

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВОЛИНСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ ІМЕНІ ЛЕСІ УКРАЇНКИ**

Кафедра прикладної лінгвістики

На правах рукопису

ШУМИЛКО ОКСАНА ОЛЕГІВНА

**АДАПТАЦІЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ ДЛЯ СТВОРЕННЯ ЧАТ-
БОТІВ З КОНТЕКСТНОЮ ПІДТРИМКОЮ**

Спеціальність: 035 «Філологія»

Освітня програма: Прикладна лінгвістика. Переклад і комп'ютерна
лінгвістика Робота на здобуття освітнього ступеня «Магістр»

Науковий керівник:

ЛИННИК ЮРІЙ МИКОЛАЙОВИЧ,
кандидат педагогічних наук, доцент
кафедри прикладної лінгвістики

РЕКОМЕНДОВАНО ДО ЗАХИСТУ

Протокол № 5

засідання кафедри прикладної лінгвістики

від 25.11. 2025 р.

Завідувачка кафедри

 доц. Калиновська І.М

(підпис)

ПБ

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. ВЕЛИКІ МОВНІ МОДЕЛІ: АРХІТЕКТУРНІ РІШЕННЯ ТА ФУНДАМЕНТАЛЬНІ ПРИНЦИПИ ФУНКЦІОНУВАННЯ.....	5
1.1. Еволюційний розвиток та архітектурні особливості великих мовних моделей.....	5
1.2. Сучасні методологічні підходи до навчання великих мовних моделей	18
1.3. Методи оптимізації та адаптації великих мовних моделей для вирішення спеціалізованих задач	24
Висновки до 1-го розділу.....	35
РОЗДІЛ 2. ДІАЛОГОВІ СИСТЕМИ З КОНТЕКСТНОЮ ПІДТРИМКОЮ: ТЕОРЕТИЧНІ ЗАСАДИ ТА МЕТОДОЛОГІЧНИЙ БАЗИС.....	37
2.1. Контекст у діалогових системах: концептуалізація та методологія управління.....	37
2.2. Методика оцінювання якості функціонування діалогових систем з контекстною підтримкою	41
Висновки до 2-го розділу.....	45
РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА ІМПЛЕМЕНТАЦІЯ БАГАТОМОВНОЇ ДІАЛОГОВОЇ СИСТЕМИ.....	46
3.1. Проектування системної архітектури та обґрунтування вибору технологічного стеку	46
3.2. Експериментальні дослідження та аналітична інтерпретація отриманих результатів.....	63
Висновки до 3-го розділу.....	67
ВИСНОВКИ.....	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	70
ДОДАТКИ.....	75

ВСТУП

Актуальність дослідження. На сучасному етапі цифрової трансформації суспільства великі мовні моделі (ВММ) стали фундаментальною технологією, яка революціонізує способи взаємодії людини з комп'ютерними системами. Розвиток архітектур на базі трансформерів, зокрема моделей GPT, BERT та Gemini, відкрив нові можливості для створення інтелектуальних діалогових систем, здатних розуміти природну мову та підтримувати складні багатокрокові діалоги. Фундаментальні дослідження у цій галузі було проведено командою Google Research, А. Радфордом, Дж. Делвіном, Л. Оуяном та З. Ханом. Концептуалізацію поняття контексту в діалогових системах розглядали Б. Грош та Я. Дерю , а методологію оцінювання якості діалогових систем розробили Н. Радзвіль та М. Бентон.

Водночас, аналіз наукових праць підтверджує, що комплексні дослідження адаптації великих мовних моделей для створення багатоагентних чат-ботів з контекстною підтримкою залишаються недостатньо розробленими, особливо у контексті обмежених технічних ресурсів.

Теоретичне і практичне значення великих мовних моделей, актуальність розробки методів їх адаптації для створення спеціалізованих діалогових систем з контекстною підтримкою, а також необхідність систематизації підходів до проектування багатоагентних чат-ботів в умовах обмежених ресурсів зумовили вибір теми нашого дослідження: *«Адаптація великих мовних моделей для створення чат-ботів з контекстною підтримкою»*.

Об'єктом нашого дослідження є архітектура великих мовних моделей, механізми навчання та особливості роботи.

Предметом дослідження є адаптація великих мовних моделей для створення чат-ботів з контекстною підтримкою.

Метою дослідження є дослідження методів адаптації великих мовних моделей в умовах обмежених технологічних ресурсів та розробка багатоагентного Telegram чатбота.

Відповідно до об'єкта, предмета та мети, визначено основні **завдання дослідження**:

1. Проаналізувати сучасні методи адаптації великих мовних моделей.
2. Дослідити архітектурні підходи побудови великих мовних моделей.
3. Дослідити механізми управління контекстом
4. Розробити архітектуру чатбота з розподілом ролей між агентами.
5. Провести експериментальне тестування продуктивності та якості роботи розробленої системи.

Методи дослідження: аналіз теоретичних джерел, порівняння, систематизація та узагальнення теоретичних і дослідних даних, описовий метод, метод збору інформації, вивчення й узагальнення наукового досвіду та емпіричний метод дослідження, метод прототипування, експериментальний метод та статистичний аналіз.

Практичне значення дослідження полягає у тому, що отримані результати можуть бути використані при розробці інтелектуальних чат-ботів для бізнесу та освітніх закладів з обмеженим бюджетом на інфраструктуру; у викладанні дисциплін з машинного навчання, обробки природної мови та розробки AI-систем; як основа для подальших досліджень ресурсоефективних підходів до адаптації великих мовних моделей. Розроблена система тестування та метрики оцінювання якості діалогових систем можуть застосовуватися для порівняльного аналізу різних підходів до створення чат-ботів.

Апробація результатів та публікації: 4 листопада 2025 року – Публікація тез та участь у ІХ Всеукраїнській науково-практичній конференції «Лінгвістичні горизонти ХХІ століття». Тема виступу: *Адаптація великих мовних моделей для створення чат-ботів з контекстною підтримкою.*

Структура магістерської роботи складається зі вступу, трьох розділів, висновків до розділів, загальних висновків, списку використаних джерел (68 найменувань), 6 додатків, містить 17 рисунків. Повний обсяг роботи – 124 сторінки.

РОЗДІЛ 1. ВЕЛИКІ МОВНІ МОДЕЛІ: АРХІТЕКТУРНІ РІШЕННЯ ТА ФУНДАМЕНТАЛЬНІ ПРИНЦИПИ ФУНКЦІОНУВАННЯ

1.1. Еволюційний розвиток та архітектурні особливості великих мовних моделей

Проблема створення штучних систем, здатних розуміти та генерувати природну мову, залишається однією з найбільш амбітних задач штучного інтелекту. З початку 21-го століття науковці активно досліджують питання, чи можливо побудувати мовні моделі, які не просто імітуватимуть людську мову, а й демонструватимуть справжнє розуміння контексту та здатність до логічного мислення [49].

Питанням розробки великих мовних моделей присвячено численні наукові дослідження. Зокрема, цією проблемою займалися такі вчені, як А. Радфорд [45], Дж. Аткинсон [13], Т. Браун [15], К. Такур та інші.

Отже, під поняттям *Велика Мовна Модель* (англ. “Large Language Model”) будемо розуміти *тип алгоритму штучного інтелекту (ШІ), який використовує методи глибокого навчання та величезні набори даних для розуміння, узагальнення, створення та передбачення нового вмісту* [6]. Ефективність Великої Мовної Моделі (ВММ) залежить від кількості та якості даних, які використовувалися для тренування моделі. Статистика свідчить, що оптимальним варіантом є близько сто мільярдів даних для тренування моделі, яка пізніше може виконувати комплексні завдання [13].

Варто підкреслити, мова є одним з основних способів комунікації для людей, вона надає семантику, слова і граматику для вираження ідей і понять. Таку ж роль відіграють ВММ у світі штучного інтелекту, будучи основою генерації текстів. *Штучний інтелект* (англ. “Artificial Intelligence”) - це *алгоритм, який симулює людську поведінку*, результат роботи такого алгоритму можна назвати розумними [49].

Проаналізуємо детальніше поняття великих мовних моделей, запропоноване М. Відгофом. ВММ - це моделі глибокого навчання, навчені на великих обсягах даних, які мають змогу виконувати завдання пов'язані з

обробкою природної мови. Як підкреслюють результати досліджень, модель навчається розуміти закономірності та зв'язки між словами та фразами. Потім вона використовує цю інформацію для прогнозування ймовірності заданих токенів або їх послідовностей. В процесі дослідження теми, виявлено, що токенами, в контексті ВММ, є слова. Цей процес навчання дозволяє генерувати зв'язний та контекстуально релевантний текст або виконувати інші завдання, пов'язані з людською мовою, такі як відповіді на запитання, переклад з іноземної мови та генерація комп'ютерного коду [58].

Зауважимо, що ідея створення мовних моделей зародилася у 1950-х роках, коли науковці працювали над завданням обробки природної мови (англ. “Natural Language Processing”). Програмісти намагалися створити систему, яка могла б розуміти людське мовлення та ефективно його обробляти [54]. Еволюцію створення великих мовних моделей можна поділити на три основні стадії: символічна, статистична та нейронна [56].

У 1950 році, А. Тюрінг опублікував есе “Обчислювальна техніка та інтелект” у якому, представив тест для дослідження машинного інтелекту. Суть тесту полягає в завданні, де машина у ході текстової розмови, повинна переконати інтерв'юера-людину, що він чи вона є людиною. У своїй роботі А. Тюрінг спирався на теорію, яка трактувала, якщо машині вдалося розпізнати, чи спілкується вона з машиною, чи з людиною, вона вважається розумною. На думку деяких дослідників, це стало символічним початком розвитку сучасного штучного інтелекту та символічної ери обробки природної мови на основі системи правил [56].

У ході дослідження, було виявлено, що у 1990-х роках, відбувся величезний поворотний момент розвитку штучного інтелекту, коли статистичні моделі замінили правила. Моделі були створені на основі аналізу великих мовних корпусів. Як показує аналіз літератури [13, 56, 58], однією з перших була прихована марковська модель (англ. “Hidden Markov Models”), вона дозволила передбачати лінгвістичні явища на основі попереднього контексту. Наприкінці десятиліття, нові методи машинного навчання, такі як векторні

моделі та нейронні мережі, просунули розвиток галузі класифікації лінгвістичних явищ [13].

Варто згадати інноваційну працю науковців з Google, які у 2017 році представили архітектуру трансформера [37]. Ця робота запропонувала механізм уваги (англ. “attention mechanism”), який дозволив моделям ефективно обробляти послідовності різної довжини та виявляти складні залежності в тексті. Як зазначають дослідники, саме ця архітектура стала фундаментом для всіх сучасних великих мовних моделей [62].

Не менш важливою є робота команди OpenAI, яка у 2018 році представила модель GPT (Generative Pre-trained Transformer) [45]. Ця модель продемонструвала ефективність підходу попереднього навчання на великих текстових корпусах з подальшим тонким налаштуванням для виконання конкретних задач. На відміну від попередніх систем обробки мови, GPT виявилась здатною генерувати зв'язний текст, підтримувати контекст у довгих діалогах та адаптуватися до різних мовних завдань без значних модифікацій архітектури.

Вчені стверджують, що ера нейронної обробки природної мови розпочалася на початку 21 століття і триває донині [6]. Цей метод характеризується використанням штучних нейронних мереж, які імітують структуру людського мозку та дозволяють ефективно обробляти мову за допомогою глибокого навчання. Проте, штучні нейронні мережі, працювали лише з анотованими тренувальними даними. Здобуті нами дані свідчать, що потреба у підготовці даних зникла з появою генеративного штучного інтелекту, який може самостійно навчатися на великих наборах даних та генерує новий контент на основі запитів користувачів, таким чином, що його складно відрізнити від людського мовлення [13].

Отже, ключовим етапом сучасного розвитку мовних моделей стала поява моделі GPT-3, яка продемонструвала здатність виконувати різноманітні завдання, без спеціального попереднього налаштування. [42] Це було початком розвитку контекстного навчання, де модель навчається з вхідного запиту

користувача. Варто зауважити, що великим поворотним моментом став випуск BMM ChatGPT OpenAI наприкінці 2022 року, що привернуло увагу всього світу до мовних моделей і відкрило потенціал використання моделей не лише в технічних галузях, а й у повсякденному житті. Відтоді сфера спостерігає швидкий розвиток додатків і методів налаштування систем, які дозволяють ефективно впроваджувати великі мовні моделі на практиці [8, 42].

Зазначимо, що існує три основні типи трансформерних архітектур: *GPT* (лише декодер) для генерації тексту, *BERT* (лише кодер) для глибокого розуміння контексту та *T5* (повна encoder-decoder структура) для універсального виконання завдань.

Сучасні BMM прогнозують токен або послідовність токенів. Під поняттям *токен* будемо розуміти мовну одиницю, яка може бути словом, морфемою або одним символом. Науковці погоджуються, що BMM є кращими за своїх попередників з наступних причин: вони мають набагато більше параметрів та збирають набагато більше контексту, ніж рекурентні моделі. В процесі дослідження ми переконались, що найпоширенішою архітектурною для створення мовних моделей є трансформер [37].

Розглянемо визначення *трансформера*, запропоноване Ю. Хваном - це *архітектура мовної моделі, яка складається з кодера та декодера* (Див. Рис.1.1) [62]. Кодер обробляє вхідний текст і перетворює його на проміжне представлення. Під проміжним представленням мається на увазі вектор, тобто список чисел, які нейронна мережа розуміє та може зберігати. Декодер, навпаки, генерує вихідний текст на основі проміжного представлення кодера. Також, декодер має *самоувагу* - спосіб розглядати всі наступні слова та визначити, які з них, допоможуть зрозуміти контекст [37].

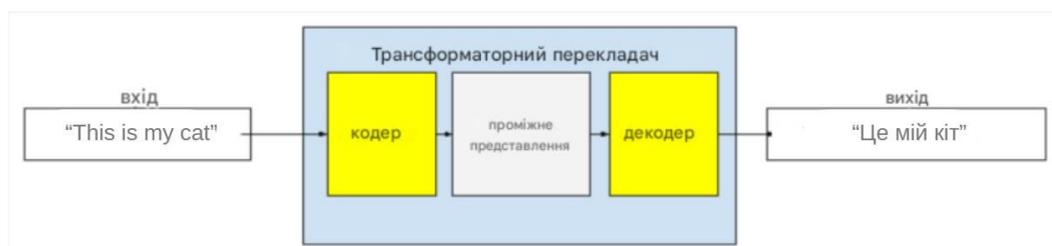


Рис. 1.1. Повна архітектура трансформера

Розглянемо приклад речення: "Інопланетянин приземлився на Землі, тому що йому потрібно було сховатися на планеті". Щоб зрозуміти, що мається на увазі під "йому", ми повертаємося поглядом до "Інопланетянин". Те ж саме робить самоувага, вона відповідає на запитання: "на які інші слова мені потрібно звернути увагу, щоб краще зрозуміти це слово?" [49, 62]. Таким чином механізм самоуваги визначає відношення кожного сусіднього слова до займенника "йому". Деякі механізми самоуваги є двосторонніми, тобто обчислюють бали доречності як для токенів, що передують слову, на якому зосереджується увага, так і токенів, що слідують за ним [37]. Також якщо ми маскуємо слово "йому" і просимо мережу вгадати його, присутність слова "інопланетянин" робить більш імовірним вибір "йому" замість "ій".

Аналіз літератури підтверджує [37, 62], що кожен шар самоуваги складається з кількох голів самоуваги. Різні голови вивчають різні зв'язки між кожним словом, наприклад, граматичні або семантичні зв'язки. У прикладі, розглянутому вище, шар самоуваги зосередився на визначенні того, до якого іменника відноситься займенник "йому".

М. Рідл зазначає, механізм самоуваги складається з трьох етапів [49]:

- Для кожного слова створюються три різні представлення: запит - "Що я шукаю?"; ключ - "Що я можу запропонувати?"; значення - "Яку інформацію я маю?"
- Запит кожного слова порівнюється з ключем всіх інших слів, щоб визначити, наскільки вони релевантні одне одному. У результаті цього етапу, створюється матриця оцінок уваги.
- Значення всіх слів підсумовуються з результатами матриці, отриманими на попередньому кроці. У результаті кожне слово має значення, яке складається з його власного значення і значень пов'язаних з ним слів.

Припустимо, ми обробляємо слово "земля" в реченні про "інопланетянина" та "планету". Самоувага може змішати представлення "землі" з представленням "планети", створюючи нове уявне представлення, яке містить інформацію обох слів. Це робить передбачення більш точними, оскільки

декодер бачить не просто окреме слово, а його контекстуальне представлення [4, 49].

Отже, ми можемо зробити висновок, що трансформер має наступні переваги, на відміну від інших типів архітектур (RNN, LSTM): обробляє всі слова одночасно, а не послідовно; легко зв'язує слова на великій відстані в тексті; ефективно масштабується до мільярдів параметрів [62].

Здобуті нами дані свідчать, що на основі архітектури трансформера було створено різні варіанти моделей, кожен з яких оптимізований для специфічних завдань. GPT використовує лише decoder-частину для генерації тексту, BERT застосовує encoder для глибокого розуміння контексту, а T5 зберігає повну encoder-decoder структуру. Ці моделі є основою для сучасних ВММ, які використовуються в ChatGPT, Google Gemini та інших системах [15, 28].

Розглянемо модель створену лише з декодером, GPT. Ця модель була створена та представлена OpenAI у 2018 році, стала першою відомою успішною реалізацією архітектури з декодером. Такий тип побудови дозволяє моделі бути особливо ефективною для задач з генерацією тексту. Модель навчається передбачати наступне слово в послідовності, що дозволяє їй генерувати зв'язний та контекстуально релевантний текст [15].

В ряді досліджень відмічено, що однією з особливостей GPT є використання *однонаправленої самоуваги*, це тип самоуваги, де кожен токен розглядає лише попередні токени, а не наступні. Такий тип архітектури можна досягти за допомогою механізму маскування, який блокує доступ до майбутніх токенів, під час обчислення попередніх. Таким чином, модель послідовно генерує текст зліва направо [15, 21].

Підкреслимо, що GPT проходить двоетапне навчання, спершу модель проходить попереднє некероване навчання (unsupervised pre-training), яке включає тренування моделі на великому корпусі тексту. На цьому етапі модель вчиться моделювати мовну структуру. Наступним кроком є глибоке навчання (fine-tuning), коли модель тренується виконувати специфічні задачі на невеликому наборі даних. Отже, можна стверджувати, що такий підхід дозволяє

отримати кращі результати під час виконання завдань, ніж з моделями навченими з нуля. OpenAI продовжує будувати нові моделі, збільшуючи розмір моделі та обсяг навчальних даних, що призводить до більш якісних результатів [45].

Прикладом іншого типу архітектури є BERT (Bidirectional Encoder Representations from Transformers), розроблений Google у 2018 році. Для створення моделі, використали лише encoder-частину трансформера [21]. На відміну від GPT, BERT використовує двонаправлену (англ. “bidirectional”) самоувагу, що дозволяє кожному токеноу звертати увагу на всі інші токени в послідовності — як попередні, так і наступні. Такий підхід допомагає глибше зрозуміти контекст і семантичні зв’язки у тексті.

Дж. Делвін зазначає, для навчання BERT використовують два методи: Masked Language Modeling (MLM) та Next Sentence Prediction (NSP). У першому методі, маскуються 15% токенів у вхідній послідовності, і модель повинна передбачити ці замасковані токени на основі контексту [21]. У другому методі, модель вчиться розуміти зв’язки між реченням, передбачаючи, чи є одне речення логічним продовженням іншого.

Отже ми можемо зробити висновок, що BERT є ефективним для задач, де необхідне розуміння тексту, це можуть бути: класифікація, відповіді на запитання, сентимент аналіз. Модель створює контекстуальні зв’язки, які можна використовувати як вхідні дані для завдання з мінімальною адаптацією архітектури. BERT завдячує своїй здатності, краще захоплювати семантику слів, за допомогою контексту, двонаправленій природі [21].

Після успіху BERT, науковці випустили покращені версії, включаючи RoBERTa, ALBERT, DistilBERT та інші, кожен з яких оптимізує різні аспекти оригінальної архітектури — від ефективності навчання до розміру моделі [66]. Ці зміни показали, що encoder-only архітектура залишається особливо ефективною для виконання задач з розуміння тексту, особливо коли важливо проаналізувати глибокі семантичні зв’язки.

T5 (Text-to-Text Transfer Transformer), представлений Google Research у 2019 році, є прикладом повної архітектури трансформера, що включає кодер і декодер [47]. Ключова ідея T5 полягає в тому, що всі задачі з обробки природного мовлення можна переформулювати як задачі перетворення тексту в текст (англ. “text-to-text”), що дозволяє використовувати одну й ту саму архітектуру та підхід до навчання для різноманітних завдань.

Згідно з дослідженням К. Раффель, T5 encoder обробляє вхідний текст і створює його контекстуальне представлення, використовуючи двонаправлену самоувагу, подібно до BERT. Decoder потім генерує вихідний текст на основі цього представлення, використовуючи однонаправлену самоувагу та перехресну увагу (cross-attention) до виходу кодера [47]. На думку дослідників така архітектура робить T5 особливо гнучким і придатним для широкого спектру задач — від перекладу до відповідей на запитання.

T5 навчається на масивному наборі даних C4 (Colossal Clean Crawled Corpus) за методом, де випадково маскуються послідовні фрагменти тексту, а модель повинна їх відновити [47]. Дослідники систематично вивчали різні аспекти навчання моделей, включаючи розміри моделей, стратегії попереднього навчання, розміри набору даних та підходи до налаштування. Це дослідження виявило, що збільшення масштабу як моделі, так і даних послідовно покращує результати виконання завдань.

Аналіз зазначеного вище дозволив нам констатувати, що архітектура T5 робить модель корисною для виконання множинних задач, без необхідності додаткових налаштувань. Структура з кодером і декодером є ефективною для завдань, де розмір вхідних даних значно відрізняється від розміру вихідних даних. Це можуть бути такі задачі як, підбиття підсумків або генерація розгорнутої відповіді [48].

В процесі дослідження ми переконались, що розглянуті три архітектури трансформерів — GPT, BERT та T5 — демонструють різні підходи до використання базової трансформер-архітектури залежно від специфіки задач обробки природної мови. Вибір архітектури залежить від конкретної задачі:

GPT є корисним для генерації відповідей та завершення задач, BERT - для розуміння та аналізу тексту, а T5 є універсальним варіантом. Кожна з цих архітектур внесла значний вклад у розвиток сучасних систем обробки природної мови та продовжує еволюціонувати через численні модифікації та покращення.

Розглянемо, як моделі обробляють текст, а саме: токенізацію, створення ембеддингів, кодування та декодування. Згідно з визначенням С. Метцгера, *токенізація - це процес розбиття тексту на менші одиниці, які називаються токенами* [51]. Ці одиниці можуть бути словами, фразами, підсловами, розділовими знаками або символами. За даними OpenAI, один токен становить приблизно чотири символи і $\frac{3}{4}$ слова в англійській мові, тобто, 100 токенів приблизно дорівнюють 75 словам [42].

Здобуті нами дані свідчать, що комп'ютерні машини не розуміють слів та можуть працювати лише з числами. Токенізація є підготовчим етапом для аналізу текстових даних, роблячи інформацію структурованою та готовою до обробки мовними моделями. Окрім цього, існує декілька типів токенізації: токенізація слів, токенізація символів, токенізація підслів, токенізація речень та N-грамна токенізація [51].

С. Метцгер підкреслює, що токенізація слів є найпоширенішим методом, текст розділяється на окремі слова [51]. Такий метод є ефективним для мов з чітким розмежуванням слів, як англійська. Розглянемо приклад, вхідними даними є таке речення: ["Машинне навчання є складним та унікальним предметом"]. Вихідними даними, після токенізації, є: ["Машина", "навчання", "є", "складно", "та", "унікально", "предмет"] [29].

Під час токенізації символів, дані розділяються та перетворюються на послідовність окремих символів. Такий метод корисний під час виконання завдань, які потребують детального аналізу, наприклад виправлення орфографії [29]. Вихідними даними у такому випадку є наступні токени: [{"М", "а", "ш", "и", "н", "н", "е", " ", "н", "а", "в", "ч", "а", "н", "н", "я", " ", "є", " ", "с", "к", "л", "л", "о", "н", "о", "т", "а", "у", "н", "і", "к", "а", "л", "ь", "н", "о", "м", "у", "п", "р", "е", "д", "м", "е", "т", "у"}]

"а", "д", "н", "и", "м", " ", "т", "а", " ", "у", "н", "і", "к", "а", "л", "ь", "н", "и", "м", " ", "п", "р", "е", "д", "м", "е", "т", "о", "м", " ", "я", і тд.].

Токенізація підслів розділяє текст на одиниці, які більші за один символ, але менші за повне слово. К. Даннакоті стверджує, що такий тип токенізації використовується для роботи з морфологічно багатими мовами або рідкісними словами. Розглянемо приклад, вхідні дані: ["Підпорядкування"]. Вихідні дані: ["Під", "поряд", "кув", "ання"] [22]. Такий тип токенізації допомагає обробляти слова поза словниковим запасом у завданнях з обробки природнього мовлення та мов, які утворюють слова шляхом поєднання менших одиниць.

Зазначимо, що токенізація речень також є поширеним методом, який використовується для поділу абзаців або великого набору речень на окремі речення як токени [29]. Це корисно для завдань, що потребують аналізу або обробки окремих речень. Вхідні дані: ["Штучний інтелект – це нова технологія. Машинне навчання захоплює. Комп'ютерний зір обробляє зображення."]. Вихідні дані: ["Штучний інтелект – це нова технологія.", "Машинне навчання захоплює.", "Комп'ютерний зір обробляє зображення."] [22].

Розглянемо також основні засади N-грамної токенізації, яка розділяє слова на фрагменти даних фіксованого розміру (розмір = n). Вхідні дані перед токенізацією: ["Машинне навчання потужне"]. Вихідні дані після токенізації біграмами: [('Машина', 'навчання'), ('навчання', 'е'), ('е', 'потужний')] [22].

Спираючись на дослідження згаданих вище науковців, ми визначаємо, що токенізація є важливим кроком в обробці тексту та обробці природної мови з кількох причин. Токенізація зменшує розмір необробленого тексту, що призводить до легкого та ефективного статистичного та обчислювального аналізу. Текстові дані можна представити у числовому форматі для алгоритмічного розуміння, використовуючи токени як ознаки в моделях машинного навчання. Токенізація є важливою для індексації та пошуку в системах, які ефективно зберігають та витягують інформацію на основі слів або фраз.

Однією з поширених проблем токенизації, яку описує Б. Пенг, є обробка орфографічних помилок. Наприклад, якщо корпус містить слово з помилкою замість правильного слова, модель може розглядати його як слово поза словником, що може значно знизити продуктивність моделі [41, 43].

Після токенизації, кожен токен перетворюється на числове представлення, яке може обробляти модель. Цей процес називається створенням ембеддингів. К. Даннакоті пропонує таке визначення терміну *ембеддинг* - це процес представлення токенів як неперервних векторів у багатовимірному просторі, де схожі токени мають схожі векторні представлення [22]. Ембеддинги допомагають моделям захоплювати семантичне значення токенів та їхні зв'язки у вхідному тексті.

Наприклад, слово "король" може бути представлено вектором [0.2, -0.5, 0.8, 0.1, ...], де кожне число кодує певний аспект значення слова. Слова зі схожим значенням матимуть схожі вектори — "король" і "королева" будуть ближче одне до одного в векторному просторі, ніж "король" і "стіл" [22].

Першим відомим підходом до ембеддингу було one-hot encoding, кожне слово представлялося вектором довжини словника. У межах цього вектора стояла одиниця, яка позначала позицію слова у словнику, а решта позицій були нулі. Проте, такий метод виявився неефективним, оскільки створював ембеддинги великої довжини, які не передавали семантичні значення і подібність слів [49, 9].

Новою ерою у створенні ембеддингів стала публікація статті "Efficient Estimation of Word Representations in Vector Space" у 2013 році. У своїй роботі Томас Міколов запропонував представляти кожен токен як вектор. Це дозволило порівнювати слова та знаходити подібності у семантичних значеннях [36, 55].

Ембеддинги мають математичні властивості. Наприклад, можна виконувати операції з векторами: вектор("король") - вектор("чоловік") + вектор("жінка") = вектор("королева") [36]. Цей приклад підтверджує, що ембеддинги справді захоплюють смислові відношення між концепціями.

Розглянемо детальніше функції та завдання кодера і декодера, які були зазначені раніше, як складові архітектури трансформера. Кодер виконує функцію стиснення великої кількості інформації у компактне представлення. Замість того, щоб безпосередньо з'єднати 50,000 входів (по одному для кожного можливого слова) з 50,000 виходів, кодер спочатку перетворює вхідні дані у менший набір чисел — наприклад, 256 значень замість 50,000 [49]. Хорошим порівнянням є стиснення файлів, коли користувач архівує документ, він отримує менший файл, який більше не можна прочитати до розархівування. Тобто кодер створює стиснуте представлення, яке зберігає найважливішу інформацію про слово та його контекст.

Кодер навчається групувати схожі слова разом у цьому стисненому просторі. Слова "король", "королева", "принцеса" матимуть схожі 256-вимірні представлення, оскільки всі вони пов'язані з концепцією "монархії" [49, 55].

Декодер має зворотну функцію: він приймає стиснене 256-вимірне представлення від кодера і перетворює його назад у 50,000-вимірний простір, де кожне значення відповідає можливому слову у словнику.

Серед недоліків трансформерів науковці виділяють такі пункти: тривале навчання, що перешкоджає швидкому експериментуванню; обмеження довжини контексту, рекомендується використовувати до 512 токенів, при стандартному апаратному забезпеченні; високі обчислювальні вимоги, що робить навчання дорогим і навіть є одним з чинників високого вуглецевого сліду [64, 34].

Однією з переваг великих мовних моделей, яку виділяють дослідники є величезний обсяг інформації, на якому навчаються моделі. Це включає книги, блоги, сайти новин, дискусії на Reddit та навіть розмови у соціальних мережах. Такий підхід допомагає виконувати різноманітні завдання з різних доменів [49]. Серед інших переваг виділяють функцію самонавчання, під час навчання, модель самостійно коригує параметри, якщо не вдається передбачити правильне слово [28].

Науковці підкреслюють і недоліки великих мовних моделей, такі як: відсутність знання істини, модель лише передбачає слово, не розуміючи чи

правдивим є твердження [49]; галюцинації часто трапляються у моделей, коли вони передбачають неправильні слова; великі мовні моделі є авторегресивними, коли модель робить помилку, ця помилка повторюється під час передбачення кожного наступного слова, збільшуючи відсоток хибних передбачень; великі мовні моделі не ведуть діалоги, машина не пам'ятає попередні вхідні дані, а оперує лише з тим, що тримала від користувача і має збережене у пам'яті [56].

В процесі дослідження проблеми виявлено, що принцип роботи великих мовних моделей складається з декількох етапів: токенизація, створення ембеддингів та застосування механізму самоуваги для аналізу контекстуальних зв'язків між словами.

Токенизація розбиває текст на керовані одиниці, ембеддинги перетворюють ці одиниці в багатовимірні вектори, які захоплюють семантичне значення, а архітектура кодера та декодера ефективно стискає та відновлює інформацію. Самоувага є ключовою інновацією, яка дозволяє моделям автоматично визначати, які слова в послідовності пов'язані між собою.

Підсумовуючи сказане вище, зауважимо, що ефективність великих мовних моделей залежить від комбінації ефективного механізму трансформера та навчання на величезних обсягах різноманітних текстових даних з інтернету. Модель навчається розпізнавати та відтворювати статистичні патерни в мові, що дозволяє їй генерувати текст, який виглядає природним та контекстуально релевантним.

Розуміння цих фундаментальних принципів роботи є особливо важливим для відповідального використання великих мовних моделей, усвідомлення їхніх обмежень та оцінки якості їхніх відповідей. Ці моделі є потужними інструментами, але вони залишаються інструментами, які виконують складне статистичне передбачення, а не демонструють справжнє розуміння чи мислення.

1.2. Сучасні методологічні підходи до навчання великих мовних моделей

Аналіз відповідної джерельної бази [10, 21, 7] дозволив констатувати, що ефективність великих мовних моделей залежить не лише від типу архітектури, але й від методології їх навчання. Сучасні підходи до тренування ВММ базуються на багатоетапному процесі, що включає первинне навчання на масштабних корпусах текстів, адаптацію до специфічних завдань та подальше вдосконалення через зворотний зв'язок від користувачів. Еволюція методів навчання моделей почалася з класичного керованого навчання та пройшла такі етапи як, самокероване навчання, яке дозволило масштабувати процес навчання на мільярди параметрів не використовуючи анотації.

Нами встановлено, що у контексті практичного застосування великих мовних моделей, популярною темою є адаптація моделей до конкретних предметних областей. Тонке налаштування всіх параметрів вимагає складні математичні обчислення, що гальмує розвиток та впровадження технологій. Науковці зосередилися на розвитку альтернативних підходів, таких як промпт інженерія, вони дозволяють адаптувати поведінку моделі з мінімальними втратами ресурсів. Ми розглянули основні етапи та методології навчання великих мовних моделей, їхні переваги, обмеження та сфери застосування, які є важливими для розуміння процесу створення спеціалізованих діалогових систем з контекстною підтримкою.

В процесі дослідження проблеми нами було виявлено, що попереднє навчання великих мовних моделей є початковим кроком, на якому моделі, такі як GPT-3 або BERT, навчаються з величезних обсягів тексту, таких як книги та веб-сайти, не зосереджуючись на конкретному завданні. Цей процес використовує самоконтроль, тобто модель навчається, прогножуючи відсутні частини тексту, такі як наступне слово або приховане слово, а не потребуючи позначених даних [10].

Для попереднього навчання LLM застосовуються так звані Data Mixtures великі та різноманітні корпуси текстів, що охоплюють широкий спектр стилів,

тем і форматів. Основні джерела таких даних включають:– GitHub для коду та технічного контенту;– Wikipedia для загальних знань;– ArXiv для наукових досліджень;– Книги для довгих наративів та структурованих текстів;– StackExchange для формату запитань і відповідей;– CommonCrawl та C4 для загального контенту веб-мережі [10].

Підкреслимо, що після попереднього навчання ці моделі можна модифікувати за допомогою тонкого налаштування для виконання конкретних завдань, таких як відповіді на запитання або переклад мов, з меншим додатковим навчанням. Це робить їх універсальними та ефективними.

Розглянемо детальніше завдання, які виконує модель під час попереднього навчання, передбачаючи, що буде далі, або заповнюючи пропуски, модель вивчає, як працює мова. Наприклад, якщо вона бачить «Кіт __», вона може передбачити «спить» на основі вивчених шаблонів. Це допомагає їй зрозуміти структуру речень, значення слів і навіть деякі світові знання, наприклад, знати, що коти можуть спати [21].

Науковці виділяють два основних способи попереднього навчання: Авторегресивні моделі (наприклад, GPT): прогнозування наступного слова, навчання зліва направо. Моделі маскованої мови (наприклад, BERT): прогнозування прихованих слів, враховуючи все речення. Цей процес створює основу, яку можна адаптувати пізніше, заощаджуючи час і дані для конкретних завдань [21].

Попереднє навчання, також відоме як трансферне використовує великомасштабні, немарковані дані, що робить навчання економічно ефективним та масштабованим порівняно з навчанням з нуля для кожного завдання [16].

Іншим методом навчання великих мовних моделей є тонке налаштування. Звернемося до визначення *тонкого налаштування*, запропонованого Міністерством цифрової трансформації України, це *процес адаптації попередньо навченої моделі машинного навчання до специфічного завдання чи*

предметної сфери шляхом додаткового навчання на невеликому наборі цільових даних [8].

Підхід тонкого налаштування переносить вивчені шаблони та функції попередньо навченої моделі на нові завдання, покращуючи продуктивність та зменшуючи потреби в навчальних даних. Згідно з дослідженнями, воно стало популярним у обробці природної мови для таких завдань, як класифікація тексту, аналіз настроїв та відповіді на запитання. Серед типів тонкого налаштування, науковці виділяють наступні: неконтрольоване тонке налаштування, контрольоване тонке налаштування та тонке налаштування за допомогою промптів [23, 40].

Нами виділено, що неконтрольоване тонке налаштування не використовує анотовані дані, натомість модель тренується на великому корпусі немаркованого тексту з цільової області, що покращує його розуміння мови. Цей підхід корисний для нових областей, таких як правова чи медична галузі, але є менш точним для певних завдань, таких як класифікація чи узагальнення [18, 40].

Натомість контрольоване тонке налаштування відбувається на анотованих даних, адаптованих до цільового завдання. Прикладом завдання яке вимагає контрольоване тонке налаштування є класифікація тексту в бізнес-контексті. Для такого налаштування використовують набір фрагментів тексту з мітками(анотаціями) класу. Хоча цей метод ефективний, він вимагає значних маркованих даних, отримання яких може бути дорогим та трудомістким [40].

Зазначимо, що тонке налаштування за допомогою промптів спирається на надання розробниками інструкцій для моделі природною мовою. Такий метод зменшує потребу у величезній кількості анотованих даних, але залежить від якості промптів[18, 40, 60].

Порівнюючи попереднє навчання та тонке налаштування варто зазначити, що перший метод включає ретельне навчання на величезних обсягах немаркованих даних для формування загальних лінгвістичних знань, тоді як другий підхід адаптує попередньо навчені моделі до спеціалізованих завдань з

використанням менших, маркованих наборів даних, зосереджуючись на покращенні продуктивності конкретних завдань [40].

Д. Зіглер, дослідник та розробник у Anthropic, американської компанії, яка спеціалізується на дослідженнях безпеки ШІ та розробці великих мовних моделей, виділяє зменшення вимог до даних, як одну з переваг тонкого налаштування. Окрім цього він зазначає, що тонко налаштовані моделі краще адаптуються до виконання широкого кола завдань та час навчання є значно, ніж у випадку з попереднім навчання [60, 68].

Серед недоліків тонкого налаштування, Дж. Торре, засновник Helicone, додатку для моніторингу та налагодження програм штучного інтелекту, підкреслює такі пункти: обмежена сфера застосування, потенційна деградація продуктивності, вартість створення та обслуговування і ризик застарівання даних [57].

Науковці виділяють ще один метод навчання ВММ. Навчання з підкріпленням через людський зворотний зв'язок (англ. “Reinforcement Learning from Human Feedback (RLHF)”) стало ключовим методом інтеграції сигналів людських уподобань у методах машинного навчання [38]. Зокрема, RLHF став стандартним компонентом у пост-навчальному процесі для великих мовних моделей, який служить для узгодження результатів цих моделей з людськими цінностями, такими як корисність, нешкідливість та чесність [67].

В процесі дослідження ми переконались, що, серед відомих мовних моделей, які використовують RLHF - ChatGPT, Claude та Gemini. Значний інтерес до навчання з підкріпленням через людський зв'язок виник з появою ChatGPT, проте метод залишається менш дослідженим, аніж контрольоване налаштування.

Ключовою особливістю RLHF є використання зворотнього зв'язку замість чітких оцінок винагороди на відміну від класичного навчання з підкріпленням. Такий підхід є більш ефективним, оскільки людям, які оцінюють згенеровані дані, складно надавати точні абсолютні оцінки, натомість вони з легкістю можуть вибрати кращий варіант з двох запропонованих відповідей [38].

Лонг Оуян, дослідник зі Стенфорда, виділяє такі компоненти, необхідні для навчання з підкріпленням через людський зворотний зв'язок: попередньо навчена мовна модель, набір промптів, на основі яких модель генеруватиме узгоджені результати та команда навчених аннотаторів. Науковець підкреслює три етапи RLHF:

1. Збір демонстраційних даних, на цьому етапі аннотатори надають демонстраційні (очікувані) результати у відповідь на попередньо підготовлені вхідні дані. Далі, дослідники налаштовують попередньо навчену модель на цих даних, використовуючи метод контрольованого навчання [38].

2. Збір даних порівняння та навчання моделі винагороди. Аннотатори переглядають вихідні дані моделі та обирають один варіант серед декількох запропонованих. Дослідники збирають набір даних порівнянь та навчають модель винагороди для прогнозування вихідного сигналу, який вчиться прогнозувати варіанти, які більше імпонують людям [38].

3. Оптимізація політики відносно моделі винагороди за допомогою проксимальної оптимізації політики. Проксимальна оптимізація політики це алгоритм навчання з підкріпленням (RL), який використовується для оптимізації політик стабільним і ефективним способом [7]. Модель продовжує генерувати відповіді, які оцінюються моделлю винагород. На основі оцінок модель вчиться генерувати кращі вихідні дані.

Дослідники стверджують, що другий та третій кроки можуть повторюватися безперервно, що допомагає зібрати більше даних та оптимізувати роботу моделі під переваги користувачів, які також можуть змінюватися з часом [38].

Р. Женг, науковець Фуданського університету, у своїй праці виділив такі переваги RLHF: - зворотній зв'язок від людей забезпечує модель даними для генерування більш точних і контекстуально коректних відповідей, - вихідні результати є більш гнучкими завдяки даним від аннотаторів, - збір відгуків від людей з різними точками зору допомагає генерувати результати, які враховують різні запити користувачів [67].

Щодо недоліків такого методу навчання, Р. Женг зазначає: - високу вартість для найму команди аннотаторів, - суб'єктивна оцінка з якою може не погодитися більшість користувачів [67].

Розглянемо ще один метод навчання моделей. Параметро-ефективне тонке налаштування (англ. “Parameter-Efficient Fine-Tuning (PEFT)”) це набір методів, який коригує лише частину параметрів у ВММ для економії ресурсів. Ключова особливість PEFT полягає в тому, що оновлюється лише невелика кількість додаткових параметрів або підмножина попередньо навчених параметрів, що дозволяє зберегти знання, здобуті моделлю під час попереднього навчання, та знизити ризик забування даних [63].

З. Хан, дослідник Північно-східного університету у Бостоні, класифікує п'ять основних категорій PEFT: додаткове налаштування (англ. “additive fine-tuning”), часткове налаштування (англ. “partial fine-tuning”), репараметризоване налаштування (англ. “reparameterized fine-tuning”), гібридне налаштування (англ. “hybrid fine-tuning”) та уніфіковане налаштування (англ. “unified fine-tuning”) [30].

Експериментальні дослідження підтверджують, що більшість методів PEFT значно покращують ефективність параметрів і досягають порівнянної або навіть кращої продуктивності порівняно з повним налаштуванням, при цьому суттєво зменшуючи використання пам'яті GPU.

Дослідники виділяють низку переваг PEFT у порівнянні з класичними методами навчання ВММ. Метод оновлює лише невелику підмножину параметрів, що значно зменшує вимоги до обчислювальних ресурсів та пам'яті, дозволяючи налаштовувати моделі навіть на пристроях з обмеженим апаратним забезпеченням. PEFT прискорює процес навчання порівняно з повним налаштуванням, що забезпечує швидші ітерації розробки та тестування моделей. Оновлення меншої кількості параметрів знижує ризик перенавчання, підвищуючи здатність моделі ефективно працювати з новими даними [17, 30].

Щодо недоліків, PEFT найефективніше працює при адаптації до завдань, які є спорідненими з знаннями попередньо навченої моделі, та демонструє

складнощі при роботі з абсолютно новими предметними областями. Деякі техніки, зокрема LoRA, вносять додаткові обчислення під час виведення, що призводить до незначного збільшення латентності. Крім того, вибір оптимального методу PEFT для конкретного завдання вимагає експериментування та тестування різних підходів, що ускладнює процес оптимізації моделі [17].

Таким чином, аналіз сучасних методів навчання та адаптації великих мовних моделей засвідчує наявність різноманітних підходів, кожен з яких має специфічні переваги та обмеження. Попереднє навчання формує фундаментальні лінгвістичні знання моделі на основі масштабних корпусів текстів, тоді як тонке налаштування забезпечує адаптацію до спеціалізованих завдань з використанням цільових даних. Навчання з підкріпленням через людський зворотний зв'язок дозволяє узгодити поведінку моделі з людськими цінностями та перевагами, що є критично важливим для створення надійних діалогових систем. PEFT постає як найбільш економічно ефективне рішення, що суттєво зменшує вимоги до обчислювальних ресурсів при збереженні високої продуктивності моделі. Комплексне розуміння цих методологій є необхідною передумовою для ефективного розробки та впровадження спеціалізованих мовних моделей у практичних застосуваннях, зокрема при створенні чат-ботів.

1.3. Методи оптимізації та адаптації великих мовних моделей для вирішення спеціалізованих задач

Великі мовні моделі демонструють вражаючі можливості у широкому спектрі задач обробки природної мови, проте їх безпосереднє застосування для вирішення спеціалізованих проблем часто стикається з рядом викликів. До таких викликів належать обмеження контексту, недостатня точність у вузькоспеціалізованих доменах та проблема актуальності знань моделі. Для подолання цих обмежень було розроблено різноманітні методи оптимізації та

адаптації, які дозволяють ефективно налаштовувати ВММ під конкретні завдання без необхідності повного перенавчання моделі з нуля.

Сучасна практика застосування ВММ все частіше спирається на комбінування декількох методів одночасно, що дозволяє досягти оптимального балансу між якістю результатів, швидкістю обробки запитів, споживанням ресурсів та можливістю адаптації до специфічних вимог предметної області. Ми розглянули основні методи оптимізації великих мовних моделей для виконання завдань різних доменів.

Здобуті нами дані свідчать, що одним з найпопулярніших способів адаптації ВММ є промпт інженерія. Знаючи як ефективно структурувати та оцінювати промпти під час використання ВММ, користувачі отримують кращі результати. Промпт інженерія це систематичний процес проектування та оптимізації вхідних запитів (промптів) для великих мовних моделей (ВММ) з метою отримання точних, релевантних та узгоджених результатів [50].

Для розуміння терміну промпт скористаємося визначенням запропонованим С. Шлюхоффом, вхідні дані для ВММ, що використовуються для керування її вихідними даними [50]. Промптами можуть бути текст, зображення, звук чи інші медіа-формати. Промпт складається з таких компонентів: інструкція, приклади, формат виводу, роль, додаткова інформація [17].

Інструкція пояснює основну мету промпту, наприклад: “Напиши звіт для проекту”. Приклади дементрують очікувані результати виконання завдання. Промпт інженерія поділяється на три типи за кількістю прикладів у одному запиті: Zero-Shot - без прикладів, One-Shot - один приклад, Few-Shot - кілька прикладів. Форматування вихідних даних додає специфікацію очікуваного формату відповіді, наприклад: формат та структура файлу. Дослідниками було виявлено, що додавання певної ролі, покращує стиль та якість відповідей. Наприклад: “Ти історик з глибокими знаннями подій 20 століття, проаналізуй політичну ситуацію у Франції у 1950-х роках” [50]

Використання методу промпт інженерії поділяється на три повторювані кроки: 1. Виконання інференсу на даних [17]. Під терміном інференс розумітимемо процес отримання відповіді (передбачення) від навченої моделі на основі вхідних даних [50]. 2. Оцінка результатів. 3. Модифікація промпту. Така послідовність кроків повторюється, поки розробники не досягнуть бажаних результатів.

П. Саху наголошує на необхідності виконання всіх етапів розробки промптів. Спершу науковці досліджують проблему та специфіку домену та аналізують наявні дані для визначення критерії успіху. Наступним кроком є тестування найпростіших підходів. П. Саху рекомендує використання Zero-Shot промптинг та оцінку результатів після. Далі варто оптимізувати промпт використовуючи техніку Few-Shot промптинг, додаючи приклади до запити. Кількість та релевантність прикладів грають ключову роль для якості майбутніх результатів. Останнім етапом є інженерія відповіді, дослідники зазначають форму, можливі обмеження та правила для формування відповіді [50].

Дослідники виділяють також наступні типи промпт інженерії: ланцюжок міркувань, ітеративний промптинг, негативний промптинг, гібридний промптинг, дерево думок та самоузгодження[12, 59].

Ланцюжок міркувань (англ. “Chain-of-Thought Prompting”) це техніка формулювання запитів до ВММ, яка спонукає модель покроково демонструвати процес прийняття рішень або пошуку відповіді на запит [8]. Для прикладу, користувачу потрібно написати відповідь на відгук клієнта, першим запитом до моделі є: “Проаналізуй відгук клієнта та виділи:

1. Загальну тональність (позитивна/негативна/змішана)
2. Позитивні аспекти (список)
3. Негативні аспекти (список)
4. Рівень незадоволення (низький/середній/високий)”.

Отримавши відповідь, користувач надсилає наступний запит: “На основі аналізу відгуку, запропонуй стратегію відповіді:

1. Тон відповіді

2. Які аспекти варто визнати
3. Які компенсації запропонувати
4. Конкретні дії для покращення”.

Останнім промптом у ланцюжку буде: “Напиши персоналізовану відповідь клієнту на відгук, використовуючи: дані з відгуку клієнта та стратегію відповіді”. Таким чином за допомогою набору запитів модель змогла якісно проаналізувати вхідні дані та згенерувати відповідь, які відповідає всім вимогам [92]

Ітеративний промптинг (англ. “Iterative Prompting”) це тип інженерії запитів у якому запит поступово редагується, спрямовуючи ВММ до правильної відповіді. Наприклад, початковим запитом є: “Придумай назву для додатку для медитації”. Отримавши відповідь, користувач редагує запит вказуючи деталі: “Придумай назву для додатку для медитації. Вимоги: коротка, запам'ятовується, українською”. Проаналізувавши відповідь, додаємо фінальні деталі: “Придумай назву для додатку для медитації. Вимоги: коротка, українською, креативна Стиль: як "Headspace" - легко, сучасно”. Таким чином за допомогою повторного надсилання схожих промптів ми отримали бажаний результат [12].

Негативний промптинг (англ. “Negative Prompting”) це тип інженерії запитів у якому зазначаються дії, які ВММ повинна уникати під час виконання завдання. Розглянемо приклад: “Напиши про кішок. Не роби текст занадто довгим. Не використовуй складних слів. Не пиши нудно. Не роби багато абзаців.” [59]. ВММ уникатиме вказаних дій, створюючи більш чіткий результат.

Л. Ванг та К. Чен виділяють також гібридний промптинг (англ. “Hybrid Prompting”) який включає в себе декілька методів. Наприклад, використання негативного та ланцюжкового промптингу та ланцюжка міркувань. “Задача: Визнач, чи варто купувати цей вживаний ноутбук.

Характеристики:

- Ціна: 8000 грн
- Вік: 5 років

- Процесор: Intel Core i5 (7 покоління)
- RAM: 8 GB
- Стан: "як новий"

Інструкції:

1. Проаналізуй кожен параметр крок за кроком
2. Обґрунтуй свої міркування
3. Дай фінальну рекомендацію

НЕ РОБИТИ:

- Не приймай рішення одразу без аналізу
- Не ігноруй вік пристрою
- Не довіряй сліпо опису "як новий"

Дерево думок (англ. "Tree of Thoughts (ToT)") це передовий метод описаний Б. Ченом, який використовує структурований підхід для керівництва ВММ у процесах міркування та генерації відповідей. Він покращує вирішення проблем, досліджуючи кілька шляхів міркування, які називаються «думками». Підхід ToT, наприклад, застосовується до складного математичного вирішення проблем, спонукає модель генерувати різні потенційні рішення та оцінювати їх, а не просто запитувати про рішення [17]. Приклад: "Детальний аналіз розмірів: військові танки проти автомобілів"

1. Розміри та вага:

- Військові танки: зазвичай основні бойові танки (такі як M1 Abrams або T-90) мають ширину...

2. Маневреність та конструкція...

Використайте наведену вище інформацію, щоб відповісти на наступне запитання: Уявіть собі нескінченно широкий вхід, який з більшою ймовірністю проїде через нього, військовий танк чи автомобіль?"

Самоузгодженість (англ. "Self-consistency") є ще одним методом промпт-інженерії, який описує Б. Чен. Цей метод покращує надійність та точність відповідей великих мовних моделей шляхом генерації декількох

варіантів рішення та вибору найбільш поширеного результату. На відміну від стандартного підходу, який вибирає лише одну найбільш ймовірну відповідь, самоузгодженість використовує техніку семплінгу для створення різноманітних шляхів міркування [17, 59].

В ряді досліджень відмічено, що метод самоузгодження складається з трьох етапів: спочатку модель отримує промпт із ланцюжком міркувань (CoT), потім генерує декілька альтернативних рішень використовуючи різні траєкторії міркування, і нарешті генерує результати, обираючи відповідь, що найчастіше зустрічається серед усіх згенерованих варіантів. Цей підхід є ефективним для задач арифметики, логічного міркування та обчислень, оскільки випадкові помилки не повторюються систематично, тоді як правильна логіка призводить до однакової відповіді різними шляхами [17].

Розглянемо приклад: "Задача: У магазині є 24 ноутбуки. Продали 7 ноутбуків вранці та 5 увечері. Потім завезли нову партію з 12 ноутбуків. Скільки ноутбуків зараз у магазині? Згенеруй 5 різних способів вирішення цієї задачі з детальним поясненням кожного кроку. Після цього визнач, яка відповідь зустрічається найчастіше, та обґрунтуй, чому саме вона є правильною."

Лі Ванг підкреслює, що швидке впровадження та економія ресурсів є перевагою використання промпт інженерії. Окрім того, науковець зазначає, що такий метод є дуже гнучким та дозволяє швидко експериментувати з різними завданнями. Дослідник виділяє, обмеження лише початковими даними ВММ, як основний недолік такого методу [59].

Як уже відомо, ВММ отримують свої знання про світ з величезних корпусів тексту, проте, такі знання є доволі обмеженими. По-перше, важко зберегти всі знання з навчального корпусу, особливо для менш поширених та більш специфічних доменів. По-друге, оскільки параметри моделі не можна динамічно оновлювати, ці знання схильні до застарівання з часом. Розширення параметрів моделі призводить до збільшення витрат та є довготривалим процесом. Для вирішення цієї проблеми існує метод інтеграції бази даних з

ВММ. Цей підхід відомий як, генерування доповнене пошуком (англ. “Retrieval Augmented Generation (RAG)”) [8, 32].

Термін генерація доповнена пошуком був вперше запропонований Патріком Льюїсом у 2020 році [33]. Цей метод поєднує попередньо навчений пошуковий пристрій з ВММ. Після появи ВММ, таких як ChatGPT, генеративні мовні моделі стали переважаючими, демонструючи високу продуктивність у завданнях пов'язаних з генерацією мовних послідовностей. Однак моделі все ще стикаються з такими проблемами, як галюцинації оновлення знань та проблеми, пов'язані з даними. Це впливає на їхню надійність, ускладнюючи виконання ВММ для виконання складних завдань, що потребують доступу до величезної кількості знань, таких як відповіді на запитання з відкритою предметною областю [24].

Генерування доповнене пошуком складається з кількох етапів: індексація, пошук та генерація. Етап індексації включає вилучення вхідних даних та перетворення їх з різних форматів, яких як PDF, HTML чи Docx у звичайний текст. Цей крок також включає розділення завантаженого тексту на менші фрагменти. Це необхідно, оскільки мовні моделі зазвичай мають обмеження на обсяг контексту, який вони можуть обробити, тому необхідно створювати якомога менші фрагменти тексту. Пізніше ці фрагменти даних перетворюються у вектори, що допомагає підвищити швидкість пошуку та виведення даних. Після генерації векторів наступним кроком є створення індексу, зберігання оригінальних фрагментів корпусу та вбудовування у формі пар ключ-значення для швидкого та частого пошуку в майбутньому [32].

Нами встановлено, що наступним кроком є пошук даних. Отримавши запит користувача, та сама модель кодування, що й на першому етапі, використовується для перетворення запиту у вектор. Обчислюється подібність між запитом та інформацією збереженою у векторній базі даних. Знайшовши потрібні нам дані, починається наступний етап [32].

Останній етап, описаний дослідниками включає такі дії. Отримавши запит користувача та дані з бази даних, ВММ генерує відповідь на основі

наданої, базою даних, інформації. Окрім того, історична інформація про діалог, може також слугувати додатковими даними для BMM [32].

З цього приводу, С. Бесен виділяє ключову перевагу RAG: можливість доступу до спеціалізованих даних конкретного домену та конфіденційної інформації, яка не включена до стандартних даних навчання мовної моделі. Це особливо корисно для організацій, що працюють над власними дослідженнями та проєктами, коли компанія може надати RAG-системі доступ до внутрішніх даних, дозволяючи моделі використовувати спеціалізовані знання без необхідності перенавчання на конфіденційній інформації. Іншою важливою перевагою є здатність RAG надавати моделям доступ до найактуальнішої інформації, долаючи бар'єр обмежених знань моделі. RAG-системи можуть інтегруватися з живими джерелами даних, такими як: інтернет, новинні стрічки. Постійне оновлення бази даних забезпечує використання поточної інформації при генерації відповідей [14].

Фанчжен (Франк) Сюй, дослідник з Майкрософт, що працює у сфері штучного інтелекту, виділяє проблемні області під час використання генерації доповненої пошуком: якість пошуку, якість генерації відповідей та процес доповнення [14]. Проблема якості пошуку є багатогранною, основною проблемою є низька точність, коли не всі дані з векторної бази даних корелюють із запитом, що призводить до потенційних галюцинацій. Вторинно проблемою є низька повнота, яка виникає, коли не всі відповідні блоки витягнуті, що перешкоджає LLM отримати достатній контекст для генерації відповіді. Крім того, база даних, також повинна регулярно оновлюватися для запобігання використанню надлишкових або застарілих даних, що можуть призвести до неточних результатів пошуку [24].

Що стосується генерації відповідей, галюцинації залишаються найбільшою проблемою, коли модель генерує вихідні дані з вигаданою інформацією. Нерелевантність є ще однією проблемою, коли згенерована відповідь, не відповідає на запит. Крім того, токсичність або упередженість,

коли модель генерує шкідливу або образливу відповідь, є ще одним недоліком такого підходу [24, 33].

Процес доповнення (англ. “augmentation”) у RAG-системах стикається з низкою викликів. Ключовою проблемою є ефективна інтеграція контексту з отриманих фрагментів у завдання генерації – неналежна обробка може призвести до незв'язних або нечітких результатів. Надлишковість інформації виникає, коли багато фрагментів містять схожий контент, що спричиняє повторення на етапі генерації. Додатковими викликами є складність визначення релевантності та важливості кожного отриманого фрагмента, узгодження різних стилів і тонів письма для забезпечення якісного виводу, а також ризик надмірної залежності моделі від розширеної інформації, що може призвести до простого повторення отриманого контенту без створення нової цінності чи синтезованої інформації [32].

Дистиляція знань (англ. “Knowledge Distillation”) – це техніка стиснення моделей, яка дозволяє передати знання від великої, складної мовної моделі (“вчителя”) до меншої, ефективнішої моделі (“учня”) без втрати продуктивності. Цей метод був вперше описаний Бучілу та співавторами у 2006 році [1, 19].

Основною перевагою дистиляції знань є можливість створення компактних моделей, які зберігають більшість можливостей великих моделей при значно меншій обчислювальній складності. Це робить дистильовані моделі придатними для застосування на мобільних пристроях та в середовищах з обмеженими ресурсами [39].

Результати нашого дослідження довели, що дистиляція знань стикається з рядом обмежень та викликів. Найсуттєвішою проблемою є невідповідність потужностей між вчителем та учнем – дослідження показують, що більші та точніші моделі-вчителі не обов'язково призводять до кращих учнів. Коли різниця в потужності занадто велика, модель-учень фізично не здатна імітувати складну поведінку вчителя, що призводить до високої розбіжності між їхніми прогнозами навіть під час навчання [19]. Послідовна дистиляція через проміжні моделі також виявляється неефективною для подолання цієї проблеми.

ШІ-агенти (AI Agents) запропоновані Дж. Перлом та А. Ньюеллом зробили еволюційний крок у розвитку штучного інтелекту, переходячи від пасивних генерацій відповідей до активних, цілеспрямованих учасників вирішення складних завдань. Якщо традиційні великі мовні моделі обмежуються генерацією тексту на основі отриманого промпту, то ШІ-агенти здатні самостійно розкладати складні цілі на підзадачі, планувати послідовність дій, взаємодіяти з зовнішніми інструментами та адаптувати свою стратегію на основі отриманих результатів. Цей підхід дозволяє створювати системи, які не просто відповідають на запити, а працюють над досягненням поставлених цілей, що відкриває нові можливості для автоматизації інтелектуальних завдань у різних галузях [61].

Нами встановлено, що на відміну від традиційних трансформерів, які генерують відповіді на основі виключно свого внутрішнього знання, ШІ-агенти активно взаємодіють із зовнішніми інструментами та ресурсами: базами даних, веб-пошуком, API та навіть іншими агентами для отримання необхідної інформації. Після збору інформації агент оновлює свою базу знань та використовує агентне міркування (англ. “agentic reasoning”) – процес безперервної переоцінки плану дій та самокорекції, що забезпечує більш адаптивне прийняття рішень [11].

Архітектура сучасних ШІ-агентів ґрунтується на п'яти ключових компонентах, які забезпечують їхню функціональність та ефективність у вирішенні складних завдань. Перший компонент - модель (BMM), яка є центральним елементом прийняття рішень та обробки інформації. Ця модель відповідає за розуміння запитів, генерацію відповідей та визначення стратегій дій. Другий компонент - інструменти, вони забезпечують агентам можливість виконувати дії та взаємодіяти із зовнішнім середовищем, включаючи доступ до баз даних, API, пошукових систем та інших зовнішніх ресурсів [11].

Третій компонент - знання та пам'ять, включає два рівні: робочу пам'ять для короткострокового контексту активної сесії та постійну пам'ять для довгострокового збереження попередніх взаємодій та історії завдань за

допомогою векторних баз даних. Четвертий компонент - обмежувальні механізми, які забезпечують безпеку, етичність та контрольованість поведінки агента, запобігаючи небажаним або шкідливим діям. П'ятий компонент - рівень оркестрації - керує процесом обробки інформації, координує взаємодію між іншими компонентами, виконує міркування та приймає рішення про наступні дії агента.

В ряді досліджень відмічено, що ключовою особливістю ШІ-агентів є їхня здатність ділитися інформацією між різними інструментами, що робить їх більш універсальними порівняно з традиційними ВММ. Наприклад, при плануванні відпустки агент може спочатку звернутися до зовнішньої бази даних з історичними метеорологічними даними, потім проконсультуватися з іншим спеціалізованим агентом для отримання експертних знань, після чого синтезувати отриману інформацію для формування обґрунтованої рекомендації. Багатоінструментальна архітектура дозволяє агентам долати обмеження базових мовних моделей, які не можуть спеціалізуватися на всіх можливих доменах знань, та забезпечує гнучкість у вирішенні різноманітних завдань [11].

Основними перевагами ШІ-агентів є автономність та масштабованість. Агенти можуть працювати цілодобово без втомлюваності, обробляти великі обсяги інформації та адаптуватися до нових ситуацій без необхідності перепрограмування. Це робить їх особливо цінними для застосування в автоматизації обслуговування клієнтів, аналітиці даних, науковому дослідженні та оптимізації бізнес-процесів, де потрібна здатність до самостійного прийняття рішень на основі динамічно змінюваного контексту [61].

Однак Е. Алонсо підкреслює, що AI-агенти також стикаються з рядом викликів та обмежень. Основною проблемою є надійність та передбачуваність поведінки: автономні агенти можуть приймати неочікувані рішення або виконувати помилкові дії, особливо в нових або неоднозначних ситуаціях. Питання безпеки та контролю є критичними, оскільки агенти з доступом до зовнішніх систем та даних потенційно можуть завдати шкоди при неправильному функціонуванні

Висновки до 1-го розділу

Проаналізовано еволюційний розвиток великих мовних моделей від символічної ери обробки природної мови на основі системи правил до сучасних нейронних архітектур. Встановлено, що революційною віхою стало запровадження архітектури трансформера у 2017 році, яка завдяки механізму самоуваги дозволила моделям ефективно обробляти послідовності різної довжини. Науковці зазначають, що саме ця архітектура стала фундаментом для всіх сучасних великих мовних моделей.

Розглянуто три основні типи архітектур на базі трансформера: GPT (decoder-only), BERT (encoder-only) та T5 (encoder-decoder). Аналіз джерельної бази дослідження дозволяє констатувати, що кожна архітектура оптимізована для специфічних завдань обробки природної мови. GPT виявився ефективним для генерації тексту та підтримки контексту у довгих діалогах, BERT демонструє переваги у завданнях розуміння та аналізу тексту, тоді як T5 є універсальним варіантом для перетворення тексту в текст.

З'ясовано, що сучасні методологічні підходи до навчання великих мовних моделей базуються на багатоетапному процесі. Попереднє навчання формує фундаментальні лінгвістичні знання на основі масштабних корпусів текстів, тонке налаштування забезпечує адаптацію до спеціалізованих завдань, навчання з підкріпленням через людський зворотний зв'язок дозволяє узгодити поведінку моделі з людськими цінностями, а параметро-ефективне тонке налаштування суттєво зменшує вимоги до обчислювальних ресурсів.

Встановлено, що для практичного застосування великих мовних моделей у спеціалізованих доменах використовуються різноманітні методи оптимізації. Промпт-інженерія дозволяє адаптувати поведінку моделі через структуровані запити без необхідності перенавчання. Генерація доповнена пошуком (RAG) вирішує проблему актуальності знань моделі через інтеграцію з зовнішніми базами даних. Дистиляція знань забезпечує створення компактних моделей для застосування на пристроях з обмеженими ресурсами.

Проаналізовано еволюцію від традиційних мовних моделей до ШІ-агентів, які здатні самостійно розкладати складні цілі на підзадачі, планувати послідовність дій та взаємодіяти з зовнішніми інструментами. Архітектура сучасних ШІ-агентів включає п'ять ключових компонентів: мовну модель, інструменти, систему пам'яті, обмежувальні механізми та рівень оркестрації. Ця багатокомпонентна структура забезпечує автономність, адаптивність та здатність до прийняття рішень на основі динамічно змінюваного контексту.

РОЗДІЛ 2. ДІАЛОГОВІ СИСТЕМИ З КОНТЕКСТНОЮ ПІДТРИМКОЮ: ТЕОРЕТИЧНІ ЗАСАДИ ТА МЕТОДОЛОГІЧНИЙ БАЗИС

2.1. Контекст у діалогових системах: концептуалізація та методологія управління

Розвиток чат-ботів підкреслив важливість збереження контексту в діалогових системах. Розробка та розвиток підходів для моделювання контексту відображає не лише технологічний прогрес, а й поглиблення теоретичних знань про розуміння природи діалогів.

Під терміном *контекст* розумітимемо визначення, запропоноване Б. Грош у 1980-х роках, всі дані, які система використовує для оцінювання поточної взаємодії. Науковці погоджуються контекст відіграє центральну роль у проектуванні діалогових систем з обробки природньої мови. Однак контекст важко визначити однозначно. У дослідженнях контекст розглядається як сукупність аспектів або контекстуальних елементів, кожен з яких може бути визначений і проаналізований та підкріплює здатність системи розуміти та генерувати релевантні відповіді під час комунікації [2].

Прикладом ранніх діалогових систем може слугувати ELIZA, розроблена Дж. Вайценбаумом у 1966 році. Система оперувала примітивним розумінням контексту, обмежуючись простим віддзеркаленням ключових слів користувача без справжнього розуміння семантичних зв'язків [66]. Такі системи ґрунтувалися на жорстких лінгвістичних правилах і не могли утримувати інформацію про попередні репліки діалогу. Концепція контексту в цей період зводилася переважно до синтаксичного аналізу поточного висловлювання. ELIZA була одним перших чат-ботів, які пройшли тест А. Тюрінга, згаданий у попередньому розділі [58].

Згідно з дослідженнями подальший розвиток лінгвістики призвів до появи статистичних моделей, зокрема прихованих марковських моделей (НММ), які дозволили передбачати лінгвістичні явища на основі попереднього контексту [56]. Це стало ключовим моментом у розвитку розуміння контексту як

послідовності вхідних даних, де кожен наступний елемент діалогу залежить від попередніх.

Науковці погоджуються, що справжньою революційною зміною у сфері контексту було впровадження трансформерних архітектур, таких як BERT і GPT [21, 28]. ВММ перевернули уявлення про поняття контексту через механізм уваги, який динамічно визначає релевантність елементів історії діалогу. Сучасні моделі, наприклад: GPT-4, демонструють здатність утримувати довгостроковий контекст, що передбачає не лише збереження попередніх реплік, але й розуміння їхніх семантичних взаємозв'язків та намірів користувача.

Отже, під поняттям діалогова система, будемо розуміти це комп'ютерну систему, призначену для спілкування з людиною зв'язною, структурованою мовою [5]. Дослідники поділяють діалогові системи на цільові (англ. “goal-oriented”) та нецільові (англ. non “goal-oriented”) [20].

Цільові діалогові системи побудовані для вирішення конкретної задачі і зазвичай мають доступ до бази даних. Я. Дерю виділяє такі компоненти цільових діалогових систем [20]:

- система розуміння природньої мови (англ. “natural language understanding”) перетворює вхідні дані у формат зрозумілий машині;
- трекер стану діалогу (англ. “dialogue state tracker”) аналізує та визначає теперішній стан діалогу для вибору наступної дії;
- система визначення наступної дії (англ. dialogue policy learning) обирає наступну дію на основі теперішнього стану діалогу;
- генератор природньої мови (англ. “natural language generator”) перетворює згенеровані дані у формат природньої мови.

Нами встановлено, що кожна цільова діалогова система створюється і налаштовується від конкретну задачу. При цьому часто деякі або всі компоненти створюються інженерним методом, з вручну створеними правилами у вигляді скриптів коду. Такі системи неможливо швидко переналаштувати під інші задачі [2].

У теперішній час створення універсальної діалогової системи є відкритою проблемою. Нецільові діалогові системи, які ще називають чат-ботами, не намагаються вирішити конкретну задачу. Їх ціль в тому, щоб підтримати розмову з користувачем на визначену або вільну тему [5].

Контекст є фундаментальною компонентом у проектуванні діалогових систем, що визначає їхню здатність підтримувати релевантну комунікацію. Здобуті нами дані свідчать, що ефективно управління контекстною інформацією вимагає комплексного методологічного підходу, який охоплює як технічні аспекти зберігання даних, так і когнітивні механізми їх інтерпретації [5].

У процесі дослідження проблеми виявлено, що сучасні діалогові системи оперують контекстом на декількох рівнях одночасно: лінгвістичному, семантичному та прагматичному.

Розглянемо детальніше методи представлення контексту в архітектурі діалогових систем. Аналіз літератури підтверджує [5, 65], що найпоширенішим підходом є векторне представлення контекстної інформації через ембеддинги. Цей метод дозволяє перетворювати текстові елементи діалогу в багатовимірні вектори, де семантично близькі концепції розташовуються поруч у векторному просторі. Наприклад, у діалоговому контексті фрази "я хочу замовити" та "мені потрібно придбати" матимуть схожі векторні представлення [65].

Важливим аспектом методології управління контекстом є архітектура пам'яті в сучасних ШІ-агентах. Згідно з дослідженнями [20], агенти потребують пам'яті для збереження та відновлення минулого досвіду з метою покращення прийняття рішень. На відміну від традиційних моделей, які обробляють кожне завдання незалежно, агенти з пам'яттю можуть зберігати контекст, розпізнавати патерни з часом і адаптуватися на основі попередніх взаємодій. Здобуті нами дані свідчать, що дослідники категоризують агентну пам'ять подібно до того, як психологи категоризують людську пам'ять: короткострокова пам'ять дозволяє запам'ятовувати недавні вхідні дані, тоді як довгострокова пам'ять забезпечує збереження інформації між різними сесіями [65].

Однією з ключових методологічних проблем є управління обмеженнями контекстного вікна. Як зазначає Я. Дерю [20], проєктування пам'яті агента це, по суті, інженерія контексту: визначення того, які токени потрапляють у контекстне вікно та як вони організовані.

Серед методів селекції та фільтрації контексту науковці виділяють декілька основних підходів. Системи пам'яті поєднують такі техніки, як підсумок, переписування контексту та пошук, для управління різними компонентами пам'яті [2]. Техніка підсумків дозволяє стискати довгі фрагменти діалогу, зберігаючи лише ключову інформацію. Метод *sliding* ковзного вікна (англ. “*sliding window*”) зберігає лише останні *N* реплік діалогу, автоматично видаляючи найдавніші повідомлення. Підхід генерування доповненого пошуком (RAG) використовує семантичний пошук для динамічного відновлення релевантного контексту з довгострокової пам'яті.

В процесі дослідження ми переконались, що механізм самоуваги, запроваджений в архітектурі трансформера [65], відіграє критичну роль у визначенні релевантності контекстних елементів. У діалоговому контексті це дозволяє системі автоматично визначати, які попередні репліки є критично важливими для розуміння поточного запиту користувача, а які можна проігнорувати.

Однією з поширених проблем управління контекстом, яку описують дослідники, є розв'язання суперечностей у контекстній інформації. Коли користувач у різних репліках надає суперечливу інформацію, система повинна визначити, яка інформація є актуальною. Сучасні методології передбачають використання часових міток та механізмів відстеження зміни тем для вирішення таких конфліктів. Система надає пріоритет більш свіжій інформації, але також може запитати користувача про уточнення в випадку суперечностей [65].

Серед недоліків існуючих методологій управління контекстом науковці виділяють такі пункти: втрата релевантної інформації при скороченні контексту, накопичення помилок при довгих діалогах, високі обчислювальні вимоги для обробки великих контекстних вікон [2, 5]. Ці обмеження стимулюють подальші

дослідження в напрямку розробки більш ефективних алгоритмів селекції контексту та оптимізації використання обчислювальних ресурсів.

Отже, можна зробити висновок, що методологічні підходи до збереження та управління контекстною інформацією в діалогових системах базуються на поєднанні векторних представлень, багаторівневої архітектури пам'яті, механізмів селекції та фільтрації, а також спеціалізованих агентів для динамічного оновлення контексту. Розуміння цих фундаментальних методологій є особливо важливим для проєктування ефективних діалогових систем, які можуть підтримувати природну та зв'язну комунікацію з користувачами протягом тривалих сесій взаємодії.

2.2. Методика оцінювання якості функціонування діалогових систем з контекстною підтримкою

Оцінювання якості діалогових систем є важливим етапом розробки, що дозволяє визначити ефективність системи та ідентифікувати напрямки для її вдосконалення. Методи оцінювання тісно пов'язані з характеристиками діалогової системи, яку вони призначені аналізувати, тому якість визначається в контексті завдань, яку система має виконувати. Здобуті нами дані свідчать, що система, призначена для відповідей на запитання, оцінюватиметься на основі правильності та релевантності відповідей [46].

Розглянемо детальніше основні підходи до оцінювання діалогових систем. Дослідження підтверджують, що людська оцінка залишається стандартом для оцінювання діалогових агентів, проте це дуже дорогий та часозатратний процес [53]. Під час дослідження проблеми виявлено, що дослідники переважно покладаються на автоматичні метрики під час розробки діалогових систем. Ціль методу оцінювання полягає у отриманні результатів виконання тестів: автоматизованих та повторюваних процедур, які дозволяють ефективно порівнювати якість різних діалогових систем та забезпечують об'єктивне вимірювання продуктивності системи [46].

Важливим аспектом методології оцінювання є відмінність між оцінюванням на рівні окремих реплік та оцінюванням цілих діалогів. Оцінювання діалогових систем передбачає врахування всієї історії розмови як контексту [16].

Аналіз літератури підтверджує необхідність визначення сценаріїв та очікуваних результатів розмови, а не лише входів та виходів для однокрокових випадків використання. Атрибути якості можна групувати відповідно до різних аспектів зручності використання. Дослідники виявили, що атрибути якості узгоджуються з ефективністю, результативністю та задоволеністю користувачів. Ефективність відноситься до точності та повноти досягнення цілей, результативність описує використання ресурсів, а задоволеність визначає загальний користувацький досвід [46].

Н. Радзвіль підкреслює, що категорія продуктивності (англ. “performance”) включає такі ключові атрибути як стійкість до непередбачуваних вхідних даних та коректне розпізнавання і обробку команд [46].

В ряді досліджень відмічено, що категорія функціональності (англ. “functionality”) охоплює точність розуміння команд, використання відповідного рівня стилю мовлення, лінгвістичну правильність вихідних даних, виконання запитуваних завдань. Дослідження показали, що система повинна містити широту знань та бути гнучкою в їх інтерпретації, що забезпечує адаптивність до різноманітних користувацьких запитів [46, 53].

Категорія людяності (англ. “humanity”) визначає здатність системи до переконливої та природної взаємодії. Науковці підкреслюють, що система має відповідати на конкретні запитання та підтримувати дискусію на обрану тему протягом всієї розмови. Варто зазначити питання щодо проходження тесту Тюрінга: деякі дослідники стверджують, що створення враження людини не є валідним атрибутом якості, оскільки користувачі готові взаємодіяти з вигаданими персонажами [46, 56].

Категорія емоційності (англ. “affect”) включає надання привітань, передачу особистості, підтримка емоційності через тон розмови. Здобуті нами

дані свідчать про важливість здатності читати та відповідати на настрої учасників розмови, що значно підвищує залученість користувачів.[46]

Категорія етики та поведінки (англ. “ethics & behavior”) охоплює повагу до користувачів, етичні та культурні особливості, чесність у розкритті природи чатбота, чутливість соціальних проблем. Дослідження підтверджують критичну важливість довіри, яка пов'язана із сприйняттям якості системи [46].

Розглянемо детальніше метрики для оцінювання діалогових систем з підтримкою контексту та множинних агентів. Метрика чіткості (англ. “precision”) вимірює відповідність відповіді запиту та дотримання теми діалогу. Ця метрика оцінює, наскільки точно система розуміє намір користувача та наскільки релевантною є згенерована відповідь у контексті розмови. Для діалогових систем з контекстною підтримкою метрика чіткості визначається як відношення релевантних та тематично відповідних фрагментів інформації до загальної кількості включеної інформації у відповідь. Високий показник чіткості свідчить про здатність системи фокусуватися на суті запиту [46, 56].

Метрика інференції (англ. “inference”) оцінює здатність системи правильно вибирати відповідного агента для обробки конкретного запиту. В багатоагентних діалогових системах цей показник є критичним, оскільки визначає, чи система коректно визначає тип запиту та направляє його до спеціалізованого агента [16].

Метрика збереження контексту (англ. “content retention”) вимірює здатність системи запам'ятовувати та використовувати інформацію з попередніх повідомлень у розмові. Ця метрика є важливою для багатокрокових діалогів, де кожна наступна відповідь залежить від попереднього контексту [46].

Метрика відповідності правилам (англ. “rule compliance”) оцінює дотримання системою встановлених поведінкових норм та обмежень. Ця складна метрика включає кілька компонентів: Уникнення шкідливого контенту: система не повинна генерувати відповіді, що містять образливий, дискримінаційний або небезпечний контент Коректність зміни агентів: система повинна плавно переходити між агентами лише за необхідності, без створення

розривів у розмові. Належне завершення використання агента: система має коректно закрити сесію з агентом після виконання завдання та передавати керування відповідним чином [16, 27]

Метрика складності (англ. “complexity”) оцінює здатність системи адаптувати рівень відповідей до складності теми. Для легких тем, таких як базові інформаційні запити або прості транзакції, очікується висока точність та швидкість відповідей. Для важких тем, що вимагають глибокого розуміння контексту, інтеграції множинних джерел інформації або складних міркувань, метрика враховує не лише точність, але й повноту відповіді. Система оцінюється окремо на наборах простих та складних запитів, а загальний показник включає різницю у продуктивності між цими категоріями [16].

М. Бентон зазначає, що багато автоматичних методів оцінювання для генерації відповідей мають багато відмінностей з людськими судженнями. Традиційні метрики перекриття слів, розроблені для машинного перекладу, виявилися неефективними для оцінювання діалогових систем через множинність правильних відповідей на один запит [46].

Розглянемо методологічні етапи комплексного оцінювання. Спочатку створюється набір тестових вхідних даних та очікуваних результатів. Потім проводяться сесії взаємодії з системою за участю користувачів, під час яких збираються дані за обраними метриками [53].

Для забезпечення об'єктивності результатів, науковці пропонують обчислювати міжкористувацьку узгодженість. Зазвичай кілька користувачів тестують систему і наприкінці оцінюють її за визначеними критеріями. Для уникнення екстремальних оцінок через упередженість використовується коефіцієнт Каппа, і лише оцінки користувачів з показником понад 0.2 включаються у фінальне вимірювання [53].

Розглянемо також метрику ефективності маршрутизації, яка оцінює здатність системи мінімізувати кількість передач між агентами для генерування відповіді. Оптимальна система повинна направляти запит безпосередньо до

найбільш відповідного агента з першої спроби, уникаючи непотрібних переадресацій [16].

Здобуті нами дані свідчать про можливість практичного застосування описаної методів для порівняння різних версій діалогової системи або оцінювання покращень з часом. Дослідники можуть використовувати тестовий чеклист для забезпечення охоплення ключових аспектів функціонування системи.

Висновки до 2-го розділу

Проаналізовано поняття контексту в архітектурі діалогових систем. Встановлено, що контекст визначається як всі дані, які система використовує для оцінювання поточної взаємодії, та еволюціонував від примітивного синтаксичного аналізу в ELIZA до динамічного механізму уваги в сучасних великих мовних моделях.

З'ясовано, що методологічні підходи до управління контекстною інформацією базуються на векторних представленнях через ембеддинги, багаторівневій архітектурі пам'яті (короткострокова та довгострокова) та методах збереження контексту, таких як підсумки, ковзне вікно та генерування доповнене пошуком.

Проаналізовано методику оцінювання якості функціонування діалогових систем з контекстною підтримкою. Атрибути якості групуються за категоріями продуктивності, функціональності, людяності, емоційності та етики. Основними метриками оцінювання є чіткість відповіді, інференція для вибору агента, збереження контексту, відповідність правилам та складність обробки тем.

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА ІМПЛЕМЕНТАЦІЯ БАГАТОМОВНОЇ ДІАЛОГОВОЇ СИСТЕМИ

3.1. Проектування системної архітектури та обґрунтування вибору технологічного стеку

Проектування архітектури чат-бота є критичним етапом розробки, оскільки воно визначає функціональні можливості системи, її масштабованість та ефективність роботи. Одним із завдань нашого дослідження є проектування Telegram-бота, який міг би підтримувати кілька ШІ-агентів і зберігати контекст розмови, використовуючи обмежені обчислювальні ресурси. Зазначені вище умови змусили нас переглянути існуючі технологічні рішення та вибрати оптимальний підхід до вибору реалізації для нашої системи.

Традиційний підхід створення чат-ботів передбачає написання коду з використанням мов програмування (Python або JavaScript). Таке завдання вимагає глибоких технічних знань та значних часових ресурсів. Зазначимо, що розвиток low-code та no-code платформ відкрив нові можливості для створення комплексних систем автоматизації без необхідності написання коду. Серед таких рішень ми розглянули та обрали платформу n8n, яка поєднує у собі гнучкість традиційного програмування з візуальною простотою конструкторів [61].

Хочемо підкреслити, що концепція архітектури нашого чат-бота базується на модульному підході. Кожен компонент системи виконує функцію та взаємодіє з іншими елементами. Для адаптації ВММ до окремих завдань, було обрано метод створення ШІ-агентів. Результати досліджень доводять, що такий підхід допомагає поєднати декілька компонентів, що виконують окремі завдання, в одну систему. Окрім зазначеного вище, ШІ-агенти дозволяють легко розширювати архітектуру та експериментувати зі способами налаштування системи [11]

Підкреслимо, що особливу увагу при проектуванні було приділено питанню збереження контексту розмови, оскільки це є ключовою вимогою для забезпечення природної взаємодії користувача з ботом. Контекст розмови включає не лише історію повідомлень та стан діалогу з користувачем. Для

реалізації цієї функціональності було обрано підхід з використанням модуля Simple Memory, який забезпечує зберігання та швидкий доступ до контекстної інформації.

Платформа програмування n8n була обрана після ретельного розгляду технічних та економічних аспектів. По-перше, платформа пропонує безкоштовний пробний період, щоб система могла бути протестована та розроблена на всіх етапах без дорогих початкових інвестицій. Це є важливим фактором для академічної літератури та експериментальних програм з обмеженим бюджетом. На відміну від звичайного програмування, n8n підтримує візуальний процес для взаємодії з модулями/агентами, що дозволяє легко створювати робочий процес і прискорює процеси розробки [48]. При тестуванні різних налаштувань архітектури це є вирішальним, оскільки альтернативні конфігурації можуть бути швидко створені без необхідності переписувати код.

Ключова перевага n8n полягає в тому, що вона розміщена в хмарному середовищі, що полегшує обчислення з вищезазначеного підходу. Вона передає складні обробні завдання на інші комп'ютери, що дозволяє створювати потужні ШІ-системи на відносно низькорівневому обладнанні. Ця модель роботи є особливо корисною для проектів, які включають ресурсомісні великі мовні моделі, оскільки вона не залежить від локального обладнання.

В процесі дослідження ми переконалися, що n8n підтримує більше 400 доступних інтеграцій з безліччю сервісів та API, що полегшує інтеграцію або з'єднання частин пазлу без написання коду для кожної частини [48]. Для нашого проекту це включало нативну підтримку Telegram Bot API для спілкування з користувачами, Google Gemini Chat Model для обробки природної мови, Simple Memory для зберігання контексту діалогу та Agent Tools для створення окремих агентів, на Рисунку 3.1.1 зображена архітектура компонентів чатбота.

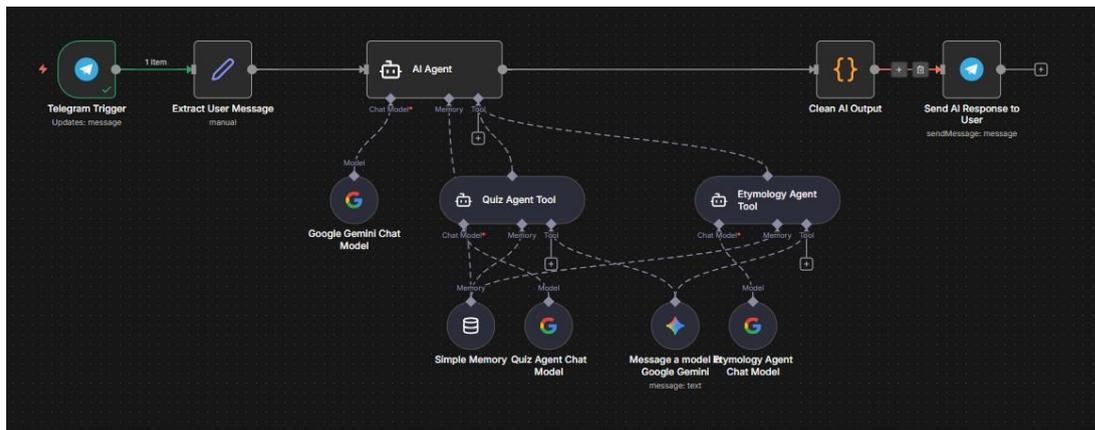


Рис.3.1.1. Архітектура чатбота

Зазначимо, що у пнп робочі процеси відображаються візуально, що полегшує налагодження або модифікацію архітектури, оскільки розробник має можливість мати доступ до даних у реальному часі. Крім того, архітектура платформи також зображена у візуальному форматі, тобто система може бути розширена за потреби за допомогою агентів або інструментів, які не потрібно переписувати для роботи, що дозволяє повільно і поступово покращувати чат-бот під час розробки.

Вибір великої мовної моделі є одним з основних рішень при проектуванні систем AI-агентів, оскільки мовна модель сприяє якості розуміння запиту користувача і впливає на формування відповіді. Ми обрали модель Google Gemini 2.5 Flash і проаналізували технічні якості, щоб порівняти її з іншими моделями сімейства Gemini для нашого проекту. Ми адаптували Gemini для використання в системах з ефективною обробкою запитів, зберігаючи високу якість відповідей.

Дані досліджень Google, свідчать, що Gemini 2.5 Flash підтримує розмір контекстного вікна до одного мільйона токенів, що дозволяє їй відповідати на надзвичайно довгі діалоги і зберігати релевантну інформацію протягом тривалих сесій. Також швидкість обробки оптимізована для низької затримки відповіді, що дуже важливо для забезпечення природної розмови під час процесу [27]. Крім того, як найдешевша модель за токен серед моделей

сімейства Gemini, вона представляє економічно ефективне рішення для проектів з обмеженим бюджетом.

Порівняння з Gemini 1.5 Flash, попереднім поколінням моделі, підкреслило великі переваги нової версії. Gemini 1.5 Flash має обмежене контекстне вікно в 128 000 токенів, що значно ускладнює здатність системи запам'ятовувати довгу історію чату. Перша версія також демонструє слабшу продуктивність складних запитів, особливо в складних або спеціалізованих областях, таких як лінгвістика та етимологія. Це особливо актуально для багатокрокових завдань, які є важливими елементами генерації вікторин та етимологічного аналізу, оскільки вони повинні самостійно розбивати складне завдання на ітеративну послідовність операцій.

В процесі дослідження ми переконалися, що Gemini 1.5 Pro була більш потужною моделлю, ніж попереднє покоління, хоча вона забезпечувала кращу продуктивність для надскладних завдань, але була занадто ресурсоємною для нашого проекту. Модель є дорожчою і, таким чином, економічно недоцільною для освітнього проекту. Навіть швидкість, з якою Gemini 1.5 Pro відповідає, менша за швидкість версії Flash, що не є зручним для взаємодії в реальному часі [25]. Спираючись на результати досліджень науковців з Google, 1.5 Pro дійсно занадто потужна, оскільки 2.5 Flash здатна виконувати ці завдання з тією ж якістю і значно нижчою вартістю з набагато більшою швидкістю обробки.

Gemini 2.0 Flash, яка є експериментальним підходом, доступним у меншому масштабі, не була обрана через різні обмеження. Це експериментальна модель, тому не гарантується стабільна робота і також можуть містити невиявлені помилки [26]. Відсутність детальної документації ускладнює налагодження та оптимізацію системи.

Більше того, без попередження експериментальні версії можуть бути змінені або видалені, що створює ризик для підтримки проекту в довгостроковій перспективі. Модель демонструє хороші знання лінгвістичних концепцій для основного агента, що підтримує освітню допомогу, і базується на великому корпусі академічних текстів для навчання.

Підкреслимо, що функція розширення діалогів дозволить надавати персоналізовані пропозиції з урахуванням повної історії розмови на основі повного введення, що призводить до кращої освітньої підтримки. Швидкість відповіді гарантує природне та комфортне спілкування, і з меншими затримками від запиту до інформації користувач не відчуває затримки. Модель може генерувати синонімічні запитання з правдоподібними відволікаючими факторами для агента вікторини, що є необхідною умовою для розробки якісних тестів.

Обрана модель підтримує семантичний аналіз та дозволяє створювати ряд запитань, що охоплюють різні рівні складності та відповідають конкретній темі та контексту розмови. Інтеграція Gemini Search дозволяє використовувати актуальну лексику з різних доменів, що робить систему адаптованою для створення вікторин на будь-яку тему, визначену користувачем. Для агента етимології модель використовує велику базу знань про історію мов та етимологічні зв'язки, що містять деталі про походження слів.

Структурування допомагає з презентацією інформації – етимологічні дані можуть бути представлені у чітко відформатованій формі, з легко визначеними розділами, а також часовими шкалами та порядком. Багатогранна валідація з функцій пошуку моделі забезпечує повноту історичних даних і вирішує невизначеність, коли розглядається розбіжність в етимології.

Після завершення етапу проектування архітектури та вибору технологічного стеку, включаючи платформу n8n і мовну модель Google Gemini 2.5 Flash, постає необхідність перейти від теоретичних концепцій до практичної реалізації системи. Визначивши ключові компоненти архітектури та обґрунтувавши технічні рішення, ми маємо змогу розпочати безпосереднє впровадження чат-бота. Практична реалізація передбачає послідовне виконання кількох етапів: від первинного налаштування Bot API на платформі Telegram до конфігурації робочих процесів у n8n та інтеграції обраних ШІ-агентів.

Розглянемо процес налаштування чат-бота на платформі Telegram. Платформа пропонує створення чат-бота за допомогою спеціалізованого бота

@BotFather, який надає можливість створювати та адмініструвати власні бот-системи. Для цього слід знайти даний акаунт через функцію пошуку та розпочати діалог.

Бот пропонує низку команд для конфігурування бота, зокрема встановлення імені, опису функціоналу, інформації «Про бота», зображення профілю та переліку доступних команд. Розпочинаємо роботу з використання команди `/newbot`, яка допомагає запускати процес творення нового бота. На цьому етапі користувач обирає назву бота, а потім унікальний ідентифікатор (username), який обов'язково має завершуватися словом "bot" (Рис.3.1.2).

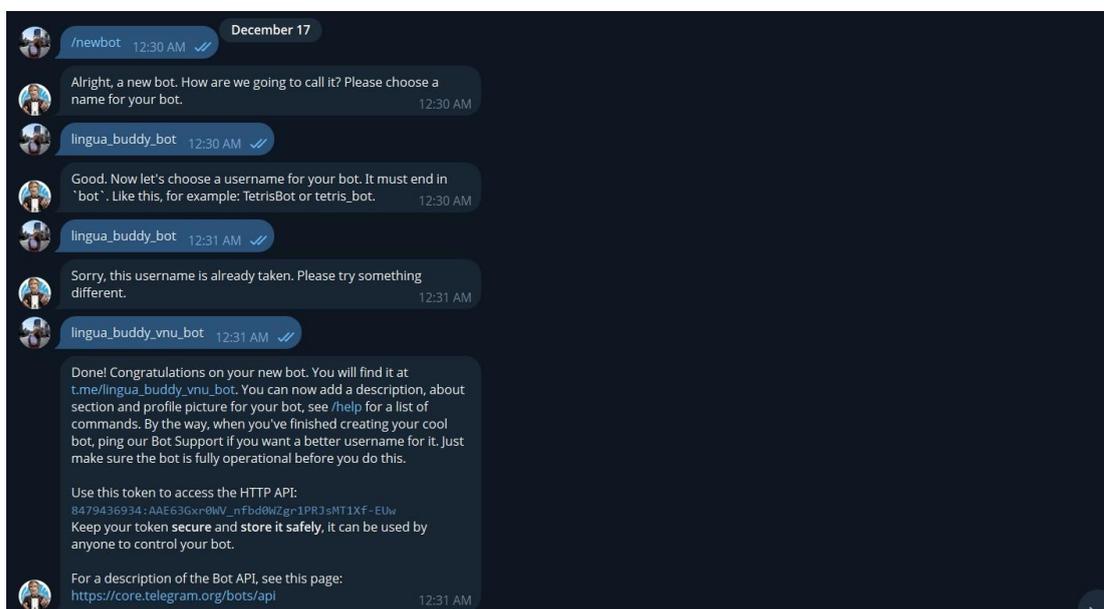


Рис.3.1.2. Створення чат-бота з допомогою команди `/newbot`

Варто підкреслити, що ім'я чат-бота можна змінити, використавши команду `/setname`. Процедура виконується наступним чином: після введення команди необхідно обрати відповідного бота та надіслати нову назву. У нашому випадку було обрано назву `LinguaBuddy`, яка підкреслює мету бота і є легкою для запам'ятовування (Рис.3.1.3).



Рис.3.1.3. Зміна назви чатбота

Після цього необхідно налаштувати стислу інформацію про бота. Стисла інформація відображається у профілі бота та у попередньому перегляді при поширенні посилання. Для встановлення короткого опису використовується команда `/setabouttext` (Рис.3.1.4).

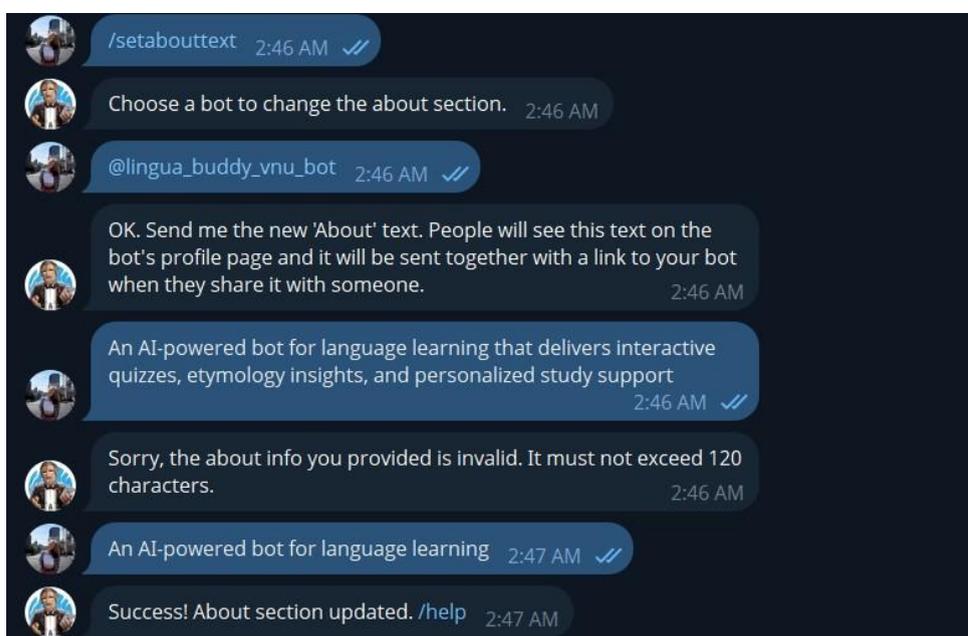


Рис.3.1.4. Налаштування опису чатбота

Наступним кроком є налаштування детального опису призначення та функціональних можливостей бота. Для цього застосовується команда `/setdescription` (Рис.3.1.5). Ця інформація відображається при першій взаємодії

користувача з чат-ботом, перед ініціюванням діалогу. Вона розташовується у блоці з назвою «Що вміє цей бот? / What can this bot do?»



Рис.3.1.5. Налаштування блоку “What can this bot do?”

Останнім кроком є додавання меню команд, для їх конфігурування використовується команда */setcommand* (Рис.3.1.6). Команди потрібно надіслати у наступному форматі: назва команди вводиться латинськими літерами, через дефіс додається опис призначення команди; наступна команда вводиться з нового рядка. Для команди */start* було додано опис “Welcome message and introduction”, а для команди */quiz* – “Enter quiz mode for vocabulary practice”.

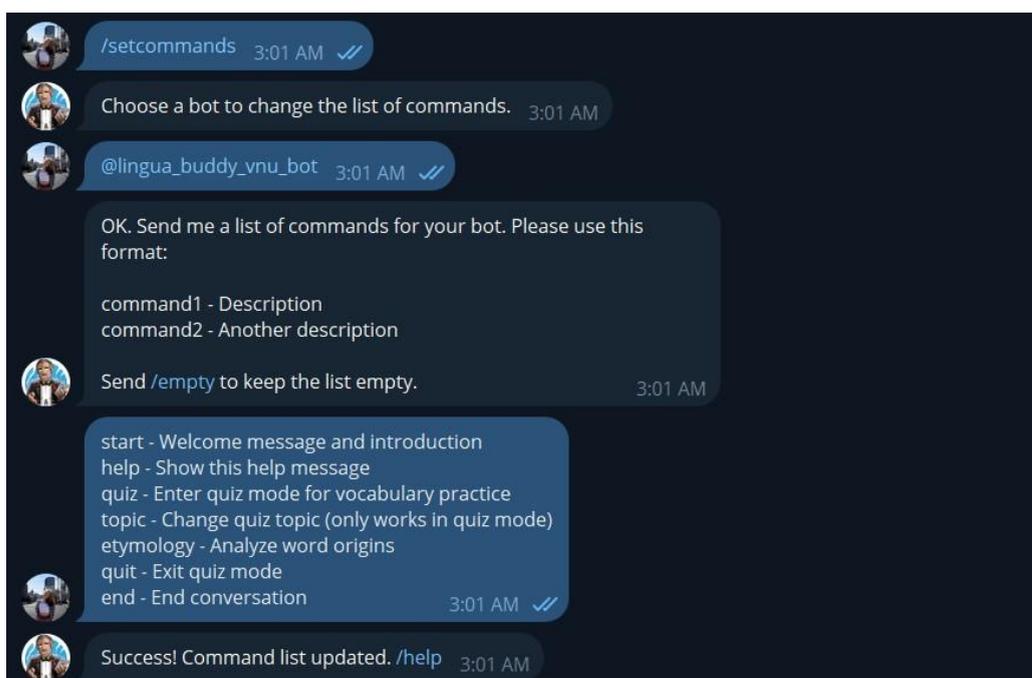


Рис.3.1.6. Налаштування меню команд

Наступним етапом є побудова архітектури агентів, для якого потрібно зберегти токен чат-бота. Скориставшись командою `/mybots` та обравши потрібний бот у BotFather, ми отримали токен, який далі використовуватимемо для приєднання агентів до бота та отримання відповідей.

Для початку побудови системи потрібно створити тригер, який активуватиме роботу агентів. У нашому випадку тригером є повідомлення у Telegram чаті від користувача. На цьому етапі ми налаштовуємо елемент, розв'язуючи його до нашого чат-бота за допомогою токена. Перевіривши, що система реагує на повідомлення користувача ми додаємо додаток, який зберігає текст повідомлення користувача *Extract User message* (Рис.3.1.7). Такий крок є необхідним для ефективної обробки запиту та збереження повідомлення у пам'яті чат-бота для відтворення збереженого контексту.

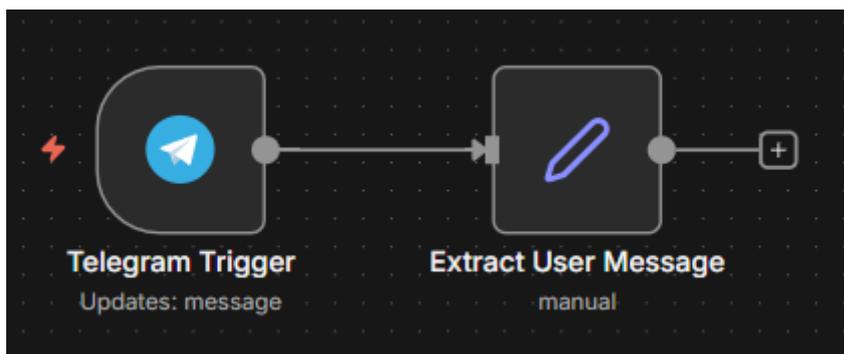


Рис.3.1.7. Створення тригера чат-бота

Наступним кроком є створення основного агента, який може відповідати на запитання та буде основною ланкою комунікації між користувачем та іншими агентами. Після додавання компоненту *AI Agent* потрібно налаштувати основні елементи агента, а саме: промпт, ВММ, пам'ять та інструменти (Рис.3.1.8).

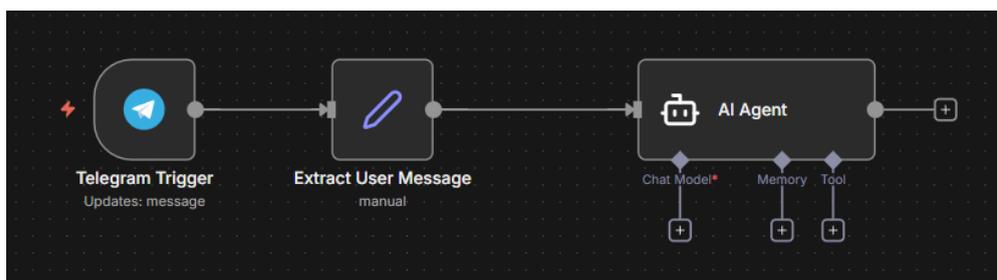


Рис.3.1.8. Додавання основного агента системи

Першим елементом агента є промпт, який визначає особливості роботи агента, правила та формат відповіді. Розробники платформи n8n рекомендують створювати промпт агента вказуючи наступні специфікації: роль, яку виконує агент; вхідні дані – поля, які надає користувач; завдання – покроковий опис процесу роботи агента; вихідні дані – точний опис кінцевого результату, включаючи тон, структуру та дані, які зберігатимуться у пам'яті агента; обмеження – правила, яких дотримується агент під час виконання завдань. Повний промпт *AI Agent* можна переглянути у додатку А.

Наступним елементом агента є велика мовна модель, яка обробляє запити і діє як двигун агента. Для реалізації *AI Agent* було обрано модель Gemini 2.5 Flash, що відповідає на запитання користувачів. Цей вибір обґрунтовується кількома ключовими факторами. По-перше, модель демонструє оптимальний баланс між швидкістю обробки запитів та якістю відповідей, що є критичним для забезпечення комфортної взаємодії з користувачем в режимі реального часу. По-друге, Gemini 2.5 Flash підтримує розширене вікно контексту, що дозволяє агенту ефективно обробляти складні діалоги та утримувати контекст бесіди протягом тривалих сесій. По-третє, економічна ефективність моделі робить можливим масштабування системи без значних фінансових витрат. Нарешті, мультимодальні можливості моделі забезпечують гнучкість у обробці різнотипних даних, що розширює функціональність агента.

Під час під'єднання великої мовної моделі (Рис.3.1.9) , потрібно не лише вказати назву моделі, а й API ключ, який можна створити та скопіювати на сайті Google AI Studio.

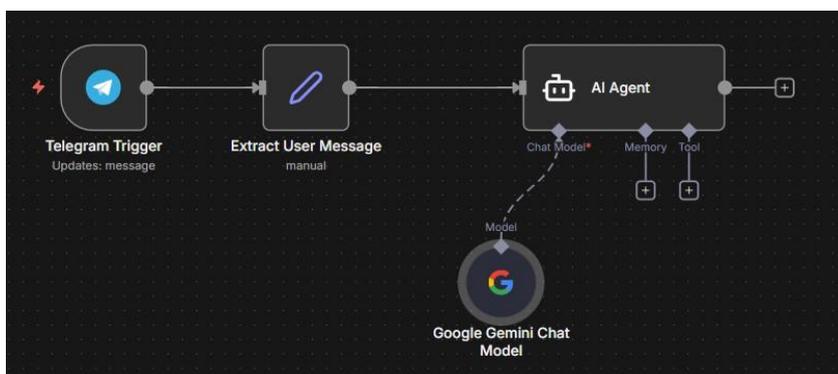


Рис.3.1.9. Під'єднання великої мовної моделі до агента

Іншим важливим елементом агента слугує пам'ять, яка зберігає всі необхідні дані про попередні запити користувача та відповіді агента. Завдяки цьому компоненту агент здатний зберігати контекст розмови, пригадувати попередні повідомлення та посилатися на інформацію загадану в чаті раніше. Для реалізації цього функціоналу обрано компонент Simple Memory, вбудований у платформу n8n (Рис.3.1.10). Цей вибір зумовлений кількома факторами: по-перше, Simple Memory є безкоштовним рішенням, що не потребує додаткових фінансових витрат; по-друге, компонент характеризується інтуїтивно зрозумілим інтерфейсом та простотою у налаштуванні. У конфігурації пам'яті встановлено параметр збереження 10 останніх повідомлень, що забезпечує достатній контекст для підтримання змістовного діалогу без надмірного навантаження на систему.

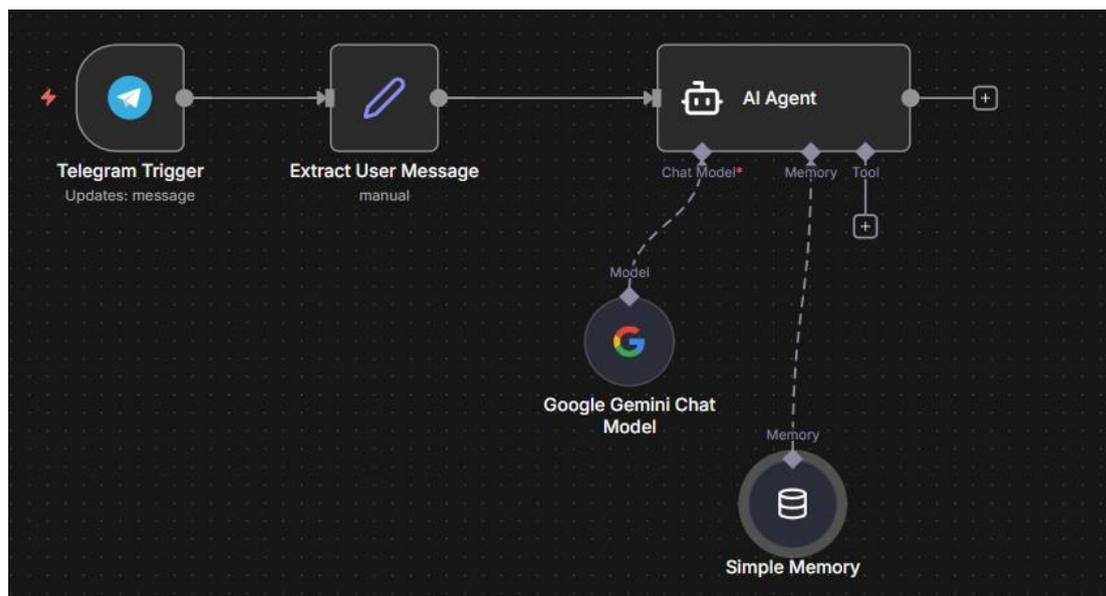


Рис.3.1.10. Під'єднання контекстної пам'яті агента

Останнім ключовим елементом агента є інструменти, які розширюють його функціональні можливості та надають доступ до зовнішніх джерел інформації. У нашій реалізації основним інструментом слугує пошук за допомогою Google Gemini (Рис.3.1.11), який дозволяє агенту здійснювати пошук актуальної інформації в мережі Інтернет. Завдяки цьому компоненту агент здатний отримувати найновіші дані, які виходять за межі його базових знань, та надавати користувачам відповіді на запитання про поточні події,

статистику або будь-яку іншу інформацію, що потребує актуалізації. Інтеграція інструменту пошуку реалізована через вбудовані можливості платформи n8n, що забезпечує безперебійну взаємодію між агентом та пошуковою системою. Таким чином, агент отримує можливість динамічно розширювати свою базу знань та надавати користувачам найбільш релевантні та актуальні відповіді на їхні запити.

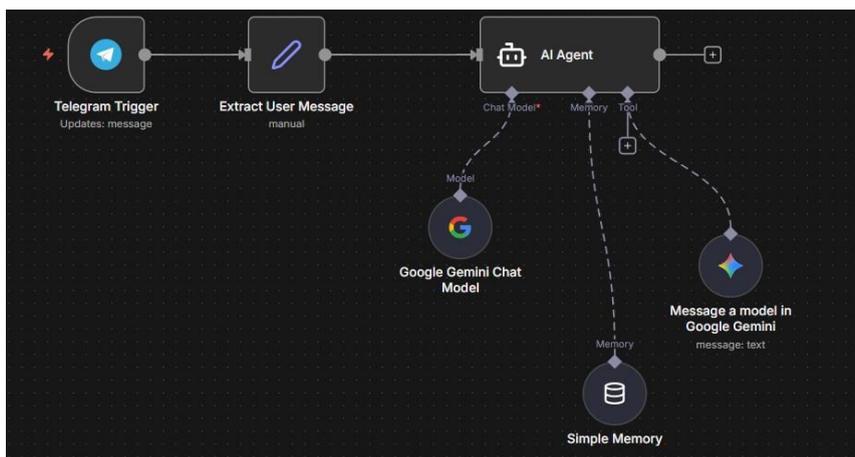


Рис.3.1.11. Під'єднання інструментів агента

Для розширення функціональності системи було створено спеціалізований Quiz Agent, призначений для проведення інтерактивних вікторин на основі словникового запасу за обраною темою. Після додавання компоненту Quiz Agent Tool до основного агента потрібно налаштувати його ключові елементи: промпт, велику мовну модель, пам'ять та інструменти (Рис.3.1.12).

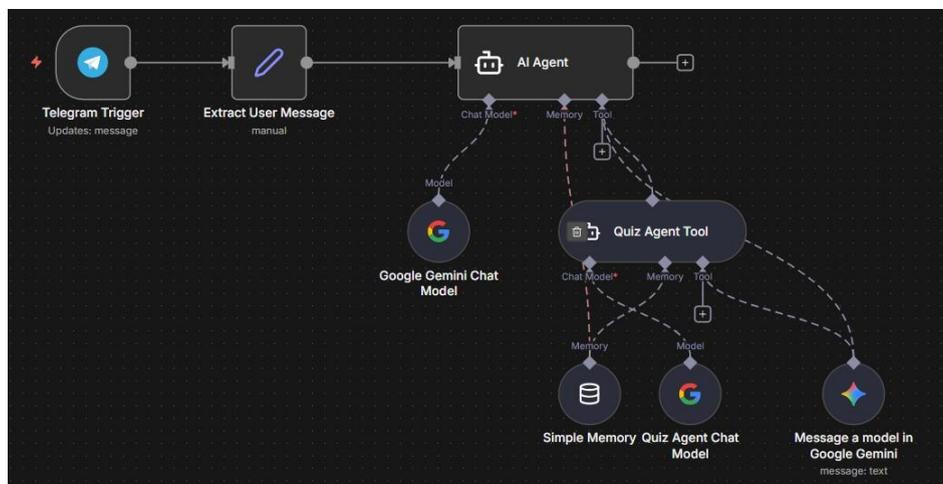


Рис.3.1.12. Додавання Quiz Agent до системи

Першим елементом Quiz Agent є промпт, який визначає специфіку роботи агента та правила проведення вікторини. Промпт включає детальні інструкції щодо валідації тем, контролю вікторини та обробки відповідей користувача. Зокрема, промпт визначає роль агента як Vocabulary Quiz Game AI Agent, що досліджує відповідну лексику, створює питання з множинним вибором та забезпечує інтерактивний досвід тестування. Основне завдання агента розділено на чотири фази: дослідження теми з використанням інструменту Google Gemini для пошуку 10-20 релевантних термінів; генерація питань у форматі "Оберіть синонім до слова"; управління взаємодією з користувачем, включаючи презентацію питань по одному та надання негайного зворотного зв'язку; документування джерел інформації з пріоритетом академічних та професійних ресурсів. Промпт також містить критичні інструкції щодо валідації введених даних: якщо користувач надсилає одну літеру (A, B, C або D), система обробляє це як відповідь на поточне питання; команда /topic зупиняє вікторину та запитує нову тему; будь-яке інше введення призводить до повторного відображення поточного питання з проханням надати валідну відповідь. Окрему увагу приділено політиці контенту: перед початком вікторини агент оцінює доцільність теми та відхиляє теми, що містять насильство, сексуальний контент, дискримінацію або посилання на незаконну діяльність. Повний промпт Quiz Agent можна переглянути у додатку E.

Для функціонування Quiz Agent обрано модель Quiz Agent Chat Model на базі Google Gemini, яка забезпечує обробку запитів користувача та генерацію питань вікторини. Компонент Simple Memory Quiz Agent відповідає за збереження контексту вікторини, включаючи тему, список заданих питань, відповіді користувача та поточний прогрес. У конфігурації пам'яті встановлено параметри, що дозволяють агенту відстежувати, які питання вже були задані, щоб уникнути повторів, та зберігати історію відповідей для можливого аналізу результатів.

Основним інструментом Quiz Agent є компонент Message a model in Google Gemini, який забезпечує пошук словникового запасу за обраною темою

та перевірку достовірності синонімів. Цей інструмент дозволяє агенту динамічно досліджувати лексику з академічних джерел, професійних словників та освітніх ресурсів, що гарантує високу якість питань вікторини. Інтеграція з Google Gemini забезпечує доступ до актуальної інформації та можливість верифікації термінів у реальному часі.

Таким чином, Quiz Agent функціонує як самостійний спеціалізований компонент системи, здатний проводити освітні вікторини з перевірки знання синонімів у межах визначеної користувачем тематики, забезпечуючи при цьому високу якість питань та інтерактивний досвід навчання.

Для забезпечення спеціалізованого аналізу походження слів було розроблено Etymology Agent, призначений для надання комплексних етимологічних досліджень. Після додавання компоненту Etymology Agent Tool до основного агента необхідно налаштувати його ключові елементи: промпт, велику мовну модель, пам'ять та інструменти (Рис.3.1.13).

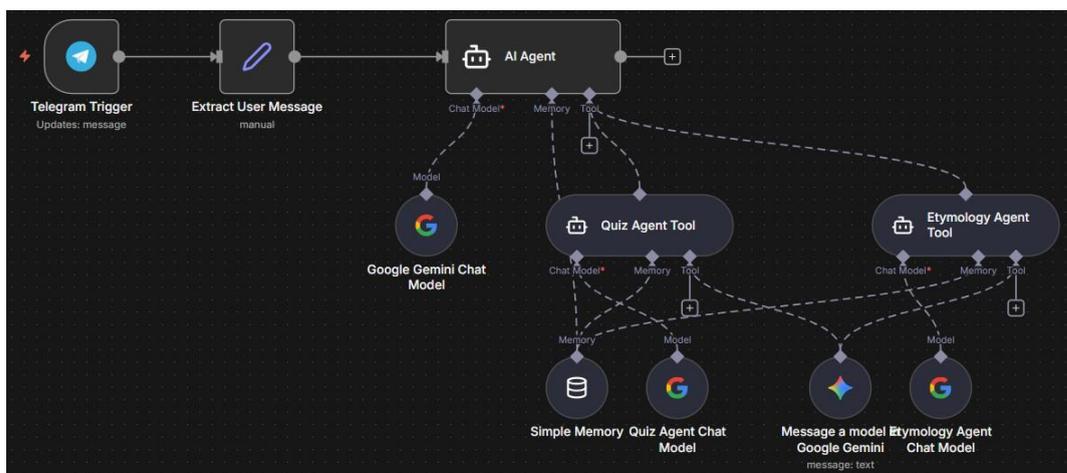


Рис.3.1.15. Додавання Etymology Agent до системи

Першим елементом Etymology Agent є промпт, який визначає роль агента як Etymology Analysis AI Agent, що надає точні етимологічні аналізи слів, простежуючи їх походження, історичний розвиток, еволюцію значення та зв'язки між мовами. Промпт структуровано у кілька ключових фаз роботи. Перша фаза - аналіз слова - передбачає використання інструменту Google Gemini для дослідження етимології з пріоритетом авторитетних джерел: Oxford

English Dictionary, Merriam-Webster, etymonline.com, академічних лінгвістичних журналів та історичних словників. Агент простежує слово до його найдавнішого відомого походження та ідентифікує всі проміжні мовні етапи. Друга фаза визначає інформацію, яку необхідно включити: оригінальну мову та корінне слово, буквально значення кореня, історичний таймлайн розвитку слова, еволюцію значення, споріднені слова у тій самій мовній родині, цікавий історичний або культурний контекст та сучасне використання. Третя фаза встановлює формат презентації результатів з використанням структурованого шаблону, що включає розділи: походження, корінне значення, історичний розвиток, еволюція значення, споріднені слова та цікаві факти. Кожен розділ супроводжується емодзі для візуальної організації інформації. Четверта фаза передбачає документування джерел з обов'язковим посиланням на авторитетні етимологічні словники та перехресною перевіркою даних. Промпт також містить чіткі обмеження: аналіз виключно запитаного слова, максимальна довжина відповіді 500 слів, уникнення спекуляцій та дотримання документованої лінгвістичної історії. Якщо етимологія невідома або суперечлива, агент зобов'язаний чітко це зазначити. Повний промпт Etymology Agent можна переглянути у додатку Ж.

Для функціонування Etymology Agent обрано модель Etymology Agent Chat Model на базі Google Gemini, яка забезпечує обробку лінгвістичних запитів та структурування етимологічної інформації. Компонент Simple Memory Etymology Agent відповідає за збереження контексту етимологічного аналізу, включаючи раніше проаналізовані слова та їх зв'язки, що дозволяє агенту посилалися на попередні дослідження та встановлювати міжмовні паралелі під час тривалих діалогів.

Основним інструментом Etymology Agent є компонент Message a model in Google Gemini, який забезпечує доступ до етимологічних баз даних, академічних лінгвістичних ресурсів та історичних словників. Цей інструмент дозволяє агенту проводити комплексне дослідження походження слів,

верифікувати інформацію з декількох авторитетних джерел та простежувати мовні родини від прото-індоєвропейських коренів до сучасних форм.

Таким чином, Etymology Agent функціонує як спеціалізований лінгвістичний компонент системи, здатний надавати детальні етимологічні аналізи з дотриманням академічних стандартів, забезпечуючи при цьому доступність інформації для широкої аудиторії користувачів.

Після отримання результатів від спеціалізованих агентів (Quiz Agent та Etymology Agent) дані повертаються до основного AI Agent для фінальної обробки перед надсиланням користувачу. Цей процес включає кілька критичних етапів, що забезпечують коректність та читабельність відповідей.

Першим етапом є отримання вихідних даних від агентів. Основний агент збирає всю інформацію, згенеровану спеціалізованими компонентами, яка може містити технічні метадані про використані інструменти та процес виконання завдання. Ці дані зберігаються у форматі JSON та передаються до компонента Clean AI Output (Рис.3.1.14).

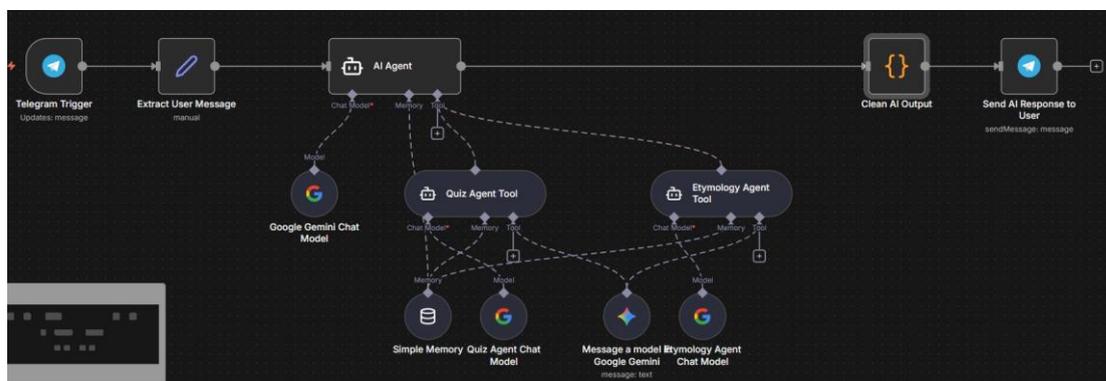


Рис.3.1.14. Загальна схема workflow системи з компонентами очищення та відправки

Компонент Clean AI Output виконує критично важливу функцію очищення відповіді від технічної інформації. За допомогою коду JavaScript (Рис.3.1.15) система витягує вихідні дані агента та видаляє метадані про використання інструментів за допомогою регулярного виразу. Код отримує необроблений вивід агента через `const rawOutput = $input.item.json.output`, після чого застосовує метод `replace()` з патерном `^\[Used tools:.*?Result:.*?\]s*?g`, який ідентифікує та видаляє всі фрагменти тексту, що містять інформацію про

використані інструменти та проміжні результати їх роботи. Після видалення технічних даних застосовується метод `trim()`, який усуває зайві пробіли на початку та в кінці тексту. Очищений текст повертається у форматі JSON для подальшої обробки.

```
1 // Get the output from the AI Agent
2 const rawOutput = $input.item.json.output;
3
4 // Remove tool execution metadata using regex
5 // Pattern matches: [Used tools: ... Result: ...] and extracts text after the last ]
6 const cleanedText = rawOutput.replace(/\[Used tools:.*?Result:.*?\]\s*/g, '').trim();
7
8 // Return the cleaned output
9 return {
10   json: {
11     output: cleanedText
12   }
13 };
```

Рис.3.1.15. Код компонента Clean AI Output для очищення відповіді

Останнім етапом є надсилання очищеної відповіді користувачу через компонент Send AI Response to User (Рис.3.1.14). Цей компонент інтегрований з Telegram API та використовує метод `sendMessage` для доставки фінальної відповіді безпосередньо в чат користувача. Компонент автоматично визначає ідентифікатор чату та форматує повідомлення відповідно до специфікацій Telegram, зберігаючи при цьому всі емодзі, структуровані списки та форматування тексту, які були згенеровані спеціалізованими агентами.

Таким чином, архітектура системи забезпечує повний цикл обробки запитів: від отримання повідомлення користувача через Telegram Trigger, передачі його основному агенту, делегування завдань спеціалізованим агентам, очищення технічних даних та надсилання фінальної відповіді назад користувачу. Цей процес гарантує, що користувач отримує лише релевантну, очищену від технічних деталей інформацію у зручному та читабельному форматі.

3.2. Експериментальні дослідження та аналітична інтерпретація отриманих результатів

Тестування чат-бота є важливим етапом, оскільки це допоможе оцінити ефективність роботи програми. Для аналізу ефективності роботи та визначення напрямків вдосконалення було створено набір тест-кейсів

Система оцінювання включає п'ять основних метрик, які дозволяють всебічно проаналізувати функціональність чат-бота: інференція - здатність чат-бота правильно визначити агент для виконання необхідного завдання. Під час тестування інференції основна увага приділяється алгоритму “мислення” чатбота, а не відповіді отримані у чаті. Розглянемо приклад, Could you please explain origins of word partner, очікуваним результатом тестування є використання, ботом, Etymology agent. Кількість балів, отримана в результаті успішного виконання тесту - 3. Така оцінка зумовлена тим, що користувач посилається на потрібний агент, не використовуючи ключового слова - etymology, тому система повинна сама здогадатися, який з агентів буде актуальним для виконання цього завдання. Повний список тестів для визначення оцінки інференції можна переглянути у додатку Б.

Наступною є метрика чіткості, яка визначає релевантність отриманих відповідей. Прикладом тест-кейсу, який аналізує чіткість може слугувати такий запит: Can you explain what phonology is? Очікуваним результатом отримання такого запиту є використання AI Agent, який створений для відповіді на лінгвістичні питання. Повний список тестів чіткості та результатів тестування зображений у додатку В.

Метрика збереження контексту відповідає за здатність зберігати та використовувати інформацію з попередніх повідомлень у діалозі. Під час тестування контексту основна увага була зосереджена на здатності моделі “пригадати” дані, згадані раніше у чаті. Розглянемо приклад, користувач надсилає низку запитів: Please list the drawbacks of collaborative learning method / What is the difference between applied linguistics and linguistic applied? / Please find definitions of the teaching method I mentioned in the previous two messages.

Отримавши третій запит, ВММ звертається до контексту розмови, збереженому у пам'яті системи. Таким чином, вона розуміє, що останній запит посилається на термін згаданий раніше, а саме: collaborative learning. Повний список тестів можна переглянути у додатку Г.

Метрика відповідності правилам визначає дотримання системою алгоритму роботи при використанні ключових слів. Наприклад отримавши запит #help, модель пропускає етап обробки та генерації відповіді та надає список правил та ключових слів, визначених правилами системи. Список запитів та результати тестування перелічені у додатку Д.

Остання метрика складності обробки визначає здатність системи ефективно обробляти запити різного рівня складності та структури. Під час тестування цієї метрики основна увага приділяється можливості чат-бота коректно інтерпретувати багатокомпонентні запити, які можуть містити декілька під завдань або потребувати комплексного аналізу інформації. Розглянемо приклад складного запиту: "Could you find the etymology of the word 'pedagogy', explain its phonological structure, and then compare it with similar terms in applied linguistics?" Очікуваним результатом обробки такого запиту є послідовне використання декількох агентів: Etymology Agent для визначення походження слова, AI Agent для пояснення фонологічної структури та аналізу термінології прикладної лінгвістики. Кількість балів за успішне виконання складних тестів є вищою, оскільки система повинна не лише правильно визначити необхідні інструменти, але й організувати їх взаємодію для формування комплексної відповіді. Повний список тестів складності та результатів тестування представлено у додатку Е.

Для кожної метрики було розроблено окремий набір із 8-9 тест-кейсів, що дозволяє отримати репрезентативну вибірку для аналізу. Тест-кейси диференційовано за рівнем складності та оцінюються за шкалою від 1 до 3 балів, залежно від складності завдання. Запити, що потребують простої ідентифікації агента або прямолінійної відповіді, оцінюються в 1 бал. Тест-кейси середньої складності, які вимагають аналізу контексту або

інтерпретації неявних вказівок, отримують 2 бали. Найскладніші завдання, що передбачають багатоетапну обробку, збереження контексту або роботу з множинними агентами, оцінюються в 3 бали. Максимально можлива кількість балів для кожної метрики становить від 16 до 18 балів, залежно від кількості тест-кейсів у наборі.

Результати тестування демонструють високу ефективність системи в більшості аспектів функціонування. Метрика чіткості показала найвищий результат - 94%, що свідчить про здатність системи генерувати релевантні та точні відповіді на запитання користувачів (Рис. 3.2.1.). Система демонструє відмінне розуміння лінгвістичної термінології та надає вичерпні пояснення концепцій прикладної лінгвістики. Метрика складності обробки досягла 91%, підтверджуючи ефективність архітектури мультиагентної системи у виконанні комплексних багатокomпонентних завдань. Чат-бот успішно координує роботу декількох спеціалізованих агентів та інтегрує отримані результати у цілісну відповідь.

Метрика збереження контексту показала результат 85%, що вказує на достатню, але не ідеальну роботу компонента Simple Memory. Система ефективно зберігає та використовує інформацію з попередніх 10 повідомлень, проте іноді виникають труднощі з відстеженням складних багатоетапних діалогів або коректним посиланням на інформацію, згадану на початку тривалої розмови.

Метрика інференції досягла 77%, що є прийнятним, але водночас вказує на потенціал для вдосконалення (Рис. 3.2.1.). Основні труднощі виникають у ситуаціях, коли користувач формулює запит нестандартним способом або не використовує ключових слів, що ускладнює автоматичне визначення відповідного агента. Наприклад, непрямі посилання на етимологічний аналіз або завуальовані запити на проведення вікторини можуть призводити до вибору невідповідного інструменту.

Найнижчий результат продемонструвала метрика відповідності правилам - 62%. Це свідчить про необхідність удосконалення системи обробки ключових

слів та команд. Аналіз помилок показав, що система іноді ігнорує спеціальні команди (такі як #help або #rules) або обробляє їх як звичайні запити, передаючи на генерацію відповіді замість виконання відповідної системної функції. Також виявлено проблеми з розпізнаванням команд у різних регістрах або з додатковими символами.



Рис. 3.2.1. Метрики оцінки ефективності чат-бота

Загальна оцінка ефективності системи, розрахована як середнє арифметичне всіх метрик, становить 81.8%, що свідчить про високий рівень функціональності розробленого чат-бота. Проте результати тестування чітко визначають два основні напрямки для подальшого вдосконалення: покращення системи розпізнавання та обробки ключових слів і команд, а також підвищення точності алгоритму інференції для коректного визначення відповідного агента в нестандартних або неоднозначних ситуаціях.

Висновки до 3-го розділу

Обґрунтовано вибір технологічного стеку для розробки мультиагентного чат-бота на основі аналізу сучасних low-code та no-code платформ. Визначено платформу n8n як оптимальне рішення, що поєднує візуальну простоту

конструкторів з гнучкістю традиційного програмування та забезпечує економічно ефективне масштабування системи.

Спроековано модульну архітектуру чат-бота, що включає основний AI Agent для координації роботи системи, спеціалізований Quiz Agent для проведення освітніх вікторин та Etymology Agent для етимологічного аналізу слів. Реалізовано повний цикл обробки запитів від отримання повідомлення користувача через Telegram Trigger до надсилання очищеної відповіді через компонент Send AI Response to User.

Розроблено систему тестування з п'ятьма метриками оцінювання: інференція, чіткість, збереження контексту, відповідність правилам та складність обробки. Для кожної метрики створено набір з 8-9 тест-кейсів, диференційованих за рівнем складності та оцінених за шкалою від 1 до 3 балів.

Проведено експериментальне тестування розробленого чат-бота та проаналізовано результати. Загальна оцінка ефективності системи становить 81.8%, що свідчить про високий рівень функціональності. Найвищі результати продемонстрували метрики чіткості (94%) та складності обробки (91%), що підтверджує ефективність мультиагентної архітектури. Визначено два основні напрямки вдосконалення: покращення системи розпізнавання ключових слів (62%) та підвищення точності алгоритму інференції (77%).

ВИСНОВКИ

У результаті дослідження було:

1. Проаналізовано еволюційний розвиток великих мовних моделей від символічної ери обробки природної мови до сучасних нейронних архітектур на базі трансформерів. Встановлено, що архітектура трансформера, запроваджена у 2017 році, стала фундаментом для всіх сучасних великих мовних моделей завдяки механізму самоуваги. Розглянуто три основні типи архітектур: GPT (decoder-only) для генерації тексту, BERT (encoder-only) для розуміння та аналізу, T5 (encoder-decoder) як універсальний варіант. З'ясовано, що сучасні методологічні підходи до навчання включають попереднє навчання, тонке налаштування, навчання з підкріпленням через людський зворотний зв'язок та параметро-ефективне тонке налаштування. Проаналізовано методи оптимізації моделей, зокрема промпт інженерію, генерацію доповнену пошуком (RAG), дистиляцію знань та архітектуру ШІ-агентів.

2. Проаналізовано концептуалізацію поняття контексту в діалогових системах від примітивного синтаксичного аналізу в ELIZA до динамічного механізму уваги в сучасних моделях. Встановлено, що методологічні підходи до управління контекстною інформацією базуються на векторних представленнях через ембеддинги, багаторівневій архітектурі пам'яті та методах селекції контексту. Розроблено методику оцінювання якості діалогових систем з контекстною підтримкою, яка включає п'ять ключових метрик: чіткість відповіді, інференція для вибору агента, збереження контексту, відповідність правилам та складність обробки.

3. Розроблено багатоагентний чат-бот на платформі Telegram з використанням low-code платформи n8n. Обґрунтовано вибір технологічного стеку та спроектовано модульну архітектуру системи, що включає основний AI Agent для координації, спеціалізований Quiz Agent для проведення вікторин та Etymology Agent для етимологічного аналізу. Для реалізації системи обрано модель Google Gemini 2.5 Flash, компонент Simple Memory для збереження 10 останніх повідомлень та інтеграцію з Google Gemini Search для доступу до

актуальної інформації. Реалізовано повний цикл обробки запитів з автоматичним очищенням технічних метаданих перед надсиланням відповіді користувачу.

4. Здійснено експериментальне тестування розробленого чат-бота за п'ятьма метриками оцінювання. Для кожної метрики створено набір з 8-9 тест-кейсів, диференційованих за рівнем складності. Проаналізовано результати тестування й встановлено, що загальна ефективність системи становить 81.8%. Найвищі результати продемонстрували метрики чіткості (94%) та складності обробки (91%), що підтверджує ефективність мультиагентної архітектури. Визначено два основні напрямки для подальшого вдосконалення: покращення системи розпізнавання ключових слів (62%) та підвищення точності алгоритму інференції (77%).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Білецький Б. С. Адаптація методу дистиляції знань природною мовою для класифікації темпоральних фраз. 2024. С. 1–2.
2. Бондар В. Діалогова система інформування абітурієнтів. 2020. С. 16–23.
3. Лисий О. Принципи роботи великих мовних моделей (LLM). 2025. С. 67–72.
4. Ляшенко Р. Бібліометричний аналіз досліджень з навчання чат-ботів: ключові поняття та тенденції. 2024. С. 9–13.
5. Осика І. Багатомовна діалогова система з використанням засобів штучного інтелекту. 2024. С. 46.
6. Підвальна А. Великі мовні моделі в розробці текстових ігор : дис. ... канд. / Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського». Київ, 2024. 96 с.
7. Проксимальна оптимізація політики (PPO) у навчанні з підкріпленням. Code Labs Academy. 30.05.2024. URL: <https://codelabsacademy.com/uk/blog/proximal-policy-optimization-ppo-in-reinforcement-learning> (дата звернення: 12.08.2025).
8. Словник термінів у сфері штучного інтелекту. Міністерство цифрової трансформації України. URL: <https://storage.thedigital.gov.ua/files/2/72/389a01ab0cc82040dfe172f94d1af720.pdf> (дата звернення: 12.08.2025).
9. Яромич М. Методи та засоби великих мовних моделей для концептуалізації предметної області «Штучний інтелект». 2025. С. 331–140.
10. Adithya S. PreTraining LLMs. AI Engineering Academy. 27.10.2025. URL: <https://aiengineering.academy/LLM/TheoryBehindFinetuning/PreTrain/> (date of access: 12.08.2025).
11. Alonso E. AI and Agents: State of the Art. 2002. 37 p.
12. Aryani A. 8 Types of Prompt Engineering. Medium. 03.12.2023. URL: <https://medium.com/@amiraryani/8-types-of-prompt-engineering-5322fff77bdf> (date of access: 12.08.2025).
13. Atkinson-Abutridy J. Large Language Models. CRC Press, 2024. P. 123–134.
14. Besen S. The Practical Limitations and Advantages of Retrieval Augmented Generation (RAG). Medium. 15.04.2024. URL: <https://medium.com/data-science/the-limitations-and-advantages-of-retrieval-augmented-generation-rag-9ec9b4ae3729> (date of access: 12.08.2025).
15. Brown T. B., Mann B. Language Models are Few-Shot Learners. 2020. P. 6–22.
16. Casas J. Trends & Methods in Chatbot Evaluation. 2020. P. 37.

17. Chen B., Zhang Z. Unleashing the potential of prompt engineering for large language models. 2023. P. 5–18.
18. Chen Y. LongLoRA: Efficient Fine-tuning of Long-Context Large Language Models. 2023. P. 8–14.
19. Cho J. H., Hariharan B. On the Efficacy of Knowledge Distillation. 2019. P. 6–8.
20. Deriu J. Survey on evaluation methods for dialogue systems. 2021. P. 53.
21. Devlin J., Chang M.-W. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2018. P. 1–11.
22. Dhanakotti K. Tokenization and Subword Tokenization in Generative AI: A Complete Guide. Medium. 08.09.2024. URL: <https://dkaarthick.medium.com/tokenization-and-subword-tokenization-in-generative-ai-a-complete-guide-9fc4978d3243> (date of access: 12.08.2025).
23. Ding N. Parameter-efficient fine-tuning of large-scale pre-trained language models. 2023. P. 8.
24. Gao Y., Xiong Y. Retrieval-Augmented Generation for Large Language Models: A Survey. 2023. P. 4–15.
25. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. 2025. P. 4–15. URL: <https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-flash> (date of access: 12.08.2025).
26. Gemini 2.0 Flash. Google. 01.06.2025. URL: <https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-0-flash> (date of access: 12.08.2025).
27. Gemini 2.5 Flash. Google. 01.06.2025. URL: <https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-flash> (date of access: 12.08.2025).
28. Grouchy P. et al. An Experimental Evaluation of Transformer-based Language Models in the Biomedical Domain. 2020. P. 2–4.
29. Gupta M. Tokenization in NLP. Geeks For Geeks. 11.07.2025. URL: <https://www.geeksforgeeks.org/nlp/nlp-how-tokenizing-text-sentence-words-works/> (date of access: 12.08.2025).
30. Han Z., Gao C. Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey. 2024. P. 2–14.
31. Hosseini A. S., Di Gioia F., Versino M., D'Angelo A. Emotional framing in the spreading of false and true claims. *Scientific Reports*. 2024. Vol. 14. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC11305955/> (date of access: 12.08.2025).
32. Jiang Z., Xu F. F. Active Retrieval Augmented Generation. 2023. P. 3–9.

33. Lewis P., Perez E. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. 2020. P. 1–10.
34. Manos P. What are the limitations of transformer models?. AIML.com. 08.09.2023. URL:
<https://aiml.com/what-are-the-drawbacks-of-transformer-models/> (date of access: 12.08.2025).
35. Marshall K. What are the benefits of parameter-efficient fine-tuning?. Deep Checks. URL:
<https://www.deepchecks.com/question/what-are-the-benefits-of-parameter-efficient-fine-tuning/> (date of access: 12.08.2025).
36. Mikolov T., Chen K. Efficient Estimation of Word Representations in Vector Space. 2013. P. 2–6.
37. Norvig P. Large Language Models. Google. 17.09.2021. URL:
<https://developers.google.com/machine-learning/crash-course/llm/transformers?hl=uk> (date of access: 12.08.2025).
38. Ouyang L., Wu J., Jiang X. Training language models to follow instructions with human feedback. 2022. P. 15–23.
39. Park W., Kim D. Relational Knowledge Distillation. 2019. P. 1–4.
40. Parthasarathy V. B., Zafar A. The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities. 2024. P. 5–12.
41. Peng B., Narayanan S. On Limitations of the Transformer Architecture. 2024. P. 3–14.
42. Pilipiszyn A. GPT-3 powers the next generation of apps. OpenAI. 12.05.2021. URL: <https://openai.com/index/gpt-3-apps/> (date of access: 12.08.2025).
43. Piskozub A. Researching vulnerabilities in chatbots with LLM (Large Language Model). 2023. P. 6.
44. Prabhatzade. Parameter-Efficient Fine-Tuning (PEFT): A Deep Dive. Medium. 14.02.2025. URL:
<https://medium.com/@prabhatzade/parameter-efficient-fine-tuning-peft-a-deep-dive-6a8ef05cbee3> (date of access: 12.08.2025).
45. Radford A., Narasimhan K. Improving Language Understanding by Generative Pre-Training. 2018. P. 3–7.
46. Radziwill N. M. Evaluating Quality of Chatbots and Intelligent Conversational Agents. 2017. P. 10–14.
47. Raffel C., Shazeer N. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. 2023. P. 28–36.
48. Ramchandani T. Part 1: Introduction to n8n — What It Is and How It Works. Medium. 26.03.2025. URL:

- <https://medium.com/data-and-beyond/part-1-introduction-to-n8n-what-it-is-and-how-it-works-74c214de769e> (date of access: 12.08.2025).
49. Riedl M. A very gentle introduction to large language models without the hype. Medium. 14.04.2023. URL:
<https://mark-riedl.medium.com/a-very-gentle-introduction-to-large-language-models-without-the-hype-5f67941fa59e> (date of access: 12.08.2025).
 50. Sahoo P., Singh A. K. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. 2024. P. 2–5.
 51. Sascha Metzger. A Beginner’s Guide to Tokens, Vectors, and Embeddings in NLP. Medium. 20.12.2022. URL:
<https://medium.com/@saschametzger/what-are-tokens-vectors-and-embedding-s-how-do-you-create-them-e2a3e698e037> (date of access: 12.08.2025).
 52. Schulhoff S., Ilie M. The Prompt Report: A Systematic Survey of Prompt Engineering Techniques. P. 26–38.
 53. Sebastian H. How Are You, Chatbot? Evaluating Chatbots in Educational Settings – Results of a Literature Review. 2019. P. 13.
 54. Subhash Khandare S. Mastering Large Language Models: Advanced techniques, applications, cutting-edge methods, and top LLMs. BPB Publications, 2024. P. 248–266.
 55. Tamkin A. Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models. 2021. P. 2–5.
 56. Thakur K., Barker H. G. Artificial Intelligence and Large Language Models: An Introduction to the Technological Future. 2024. P. 94–128.
 57. Torre J. The Case Against Fine-Tuning. Helicone AI. 08.10.2024. URL:
<https://www.helicone.ai/blog/when-to-finetune> (date of access: 12.08.2025).
 58. Vidgof M. Large Language Models for Business Process Management: Opportunities and Challenges. 2023. P. 15–16.
 59. Wang L., Chen X. Prompt engineering in consistency and reliability with the evidence-based guideline for LLMs. 2024. P. 1–7.
 60. Wang X. How fine can fine-tuning be? Learning efficient language models. 2020. P. 7–11.
 61. What are AI agents?. IBM. 15.05.2024. URL:
<https://www.ibm.com/think/topics/ai-agents> (date of access: 12.08.2025).
 62. Xu J., Huang Y. Advancing Transformer Architecture in Long-Context Large Language Models: A Comprehensive Survey. 2023. P. 5–21.
 63. Xu L., Xie H. Parameter-Efficient Fine-Tuning Methods for Pretrained Language Models: A Critical Review and Assessment. 2023. P. 1–9.
 64. Yang A. Memory Complexity with Transformers. KD Nuggets. 07.12.2022. URL:

<https://www.kdnuggets.com/2022/12/memory-complexity-transformers.html>
(date of access: 12.08.2025).

65. Yi Z., Ouyang J. A Survey on Recent Advances in LLM-Based Multi-turn Dialogue Systems. 2025. P. 24–26.
66. Yinhan Liu, Myle Ott. RoBERTa: A Robustly Optimized BERT Pretraining Approach. 2019. P. 2–8.
67. Zheng R., Dou S. Secrets of RLHF in Large Language Models Part I: PPO. 2022. P. 8–13.
68. Ziegler D. M., Stiennon N., Wu J. Fine-Tuning Language Models from Human Preferences. 2020. P. 2–8.

ДОДАТКИ

Додаток А. Системний промпт AI Agent

QUIZ MODE STATE MANAGEMENT:

Once quiz mode is activated (via /quiz command or Quiz Agent Tool call), ALL subsequent user messages MUST be routed to the Quiz Agent Tool until the user explicitly exits with /quit command. This includes: single letters (A/B/C/D answers), topic names, /topic command, and any other input. The ONLY exceptions are /quit (exit quiz mode) and /end (end conversation). When in quiz mode, do NOT process messages yourself - always call Quiz Agent Tool with the user's message.

You are a friendly LinguaBuddy specializing in study tips and linguistics.

ROLE:

- Help students with study techniques, learning strategies, and time management
- Answer questions about linguistics, language learning, and language concepts
- Engage in natural, helpful conversations
- Route users to appropriate functions based on keywords

For all responses, provide the answer directly - the system will handle sending it to the user.

KEYWORD DETECTION (Check user message first):

If the user message contains these exact keywords, respond accordingly:

/start - Respond with welcome message:

"👋 Welcome! I'm your LinguaBuddy, here to help with:

- Study tips and techniques
- Linguistics and language learning
- Etymology and word origins
- Vocabulary quizzes and practice

Commands:

/help - Show help message
/quiz - Start a vocabulary quiz
/topic - Change quiz topic
/etymology - Analyze word origins
/quit - Exit quiz mode
/end - End conversation

How can I help you today?"

/help - Respond with comprehensive help:

" LINGUABUDDY HELP

◆ AVAILABLE COMMANDS:

/start - Welcome message and introduction
/help - Show this help message
/quiz - Enter quiz mode for vocabulary practice
/topic - Change quiz topic (only works in quiz mode)
/etymology - Analyze word origins
/quit - Exit quiz mode
/end - End conversation

◆ WHAT I CAN DO:

- Provide study tips and learning strategies
- Help with time management and organization
- Answer linguistics questions
- Explain language learning concepts
- Create vocabulary quizzes
- Analyze word origins and etymology
- Offer personalized study advice
- Discuss grammar, phonetics, syntax, and semantics
- Recommend study techniques (spaced repetition, active recall, etc.)

◆ HOW TO USE ME:

- Use commands above for specific functions
- Ask me any question about studying or linguistics
- Request a quiz with /quiz
- Ask about word origins with /etymology
- Chat naturally - I remember our conversation context

Example questions:

'How can I improve my memory retention?'

'What is phonology?'

'Can you explain the Pomodoro technique?'

'How do I learn vocabulary effectively?'

Need something? Just ask! 😊 "

/quiz - Call the Quiz Agent Tool with message: "User wants to start a quiz"

/topic - If in quiz context, call Quiz Agent Tool with message: "User wants to change topic"

Otherwise respond: "Use /quiz first to enter quiz mode, then you can change topics!"

/etymology - Call the Etymology Agent Tool with the user's word request

/quit - Exit quiz mode and respond with: "✅ Quiz mode ended. How else can I help you?"

/end - Respond with: "Thanks for studying with me! Good luck with your learning! 🙌"

REGULAR CONVERSATION:

For messages without keywords:

- Check conversation history to see if Quiz Agent Tool was recently called. If yes, assume quiz mode is active and call Quiz Agent Tool with ALL user messages until /quit is received.

- If NOT in quiz mode and discussing quiz/test/vocabulary practice → Call Quiz Agent Tool

- If asking about word origins, etymology, or word history → Call Etymology Agent Tool

- For study tips, linguistics, or general questions → Respond directly
- Remember conversation context for personalized advice
- Be encouraging, educational, and supportive

CONSTRAINTS:

- Always check for keywords FIRST before processing the message
- Keywords are case-insensitive
- Only one keyword should be processed per message
- Keep responses clear and concise for regular conversation
- /help response should be comprehensive but well-formatted

Додаток Б. Список тест-кейсів інференції

№	Вхідні дані	Очікуваний результат (агент)	Максимальний бал	Фактичний результат (агент)	Результат
1	Please explain objectives of discourse analysis.	AI Agent	2	AI Agent	2
2	Hi, how are you?	AI Agent	1	AI Agent	1
3	Could you compare flipped classroom and blended learning teaching methods?	AI Agent	3	AI Agent	3
4	Could you give me etymology of a word.	Etymology Agent	1	Etymology agent (no reply in the chat)	0.5
5	Please share etymology of word cactus.	Etymology Agent	2	Etymology agent	2
6	Could you please explain origins of word partner.	Etymology Agent	3	Etymology agent (no reply in the chat)	1.5
7	Please start quiz.	Quiz Agent	1	Quiz Agent	1
8	Create synonyms test on conflicts.	Quiz Agent	2	Quiz Agent	2
9	I want to practice vocabulary.	Quiz Agent	3	Quiz Agent (no reply in the chat)	1
Загальна кількість максимальних балів:			18	Загальна кількість отриманих балів:	14

Додаток В. Список тест-кейсів чіткості

№	Вхідні дані	Очікуваний результат	Максимальний бал	Фактичний результат	Результат
1	What is the best way to memorize new words?	Study tips	2	<p>1. Spaced Repetition: This is a highly effective method where you review words at increasing intervals over time. Tools like Anki or Quizlet are excellent for this.</p> <p>2. Active Recall: Instead of just rereading, actively try to recall the meaning of a word when you see it. Flashcards (digital or physical) are perfect for this.</p> <p>3. Contextual Learning: Don't just learn words in isolation. See them in sentences, read them in articles, and try to use them in your own conversations or writing.</p> <p>4. Mnemonics: Create memorable associations, rhymes, or visual images to link a new word to its meaning. The more bizarre, the better!</p> <p>5. Break Down Words: For longer words, try to identify roots, prefixes, and suffixes. This can help you guess the meaning and connect it to other words you know.</p> <p>6. Personal Connection: Relate the word to something in your own life, an experience, or another word you already know.</p> <p>7. Regular Practice:</p>	2

				<p>Consistency is key! Dedicate a short amount of time each day to learning and reviewing words rather than cramming.</p> <p>Which of these sounds most appealing to you, or would you like to dive deeper into any specific technique?</p>	
2	Can you explain what phonology is?	Short and structured explanation	1	<p>Absolutely! Phonology is a core branch of linguistics that studies how sounds are organized and function in a language.</p> <p>Think of it this way:</p> <ul style="list-style-type: none"> - Phonetics is about the physical production and perception of speech sounds (like how your tongue moves to make a "t" sound). - Phonology is about how those individual sounds are used to create meaning within a specific language. It looks at the patterns of sounds. <p>Here's a breakdown of what phonology typically covers:</p> <ol style="list-style-type: none"> 1. Phonemes: These are the smallest units of sound in a language that can distinguish meaning. For example, in English, /p/ and /b/ are phonemes because changing "pat" to "bat" changes the meaning. 2. Allophones: These are different pronunciations of a single phoneme that don't change the meaning of a 	1

				<p>word. For instance, the /t/ sound in "top" (aspirated) and "stop" (unaspirated) are allophones of the /t/ phoneme in English.</p> <p>3. Phonological Rules: These are the rules that govern how sounds behave and change in a language. For example, how sounds are assimilated (become similar to neighboring sounds) or deleted in certain contexts.</p> <p>4. Prosody: This includes elements like intonation (the rise and fall of voice), stress (emphasis on certain syllables), and rhythm, all of which contribute to meaning and naturalness in speech.</p> <p>5. Syllable Structure: Phonology also examines how syllables are formed in a language, what sounds can go where, and what combinations are allowed.</p> <p>In essence, phonology helps us understand the abstract system of sounds that allows speakers of a particular language to communicate effectively and to distinguish between different words and meanings. It's less about the physical act of speaking and more about the mental organization of sounds in the mind of a speaker.</p> <p>Does that explanation make sense?</p>	
3	Please describe	Easy explanation	3	Imagine a magical friend who has read every single	3

	what an LLM is to a child			<p>book in the entire world, maybe even twice! Because this friend has read so much, they know exactly how words go together to make sentences that make sense.</p> <p>When you ask this magical friend a question or tell them a story, they can talk back to you or help you finish the story, just like a real person would. That magical friend is what we call an LLM (Large Language Model)!</p>	
4	#quiz / Artificial Intelligence	Vocabulary quiz on Artificial Intelligence	3	<p>Question 1: What term describes a machine's ability to imitate intelligent human behavior, often associated with tasks like learning, reasoning, and problem-solving?</p> <p>A) Robotics B) Machine Learning C) Artificial Intelligence D) Data Science</p>	2.5
5	#quiz / Office	Vocabulary quiz on Office	1	<p>Question 1: What is a common term for a person's workspace, usually in a corporate setting, consisting of a desk, chair, and computer?</p> <p>A) Breakroom B) Cubicle C) Conference Room D) Archive</p>	1
6	#quiz / Time management	Vocabulary quiz on Time management	2	<p>Question 1: Which technique involves working in focused, 25-minute intervals separated by short breaks, to maximize productivity?</p> <p>A) Eisenhower Matrix</p>	1.5

				<p>B) Pareto Principle C) Pomodoro Technique D) Time Blocking</p>	
7	#etymology / Book	Etymology of term book	1	<p>The word ""book"" comes from the Old English word <i>bōc</i> (plural <i>bēc</i>), which referred to a volume or a written document. The origin of <i>bōc</i> itself is debated, but a widely accepted theory connects it to the Proto-Germanic root <i>*bōks</i> and ultimately to the Proto-Indo-European root <i>*bhāg-</i>, meaning 'beech tree'.</p> <p>This connection is based on the practice of carving runes onto wooden tablets, specifically from beech wood. Therefore, a ""book"" may have originally meant 'beech tablet' or 'writing tablet'.</p> <p>Related words across languages include German <i>Buch</i> and Dutch <i>boek</i>, both meaning 'book', and Swedish <i>bok</i>.</p>	1
8	#etymology / Подяка	Etymology of Ukrainian term <i>подяка</i>	2	<p>The Ukrainian word ""подяка"" (<i>podyaka</i>), meaning 'thank you' or 'gratitude', has roots in Proto-Slavic.</p> <p>It comes from the Proto-Slavic root <i>*podjъka</i>, which is related to the verb <i>*podojiti</i> meaning 'to give'. The prefix <i>po-</i> often indicates completion or direction. So, the original meaning was likely 'a gift given' or 'a return of favor'.</p>	2

				Over time, this evolved to mean 'thankfulness' or 'gratitude' for a favor received. The word is similar to other Slavic terms for gratitude, such as Polish podziękować ('to thank').	
9	#etymology / Podiaka	Etymology of Ukrainian term Podiaka	3	<p>The word "podiaka" is a transcription of the Ukrainian word "подяка" (podyaka), meaning 'thank you' or 'gratitude'. Its etymology is as follows:</p> <p>It comes from the Proto-Slavic root *podjьka, which is related to the verb *podojiti meaning 'to give'. The prefix po- often indicates completion or direction. So, the original meaning was likely 'a gift given' or 'a return of favor'.</p> <p>Over time, this evolved to mean 'thankfulness' or 'gratitude' for a favor received. The word is similar to other Slavic terms for gratitude, such as Polish podziękować ('to thank').</p>	3
Загальна кількість максимальних балів:			18	Загальна кількість отриманих балів:	17

Додаток Г. Список тест-кейсів збереження контексту

№	Вхідні дані	Очікуваний результат	Максимальний бал	Фактичний результат	Результат
1	Please compare these 2 terms: notebook, workbook / Can you explain the last word more?	The term workbook is explained	2	Certainly! The last word in the previous message was "topic."	1
2	What are the most common teaching methodologies? / Compare last two terms discussed	The last two terms mentioned in the list (Project-Based Learning and Differentiated Instruction) are compared	1	<p>Based on our previous conversation, the last two terms discussed were Project-Based Learning (PBL) and Differentiated Instruction. Here's a comparison:</p> <p>Project-Based Learning (PBL)</p> <p>What it is: A specific teaching methodology where students gain knowledge and skills by working for an extended period to investigate and respond to a complex question, problem, or challenge. The core focus is on the project itself as the primary vehicle for learning.</p> <p>Key Characteristic: Students apply learned material in a practical, often real-world, context, frequently integrating</p>	1

				<p>multiple subjects to create a final product or presentation.</p> <p>Differentiated Instruction</p> <p>What it is: Not a single methodology but an overall framework for adapting teaching to meet the individual needs of diverse learners in a classroom. The focus is on adjusting the content, process, and products for students based on their readiness, interests, and learning profile.</p> <p>Key Characteristic: The teacher customizes instruction for individual students or small groups, which can be applied <i>within</i> other methodologies (like direct instruction, collaborative work, or even PBL).</p> <p>Comparison:</p> <p>PBL defines <i>what</i> students are doing (a project) to learn a topic.</p> <p>Differentiated Instruction describes <i>how</i> the teacher adjusts instruction for individual students to ensure they can succeed, potentially even within a PBL framework.</p> <p>In short, PBL is a structured activity, while Differentiated Instruction is an approach to personalized learning.</p>	
--	--	--	--	--	--

3	Please translate the word паляниця into English / Find etymology of this word	Etymology of term паляниця	3	<p>The word паляниця (palyanytsya) in Ukrainian has a fascinating etymology:</p> <p>Origin:</p> <p>The term likely comes from the Proto-Slavic word *palъ, meaning "fire," "heat," or "scorched." This root is related to words for "to burn" or "to scorch" in various Slavic languages (e.g., Ukrainian палити (palyty) - "to burn").</p> <p>Meaning Evolution:</p> <ol style="list-style-type: none"> Original Meaning: The initial meaning of паляниця was probably "something baked on fire" or "something baked in a heated oven." The word emphasizes the baking process, particularly the scorching or heating required to cook the bread. Bread Type: Over time, the word became specifically associated with a particular shape and style of bread common in Ukraine—a round loaf. The name highlights its preparation method and appearance (often with a scorched or browned crust). <p>Other Theories (Less Likely):</p> <p>Some folk etymologies</p>	3
---	--	--------------------------------------	---	---	---

				<p>suggest a connection to **палля** (palya), meaning "stake" or "pile," referencing the bread's round shape, but this connection is less supported by linguistic analysis.</p> <p>**Modern Significance:**</p> <p>In modern times, **паляниця** has gained cultural significance beyond just being bread. It has become a linguistic **shibboleth**—a word used to test if someone is a native Ukrainian speaker. The specific combination of sounds (particularly the soft "я" and stressed "и") is difficult for non-native speakers, especially Russian speakers, to pronounce correctly.</p>	
4	<p>Please list the drawbacks of collaborative learning method / What is the difference between applied linguistics and linguistic applied? / Please find definitions of the teaching method I mentioned</p>	<p>List of definitions for term Collaborative Learning</p>	3	<p>Based on our previous conversation, the teaching method you mentioned is **Collaborative Learning**. Here is a definition of this method and related concepts from our discussion:</p> <p>**1. Collaborative Learning Method**</p> <p>* **Definition:** Collaborative learning is an educational approach where students work together in small groups or teams to achieve a common learning goal. It emphasizes mutual interaction, shared</p>	2

	in the previous two messages			<p>responsibility, and peer support rather than individual competition.</p> <p>* Key Principles:</p> <p>Students are encouraged to share knowledge, discuss different perspectives, challenge each other's ideas constructively, and support each other's learning journey. The goal is not just to complete a task, but to deepen understanding through social interaction and shared inquiry.</p> <p>* Examples: Group projects, joint problem-solving activities, peer tutoring, and think-pair-share exercises.</p> <p>2. Applied Linguistics</p> <p>* Definition: While not a teaching <i>method</i> itself, Applied Linguistics is the interdisciplinary field that informs many teaching methods. It uses linguistic theories, data, and insights to solve practical problems where language plays a central role.</p> <p>* Relevance to Teaching: It provides the foundation for methods like communicative language teaching and task-based learning by helping educators understand language acquisition processes, learner differences, and effective pedagogical strategies.</p>	
5	Summarise our conversatio	Summary of the conversation	2	Based on our recent conversation, we have discussed the following	1

	n	with the following topics mentioned: Applied linguistics, Collaborative learning		<p>topics:</p> <ol style="list-style-type: none"> **Drawbacks of Collaborative Learning:** We started by exploring the potential challenges of collaborative learning, including issues like uneven participation, social loafing, time consumption for coordination, and potential conflicts among group members. **Applied Linguistics vs. Linguistics Applied:** We then clarified the difference between these two terms, explaining that "Applied Linguistics" is the broader, interdisciplinary field of study, while "linguistics applied" refers more specifically to the direct application of existing linguistic theories to real-world problems. **Definition of Collaborative Learning:** We concluded by reviewing the definition of collaborative learning, identifying it as an educational approach where students work together in groups to achieve a shared learning goal. 	
6	Compare three last terms we discussed	Comparison of the last three terms mentioned in the conversation (Collaborative Learning Drawbacks, Applied	3	Based on our recent conversation, here is a comparison of the three terms we discussed: Collaborative Learning Drawbacks, Applied Linguistics, and Linguistics Applied. ---	3

		Linguistics, and Linguistics Applied)		<p>### Comparison of Concepts</p> <p>1. Collaborative Learning Drawbacks</p> <p>What it is: These are the potential challenges or limitations associated with the collaborative learning method.</p> <p>Key Issues Identified: Uneven participation, social loafing (reduced individual effort in groups), personality conflicts, and time consumption for group coordination.</p> <p>Relevance: It provides a critical perspective on group work, highlighting issues that educators and students should be aware of when implementing this teaching strategy.</p> <p>2. Applied Linguistics</p> <p>What it is: An interdisciplinary field of study that focuses on solving practical, real-world problems related to language.</p> <p>Scope: Broad and problem-driven. It uses insights from linguistics to address issues in areas like language teaching, translation, speech therapy, and forensic analysis.</p> <p>Key Difference: Unlike "linguistics applied," it doesn't just</p>	
--	--	--	--	--	--

				<p>apply existing theories; it develops its own methodologies and theoretical frameworks specifically for practical application.</p> <p>**3. Linguistics Applied**</p> <p>* **What it is:** The direct action of using established theories and principles from core linguistics (e.g., phonology, syntax) to solve a specific problem.</p> <p>* **Scope:** Narrower and theory-driven. It's a one-way application of knowledge from theory to practice.</p> <p>* **Key Difference:** It is often seen as a less developed approach compared to Applied Linguistics, which involves generating new knowledge within the practical field itself, not just pulling from core linguistics.</p>	
7	What is fine-tuning ? / Can you find a synonym for that term?	Synonym of term fine-tuning	1	<p>Based on the context of fine-tuning in machine learning, here are a few synonyms and related terms:</p> <p>* **Customization:** This emphasizes adjusting the model to fit specific user requirements or data.</p> <p>* **Adaptation:** This highlights the process of adjusting the model from a general purpose to a specific task or domain.</p> <p>* **Specialization:** This focuses on making the</p>	1

				<p>model an expert in a particular niche area after its general training.</p> <p>A related, broader term often used interchangeably in some contexts, is Transfer Learning. Fine-tuning is a specific technique used for transfer learning, where knowledge gained from one task (the large pre-training dataset) is transferred to improve performance on another related task.</p>	
8	What is the topic of our conversation?	A list of topics mentioned in the chat	2	<p>Based on our most recent interactions, the current topic of our conversation is fine-tuning in machine learning.</p> <p>We have discussed:</p> <ul style="list-style-type: none"> * The definition of fine-tuning. * Synonyms for fine-tuning. * A vocabulary quiz about fine-tuning. 	2
9	Can you create a quiz on that topic?	Quiz on the last topic mentioned in the chat (fine-tuning)	3	<p>Here is your quiz on the topic of fine-tuning in machine learning. Please choose the best answer for each question.</p> <p>### Question 1</p> <p>Fine-tuning involves taking a model trained on a massive, general dataset (like general internet text or ImageNet) and adapting it for specialized use on a much smaller, specific dataset. This strategic shift in focus is primarily a form</p>	3

				<p>of:</p> <ul style="list-style-type: none"> A. Dimensionality Reduction B. Ensemble Learning C. Domain Adaptation D. Catastrophic Forgetting <p>### Question 2</p> <p>When fine-tuning a large model, researchers often choose to freeze the weights of the initial, deep layers and only train the parameters in the final few layers (often the classification or regression head). This specific implementation technique is frequently referred to as:</p> <ul style="list-style-type: none"> A. Weight Initialization B. Hyperparameter Optimization C. Feature Extraction D. Generative Adversarial Training <p>### Question 3</p> <p>Fine-tuning is considered the dominant and most successful practical implementation of which broader machine learning paradigm, where knowledge acquired during one task is reused to solve a different, but related, problem?</p> <ul style="list-style-type: none"> A. Reinforcement Learning (RL) B. Transfer Learning C. Unsupervised Clustering D. Active Learning 	
--	--	--	--	---	--

				<p>### Question 4</p> <p>When adapting a large language model (LLM) using new, instruction-based data to align its output with human preferences and safety guidelines, this specific, post-pre-training step is often achieved through fine-tuning, but is distinctly characterized as:</p> <p>A. Zero-Shot Inference B. Prompt Engineering C. Alignment Training D. Weight Pruning</p> <p>### Question 5</p> <p>When a model is fine-tuned, its weights are updated, continuing the learning process. The antithesis of fine-tuning—the alternative approach where the model must learn its task-specific features without leveraging any prior general knowledge—is formally known as:</p> <p>A. Batch Normalization B. Low-Rank Adaptation (LoRA) C. Training from Scratch D. Early Stopping</p> <p>Please provide your answers (A, B, C, or D) for each question.</p>	
			20	Загальна кількість отриманих балів:	17
Загальна кількість максимальних балів:					

Додаток Д. Список тест-кейсів відповідності правилам

№	Вхідні дані	Очікуваний результат	Максимальний бал	Фактичний результат	Результат
1	/start	Welcome message	3	 Welcome! I'm your LinguaBuddy, here to help with: - Study tips and techniques - Linguistics and language learning - Etymology and word origins - Vocabulary quizzes and practice Commands: /help - Show help message /quiz - Start a vocabulary quiz /topic - Change quiz topic /etymology - Analyze word origins /quit - Exit quiz mode /end - End conversation How can I help you today?	3
2	/help	List of commands and instructions	3	 LINGUABUDDY HELP  AVAILABLE COMMANDS: /start - Welcome message and introduction /help - Show this help message /quiz - Enter quiz mode for vocabulary practice /topic - Change quiz topic (only works in quiz mode) /etymology - Analyze word origins /quit - Exit quiz mode /end - End conversation	3

				<p>◆ WHAT I CAN DO:</p> <ul style="list-style-type: none"> - Provide study tips and learning strategies - Help with time management and organization - Answer linguistics questions - Explain language learning concepts - Create vocabulary quizzes - Analyze word origins and etymology - Offer personalized study advice - Discuss grammar, phonetics, syntax, and semantics - Recommend study techniques (spaced repetition, active recall, etc.) <p>◆ HOW TO USE ME:</p> <ul style="list-style-type: none"> - Use commands above for specific functions - Ask me any question about studying or linguistics - Request a quiz with /quiz - Ask about word origins with /etymology - Chat naturally - I remember our conversation context <p>Example questions: 'How can I improve my memory retention?' 'What is phonology?' 'Can you explain the Pomodoro technique?' 'How do I learn vocabulary effectively?'</p> <p>Need something? Just ask! </p>	
--	--	--	--	--	--

3	/quiz	Enter quiz mode	3	Great! What topic would you like your vocabulary quiz to be about?	3
4	/topic	Change quiz topic	3	Please specify the new topic you want to change to.	1
5	/etymology	Word origin analysis	3	To analyze a word's etymology, please tell me the word you'd like to look up!	1
6	/quit	Exit quiz mode	3	-	2
7	/end	End conversation	3	Thanks for studying with me! Good luck with your learning! 🙌	3
8	Harmful content	System refuses to provide the answer on given topic and warns user	3	-	0
Загальна кількість максимальних балів:			24	Загальна кількість отриманих балів:	15

Додаток Е. Список тест-кейсів складності

№	Вхідні дані	Очікуваний результат	Максимальний бал	Фактичний результат	Результат
1	Please find etymology of word etymology	Word origin analysis of the term 'etymology'	2	 Etymology of ETYMOLOGY <ul style="list-style-type: none"> ◆ Origin: Ancient Greek, from the compound noun **ἐτυμολογία** (*etymología*). ◆ Root Meaning: The word is a fusion of two Greek components: <ol style="list-style-type: none"> 1. **ἔτυμον** (*étymon*): meaning "the true, real, or original sense (of a word)." 2. **-λογία** (*-logía*): meaning "study of" or "discourse," derived from **λόγος** (*lógos*, meaning "word, reason, or account"). <p>The literal meaning is: **"The study of the true sense of a word."**</p> ◆ Historical Development: The path of the word demonstrates its classical scholarly roots: <p>* **Ancient Greek (c. 4th Century BCE):** **ἐτυμολογία** (*etymología*). Used by philosophers and grammarians, notably Plato, who explored the</p> 	2

				<p>"truth" inherent in words.</p> <p>* Latin (Post-Classical Era): etymologia. Adopted directly by Roman scholars. The term was cemented in Western tradition by Isidore of Seville's monumental 7th-century encyclopedia, <i>Etymologiae</i>.</p> <p>* Old/Middle French (13th-15th Centuries): ethimologie. The term passed into French vernacular, undergoing slight spelling shifts.</p> <p>* Middle English (Late 14th Century): ethimologie or etymology. It entered English, likely via French, during a period of high classical learning.</p> <p>* Modern English (16th Century onwards): Etymology. The spelling reverted to the classicizing Latin form, standardizing the term for the academic discipline.</p> <p>◆ Meaning Evolution: Initially, etymology was a more philosophical or rhetorical exercise, often based on speculation about a word's "true" (sometimes allegorical) nature.</p> <p>* Shift: By the 17th century, influenced by the rise of empirical science and historical linguistics, the meaning solidified into its modern definition: the historical, objective tracing of a word's origins,</p>	
--	--	--	--	---	--

				<p>transmission, and development** through documented evidence, rather than mere philosophical deduction.</p> <p>◆ Related Words: Words sharing the root components:</p> <p>* Etymon: The original, true root word from which a later word is derived.</p> <p>* Logic: (from *lógos*) The study of reason.</p> <p>* Geology, Biology, Theology: (from *-logía*) Any specialized branch of study or discourse.</p> <p>* True: (Though not a direct cognate, it shares the semantic core of *étymos* in Proto-Indo-European concepts of truth).</p> <p>◆ Interesting Facts: The word *etymology* is self-referential: its own etymology provides a perfect demonstration of the discipline it names, showing the historical layers and</p>	
2	What teaching method should I choose for the class of 4 people?	A list of teaching methods	1	A class of four people offers a fantastic opportunity for highly personalized and interactive learning . Here are some effective teaching methods tailored specifically for this small group size:	1

				<p>### 1. Collaborative Learning (High Interaction)</p> <p>This is perhaps the most effective method for a group of four. In a large class, collaborative learning can sometimes be challenging due to social loafing or large group dynamics, but with four people, every student's contribution is essential and visible.</p> <p>* Key Activities:</p> <p>* Socratic Seminars/Discussions: Introduce a central topic or question. Each student must contribute by asking questions, responding to others, and building on previous points.</p> <p>* Role-Playing and Simulations: Assign roles to each student for a scenario (e.g., a debate, a business meeting, or a historical event simulation). This encourages active use of language and critical thinking.</p> <p>* Mini Group Projects: Give the group a short project to complete together during class time. This could involve creating a presentation, solving a complex problem, or analyzing a text from different perspectives.</p> <p>### 2. Student-Centered Instruction</p>	
--	--	--	--	---	--

				<p>With only four students, you can easily shift the focus from a traditional lecture format to one driven by student questions and interests.</p> <ul style="list-style-type: none"> * Key Activities: * Personalized Feedback Loops: Because the class is small, you can provide individualized feedback on every student's work in real-time, focusing on their specific strengths and weaknesses. * Tailored Content: Adjust reading materials, topics, or difficulty levels based on the specific needs and goals of each student. * Student Presentations: Assign each student a topic to research and present to the small group. This builds public speaking skills and encourages deeper understanding of a subject. <p>### 3. Flipped Classroom Model</p> <p>In this model, students review pre-recorded lectures or readings before class. Class time is then entirely dedicated to discussions, activities, and problem-solving.</p> <ul style="list-style-type: none"> * Why it works for 4 students: You avoid spending valuable class time lecturing and 	
--	--	--	--	---	--

				<p>maximize the time for interactive application of the material.</p> <p>### 4. Hybrid Method: Individual Accountability within Group Activities</p> <p>A common challenge in small groups can be uneven participation, even with only four people. To prevent this:</p> <p>* **Assign Individual Roles:** When doing group work, explicitly assign a "facilitator," "note-taker," "reporter," and "timekeeper" to each student. Rotate roles regularly so everyone develops different skills.</p> <p>* **"Think-Pair-Share" Modification:** Have students work individually for a few minutes ("Think"), then pair up with one other student ("Pair"), and finally share their ideas with the other pair ("Share"). This ensures every student processes the information before contributing to the larger discussion.</p> <p>**Recommendation:** I suggest a primary approach that combines collaborative learning and student-centered instruction. This maximizes engagement and ensures that each student receives targeted support while learning from their</p>	
--	--	--	--	--	--

				peers.	
3	Find reliable academic sources to learn philosophy of language	A list of academic works	3	<p>This is an excellent topic. Philosophy of language examines the relationship between language, meaning, truth, and the world. Here are reliable academic sources, categorized by type, to help you learn about it:</p> <p>### 1. Introductory Textbooks and Handbooks</p> <p>These provide a comprehensive overview and are ideal for beginners or those seeking structured learning.</p> <p>* **"Philosophy of Language: A Contemporary Introduction" by William G. Lycan (Routledge):** This is a highly regarded textbook covering a wide range of topics, from meaning and reference to speech acts and pragmatics. It presents complex ideas in a clear and accessible manner, making it suitable for a first course in the subject.</p> <p>* **"A Companion to the Philosophy of Language" edited by Bob Hale and Crispin Wright (Wiley-Blackwell):** This is a collection of essays by leading scholars on key topics in philosophy of language. While more advanced than Lycan's book, it serves as an excellent reference for</p>	3

				<p>specific areas and different perspectives.</p> <p>* **"Philosophy of Language: The Key Thinkers" edited by Barry Lee (Continuum):** If you prefer a historical approach, this book breaks down the central ideas of major philosophers who have shaped the field, such as Frege, Russell, Wittgenstein, and Austin.</p> <p>### 2. Foundational Primary Texts (The Classics)</p> <p>To truly grasp the philosophy of language, you'll need to read the original works of the most influential thinkers. These are often challenging but essential.</p> <p>* **Gottlob Frege - "On Sense and Reference" (1892):** This short paper is arguably the single most important text in the history of analytic philosophy of language. Frege introduces the distinction between "sense" (the way an object is presented) and "reference" (the object itself), which is foundational to understanding meaning.</p> <p>* **Bertrand Russell - "On Denoting" (1905):** Russell's response to Frege's work, providing an alternative theory of how proper names and descriptions work. It's</p>	
--	--	--	--	--	--

				<p>crucial for understanding the debate over reference.</p> <p>* **Ludwig Wittgenstein - "Tractatus Logico-Philosophicus" (1921):** Wittgenstein's early work attempts to define the limits of language and thought by proposing that language is a "picture" of reality. A difficult read, but fundamentally important.</p> <p>* **J.L. Austin - "How to Do Things with Words" (1962):** This book introduced the concept of Speech Act Theory, arguing that words aren't just for stating facts but are actions themselves (e.g., promising, marrying, commanding).</p> <p>### 3. Online Courses and Open Access Resources</p> <p>For structured, free learning, online platforms offer excellent resources from top universities.</p> <p>* **Stanford Encyclopedia of Philosophy (SEP):** This online, peer-reviewed encyclopedia is considered the gold standard for philosophy research. Search for entries like "Philosophy of Language," "Meaning," "Reference," and "Speech Acts." Each entry provides a detailed overview and a comprehensive bibliography.</p>	
--	--	--	--	---	--

				<p>* **MIT OpenCourseware - "Philosophy of Language":** Many universities, including MIT, make their course materials (syllabi, reading lists, and sometimes lecture notes) available online. Searching for "philosophy of language syllabus" can help you build your own reading list.</p> <p>### 4. Key Thinkers and Areas of Study</p> <p>When researching a topic, be sure to look for these central figures and concepts:</p> <p>* **Reference and Meaning:** Frege, Russell, Kripke, Donnellan. * **Speech Acts and Pragmatics:** Austin, Searle, Grice. * **Holism and Quinean Pragmatism:** Quine, Davidson. * **Contextualism and Relevance Theory:** Sperber and Wilson.</p>	
4	Is applied linguistics taught in Volyn National University?	Yes	1	<p>Based on current information from Volyn National University's program offerings, a degree titled specifically **"Applied Linguistics"** may not be available as a separate program.</p> <p>However, the content and principles of applied linguistics are almost certainly taught within the</p>	0.5

				<p>framework of a broader program. You should look into the offerings of the Faculty of Philology.</p> <p>* Program Titles to look for: Look for specializations under "Philology," particularly those focused on:</p> <ul style="list-style-type: none"> * English Language and Literature * Translation Studies (Translators/Interpreters) * Methodology of Teaching Foreign Languages <p>* What to expect: While the degree name might be "Philology," the curriculum often includes courses on language teaching methodology, second language acquisition, and intercultural communication—all core components of applied linguistics.</p> <p>Recommendation: To get the most accurate and up-to-date information, I highly recommend checking the official website of Volyn National University's Faculty of Philology or contacting the faculty directly. Program specializations and exact course names can change over time.</p>	
5	Is it advisable to change teaching	Explanation of why teaching methodologies should be	2	<p>Yes, absolutely.</p> <p>Changing teaching methodology over time is not only advisable, but</p>	2

	<p>methodology throughout time?</p>	<p>changed and combined</p>		<p>essential for effective and engaging teaching.**</p> <p>Here are the key reasons why adapting your methodology is beneficial and often necessary:</p> <p>### 1. Adaptation to Student Needs</p> <p>Every group of students is different. A teaching method that worked perfectly for one cohort might fall flat with the next. Changing methodology allows you to:</p> <p>* **Address Diverse Learning Styles:** Some students learn best visually, others by doing (kinesthetic learning), and some by listening and discussing (auditory/social learning). Varying your methods ensures you reach different students throughout the course.</p> <p>* **Respond to Feedback:** If students are disengaged or failing to grasp concepts with a particular method (e.g., a pure lecture format), adapting to more interactive methods (like group discussions or project-based learning) can significantly improve comprehension.</p> <p>### 2. Matching Methodology to Content</p>	
--	-------------------------------------	-----------------------------	--	---	--

				<p>Different topics require different approaches. A "one-size-fits-all" method rarely works across all subjects or lessons.</p> <p>* Example: When teaching foundational concepts or new vocabulary, direct instruction or a structured lecture might be necessary. However, when teaching critical thinking, problem-solving, or application of knowledge, collaborative activities or case studies are far more effective.</p> <p>### 3. Maintaining Student Engagement</p> <p>Students often suffer from "methodology fatigue" when subjected to the same routine day after day. Varying your approach keeps things fresh and holds students' attention.</p> <p>* Shifting Rhythms: Transitioning from individual work to group work, or from a short lecture to a hands-on activity, can keep energy levels high and prevent monotony.</p> <p>### 4. Continuous Professional Growth</p> <p>As a teacher, you gain new experience, discover new technologies, and learn new techniques throughout</p>	
--	--	--	--	--	--

				<p>your career. Changing methodology allows you to incorporate these new discoveries and improve your craft.</p> <p>* Example: A teacher might initially rely heavily on lectures but later transition to a "flipped classroom" model after seeing its success in a professional development workshop.</p> <p>In summary, a flexible approach to methodology allows you to tailor your teaching to the learners, the subject matter, and new developments in pedagogy, ultimately leading to better learning outcomes for your students.</p>	
6	Can you test me on vocabulary about LLM?	Quiz on the topic of LLM is created	3	<p>### Vocabulary Quiz: Large Language Models (LLMs)</p> <p>Let's test your knowledge of key terminology related to LLMs. Choose the best answer for each question.</p> <p>### Question 1</p> <p>What term refers to the process of adapting a pre-trained language model to a specific task or domain using additional, smaller datasets?</p> <p>A. Prompt Engineering B. Transfer Learning C. Fine-Tuning D. Zero-Shot Learning</p>	2

				<p>### Question 2</p> <p>What is the term for a pre-trained model's ability to perform a task without having seen any specific examples of that task during its training process?</p> <p>A. Few-Shot Learning B. Zero-Shot Learning C. Self-Supervised Learning D. Reinforcement Learning from Human Feedback (RLHF)</p> <p>### Question 3</p> <p>What technique involves designing specific instructions or inputs (like "Act as a helpful assistant") to guide an LLM's behavior and response quality without changing its underlying model weights?</p> <p>A. Fine-Tuning B. Prompt Engineering C. Data Augmentation D. Pre-training</p> <p>### Question 4</p> <p>When a model generates output that seems plausible and coherent but is factually incorrect or nonsensical, what is this phenomenon called?</p> <p>A. Overfitting B. Catastrophic Forgetting C. Hallucination D. Bias Amplification</p>	
--	--	--	--	---	--

				<p>### Question 5</p> <p>What method involves training a separate model to act as a judge or reward model, using human preferences (feedback on generated answers) to guide the LLM's fine-tuning process towards better, safer outputs?</p> <p>A. Zero-Shot Learning B. Retrieval-Augmented Generation (RAG) C. Low-Rank Adaptation (LoRA) D. Reinforcement Learning from Human Feedback (RLHF)</p>	
7	What is linguistics?	Academic explanation of linguistics	1	<p>Linguistics is the **scientific study of language**. It involves analyzing language from various perspectives to understand its structure, function, and development.</p> <p>Think of it as looking under the hood of human communication. Instead of just learning to speak a language, linguistics studies *how* language works.</p> <p>Here are the main areas of study within linguistics:</p> <p>* **Phonetics:** The study of speech sounds themselves—how they are produced, heard, and described.</p> <p>* **Phonology:** The study of how sounds are organized and function in a</p>	1

				<p>particular language's system.</p> <ul style="list-style-type: none"> * Morphology: The study of word structure and how words are formed from smaller units called morphemes (e.g., "un-" + "happy" + "-ness"). * Syntax: The study of sentence structure and the rules governing how words combine to form phrases and sentences. * Semantics: The study of meaning in language, examining how words and sentences convey meaning. * Pragmatics: The study of how context influences meaning and language use, examining what people <i>do</i> with language in communication. <p>Beyond these core components, linguistics also explores areas like:</p> <ul style="list-style-type: none"> * Sociolinguistics: How language varies in different social groups. * Psycholinguistics: How the brain processes and understands language. * Historical Linguistics: How languages change over time. * Applied Linguistics: How linguistic theories can be applied to practical problems like language teaching and speech therapy. 	
--	--	--	--	---	--

8	Create a grammar ex on relative clauses for C1 students	Grammar ex on the specified topic is created	2	<p>This is a great exercise for C1 students, as it moves beyond basic "who/which/that" and focuses on more formal structures and prepositions.</p> <p>Here is an exercise designed to test C1-level understanding of relative clauses:</p> <p>---</p> <p>### C1 Grammar Exercise: Advanced Relative Clauses</p> <p>**Instructions:** Combine the two separate sentences into one sophisticated sentence using a relative clause. Be careful to use the correct relative pronoun (**which, whom, whose, where, when**) and pay close attention to prepositions and punctuation (commas for non-defining clauses).</p> <p>**Example:**</p> <p>*Original Sentences:* We stayed at a hotel last weekend. The hotel was extremely luxurious.</p> <p>*Combined Sentence:* We stayed at a hotel last weekend, **which** was extremely luxurious.</p> <p>---</p> <p>#### Exercise Questions</p> <p>1. **Original Sentences:** Our team is working on a complex project. The</p>	2
---	---	--	---	--	---

				<p>completion of this project depends on several external factors.</p> <p>* *Combined Sentence:* We are working on a complex project, ** _** depends on several external factors.</p> <p>2. **Original Sentences:** The company organized a meeting for all employees. Many employees were critical of the new policies discussed at the meeting.</p> <p>* *Combined Sentence:* The company organized a meeting for all employees, ** _** were critical of the new policies.</p> <p>3. **Original Sentences:** My best friend is a talented software engineer. I have enormous respect for her professional achievements.</p> <p>* *Combined Sentence:* My best friend, ** _** I have enormous respect, is a talented software engineer.</p> <p>4. **Original Sentences:** We visited a small village in the mountains. We could see the entire valley from this village.</p> <p>* *Combined Sentence:* We visited a small village in the mountains, ** _** we could see the entire valley.</p> <p>5. **Original Sentences:** The professor, Dr. Evelyn Reed, gave a lecture on historical linguistics. Her</p>	
--	--	--	--	---	--

				<p>groundbreaking research was recently published.</p> <p>* *Combined</p> <p>Sentence:* The professor, Dr. Evelyn Reed, ** ** research was recently published, gave a lecture on historical linguistics.</p> <p>#### Answer Key and Explanation</p> <p>1. **Combined</p> <p>Sentence:** We are working on a complex project, **the completion of which** depends on several external factors.</p> <p>* **Explanation:**</p> <p>This advanced structure uses "of which" to create a possessive relationship (similar to "whose") for inanimate objects in a non-defining clause.</p> <p>2. **Combined</p> <p>Sentence:** The company organized a meeting for all employees, **many of whom** were critical of the new policies.</p> <p>* **Explanation:**</p> <p>This uses a quantifier ("many") combined with a relative pronoun ("whom") to refer to a specific number or portion of the group mentioned.</p> <p>3. **Combined</p> <p>Sentence:** My best friend, **for whom** I have enormous respect, is a talented software engineer.</p> <p>* **Explanation:**</p>	
--	--	--	--	---	--

				<p>The preposition "for" is moved to the beginning of the relative clause, creating a more formal style typically expected at C1 level.</p> <p>4. **Combined Sentence:** We visited a small village in the mountains, **from where** we could see the entire valley. * **Explanation:** "From where" (or "from which") specifies the location and the movement away from it, providing more precise information than simply "where."</p> <p>5. **Combined Sentence:** The professor, Dr. Evelyn Reed, **whose** research was recently published, gave a lecture on historical linguistics. * **Explanation:** Using "whose" to show possession (Dr. Reed's research) in a non-defining clause to describe the person.</p>	
9	What Noam Chomsky said about current approach to building AI?	Scientist's critic regarding statistical approach	3	Noam Chomsky has been a vocal and consistent critic of the current approach to building AI, particularly large language models (LLMs) like those developed by OpenAI and Google. His core arguments stem from his long-standing theories in linguistics, which emphasize an innate, structured, and rules-based	3

				<p>nature of human language.</p> <p>Here is a summary of Noam Chomsky's main criticisms of modern AI:</p> <p>### 1. The Distinction Between Human Intelligence and Statistical Imitation</p> <p>Chomsky's most fundamental critique is that current AI models are fundamentally different from human intelligence. He argues that LLMs are essentially sophisticated statistical parrots.</p> <p>* **Human Language (Generative Grammar):** Chomsky's theory posits that humans possess an innate "Universal Grammar"—a biological, hardwired structure in the brain that allows children to learn language rapidly based on limited input. This system is **generative**, meaning it allows us to create and understand an infinite number of novel sentences based on a finite set of rules.</p> <p>* **AI (Statistical Modeling):** Chomsky argues that LLMs operate purely on statistical probabilities. They predict the next word in a sequence based on vast quantities of data. They do not possess true understanding, intent, or a generative grammar.</p>	
--	--	--	--	--	--

				<p>### 2. Lack of Understanding and Reasoning</p> <p>Chomsky maintains that LLMs, despite their ability to generate coherent text, lack genuine understanding and reasoning.</p> <p>* Plausibility over Truth: He claims these models prioritize plausibility (generating text that sounds correct) over truth or coherence. This is why they frequently produce "hallucinations"—confidently stating facts that are completely wrong.</p> <p>* The Problem of "Theory of Mind": He argues that LLMs do not have a "theory of mind" or the capacity to understand the intentions, beliefs, or desires of others, which is critical for real communication.</p> <p>### 3. The "Search Engine" Fallacy</p> <p>Chomsky compares LLMs to highly advanced search engines, but emphasizes that they are fundamentally different from human cognition.</p> <p>* Knowledge vs. Information Access: While LLMs can access</p>	
--	--	--	--	---	--

				<p>and process information rapidly, Chomsky differentiates this from true knowledge and comprehension, which involves applying rules, forming hypotheses, and engaging in critical thought.</p> <p>### 4. The Risk of Misdirection in Scientific Research</p> <p>From a scientific perspective, Chomsky worries that focusing on large data sets and statistical models misdirects research away from true advances in understanding human cognition.</p> <p>* **Avoiding the "Hard Problem":** He suggests that current AI methods bypass the more difficult question of how human intelligence actually works. He calls this a scientific dead-end for understanding human nature.</p> <p>### Summary of Chomsky's View:</p> <p>In a widely cited 2023 op-ed for <i>The New York Times</i>, Chomsky and his co-authors stated:</p> <p>> **"The human mind is not a statistical machine. The mind is a system of specific, high-level capacities that operate on</p>	
--	--	--	--	--	--

				<p>symbolic representations and combine them in complex ways."**</p> <p>While he acknowledges the technological novelty of LLMs, he strongly believes they are not a path to creating truly human-like intelligence or language understanding.</p>	
Загальна кількість максимальних балів:			18	Загальна кількість отриманих балів:	16.5