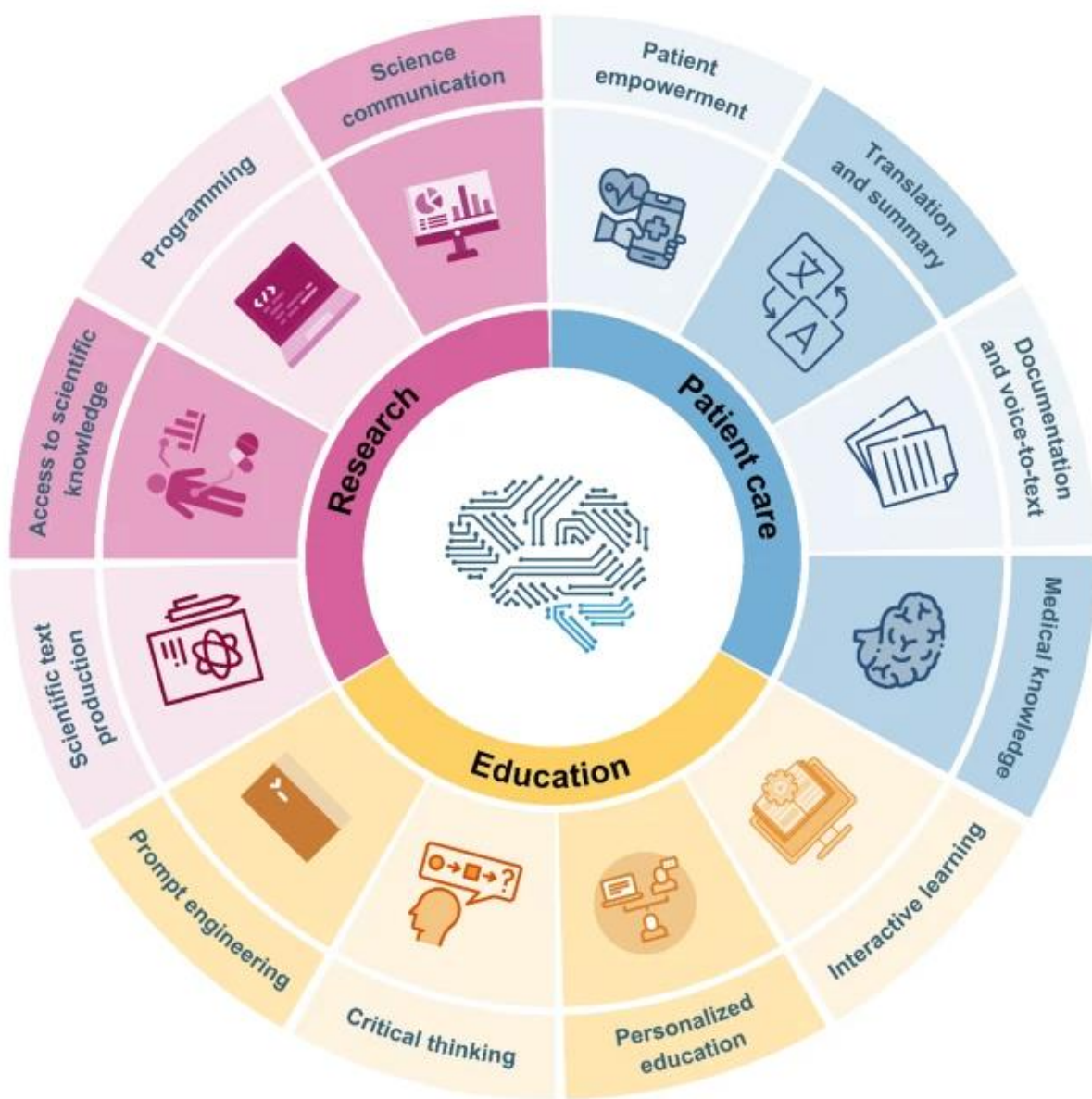


Рис. 8.1 Включення мовних моделей у всі сфери людського життя.

Завдяки плавній інтеграції в різноманітні робочі процеси LLM перекалібрували очікування, встановили нові критерії та намітили шляхи для майбутніх досліджень. Їхній вплив, хоч і глибокий сьогодні, є лише передвісником ще більш глибоких революцій, які вони обіцяють завтра.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Attention Is All You Need. [Електронний ресурс] // BuiltIN. – 2023. – Режим доступу до ресурсу: <https://arxiv.org/abs/1706.03762>.
2. Large Language Model: world models or surface statistics? [Електронний ресурс] // The Gradient. – 2023. – Режим доступу до ресурсу: <https://thegradient.pub/othello/>.
3. The Unpredictable Abilities Emerging From Large AI Models [Електронний ресурс] // LAPAAS. – 2023. – Режим доступу до ресурсу: <https://www.quantamagazine.org/the-unpredictable-abilities-emerging-from-large-ai-models-20230316/>
4. <https://cdn.openai.com/papers/gpt-4.pdf> [Електронний ресурс] / Abby Jenkins // OpenAI. – 2023. – Режим доступу до ресурсу: <https://cdn.openai.com/papers/gpt-4.pdf>.
5. 20B-parameter Alexa model sets new marks in few-shot learning? [Електронний ресурс] // Amazon. – 2023. – Режим доступу до ресурсу: <https://www.amazon.science/blog/20b-parameter-alexa-model-sets-new-marks-in-few-shot-learning>.
6. UAE's Falcon 40B, World's Top-Ranked AI Model from Technology Innovation Institute, is Now Royalty-Free [Електронний ресурс] // Businesswire. – 2023. – Режим доступу до ресурсу: <https://www.businesswire.com/news/home/20230531005608/en/UAE%E2%80%99s-Falcon-40B-World%E2%80%99s-Top-Ranked-AI-Model-from-Technology-Innovation-Institute-is-Now-Royalty-Free>
7. Google's newest A.I. model uses nearly five times more text data for training than its predecessor [Електронний ресурс] / Elias, Jennifer. – 2023. – Режим доступу до ресурсу: <https://www.cnbc.com/2023/05/16/googles-palm-2-uses-nearly-five-times-more-text-data-than-predecessor.html>
8. Introducing LLaMA: A foundational, 65-billion-parameter large language model [Електронний ресурс] // Forto. – 2023. – Режим доступу до ресурсу: <https://ai.meta.com/blog/large-language-model-llama-meta-ai/>

9. Privacy in Large Language Models: Attacks, Defenses and Future Directions
[Электронный ресурс] // Haoran Li. – 2023. – Режим доступа до ресурсу:
[https://arxiv.org/abs/2310.10383#:~:text=,Hu%2C%20Chunkit%20Chan%2C%20Y
angqiu%20Song](https://arxiv.org/abs/2310.10383#:~:text=,Hu%2C%20Chunkit%20Chan%2C%20Yangqiu%20Song)
10. Self-Instruct: Aligning Language Models with Self-Generated Instructions
[Электронный ресурс] // Yizhong Wang. – 2022. – Режим доступа до ресурсу:
<https://arxiv.org/abs/2212.10>

ДОДАТОК А

```
import { Injectable } from '@nestjs/common';
import {
  PineconeClient,
  utils as pineconeUtils,
} from '@pinecone-database/pinecone';
import { OpenAIApi } from 'openai';
import * as crypto from 'crypto';

import { DocsPart } from 'src/common/types';

import { PINECONE_INDEX_NAME } from './indexing.constants';

@Injectable()
export class IndexingService {
  constructor(
    private readonly pineconeClient: PineconeClient,
    private readonly openaiApi: OpenAIApi,
  ) {}

  public async indexExists() {
    return (await this.pineconeClient.listIndexes()).includes(
      PINECONE_INDEX_NAME,
    );
  }

  public async indexIsEmpty() {
    return (
      (
        await this.getPineconeIndex().describeIndexStats({
          describeIndexStatsRequest: {},
        })
      ).totalVectorCount === 0
    );
  }

  public async updateIndex(chunks: DocsPart[]) {
    console.log('##### creating embeddings...')
    const embeddingObjects = await this.createEmbeddings(
      chunks.map((mapChunk) => mapChunk.content),
    );
    console.log('embeddings created!')
    await pineconeUtils.chunkedUpsert(
```

```

        this.getPineconeIndex(),
        embeddingObjects.map((mapEmbeddingObject) => ({
id: this.calculateDocChunkContentHash(
            chunks[mapEmbeddingObject.index].content,
        ),
        values: mapEmbeddingObject.embedding,
        metadata: chunks[mapEmbeddingObject.index],
    })),
    PINECONE_INDEX_NAME,
);
}

public async query(prompt: string, chunksToIncludeNumber: number) {
    const [promptEmbedding] = await this.createEmbeddings([prompt]);
    const queryResponse = await this.getPineconeIndex().query({
        queryRequest: {
            topK: chunksToIncludeNumber,
            includeMetadata: true,
            vector: promptEmbedding.embedding,
            namespace: PINECONE_INDEX_NAME,
        },
    });

    if (queryResponse.matches.length === 0) {
        throw new Error('The index is empty.');
```

```

    }

    return queryResponse.matches.map(
        (mapMatch) => mapMatch.metadata as DocsPart,
    );
}

```

```

private calculateDocChunkContentHash(content: string) {
    return crypto.createHash('md5').update(content).digest('hex');
}

```

```

private async createEmbeddings(input: string[]) {
    return (
        await this.openaiApi.createEmbedding({
            input,
            model: 'text-embedding-ada-002',
        })
    ).data.data;
}

```

```

private getPineconeIndex() {
    return this.pineconeClient.Index(PINECONE_INDEX_NAME);
}

```

ДОДАТОК Б

```
const sendBtn = document.querySelector(
  '.container .chat-block .write .write-link.send',
);
const spinner = document.getElementById('loading');

const userMessageElementTemplate = document.createElement('div');
userMessageElementTemplate.classList.add('bubble', 'you');

const assistantMessageElementTemplate = document.createElement('div');
assistantMessageElementTemplate.classList.add('bubble', 'me');

sendBtn.addEventListener('click', async (_event) => {
  const promptInput = document.getElementById('prompt-input');
  const promptStr = promptInput.value;
  promptInput.value = '';

  sendBtn.classList.toggle('hidden');
  spinner.classList.toggle('hidden');

  const currentChat = document.querySelector('.chat.active-chat');

  const userMessageElement = userMessageElementTemplate.cloneNode();

  userMessageElement.innerText = promptStr;
  currentChat.appendChild(userMessageElement);

  const res = await fetch('/completion/complete', {
    method: 'POST',
    body: JSON.stringify({ prompt: promptStr }),
    headers: {
      'Content-Type': 'application/json',
    },
  });

  const assistantMessageElement = assistantMessageElementTemplate.cloneNode();
  const { text, links } = await res.json();
  const replaceRelativeDocPageUrlRegex = /\[([.*])\]\[([.*\.html])\]/g;
  const replaceImgUrlRegex = /!\[([.*])\]\[lib\/([.*])\]/g;
  // const replaceExternalUrlRegex = /\[([.*])\]\[([http.*])\]/g;
  const replacePlainUrlRegex = /(http.*)/g;
  // extract image urls and convert them to "img" tags
  const formattedText = text
    .replace(replaceImgUrlRegex, '')
    // .replace(replaceExternalUrlRegex, '<a href="$2">$1</a>')
```

```
.replace(
  replaceRelativeDocPageUrlRegex,
  '<a href="https://website.com/help/$2">$1</a>',
);

const formattedLinks = links
  .join('\n')
  .replace(replacePlainUrlRegex, '<a href="$1">$1</a>');

assistantMessageElement.innerHTML = marked.parse(`
${formattedText}

Additional links: \n${formattedLinks}
`);

assistantMessageElement.style.whiteSpace = 'break-spaces';
currentChat.appendChild(assistantMessageElement);

sendBtn.classList.toggle('hidden');
spinner.classList.toggle('hidden');
});
```