

**ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ**  
**Навчально-науковий інститут економіки, управління, права та**  
**інформаційних технологій**  
**Кафедра інформаційних систем та технологій**

# **КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття ступеня вищої освіти магістр

на тему: **«Методика порівняльного аналізу великих мовних моделей  
з опису зображень»**

Виконав: здобувач вищої освіти  
за освітньою програмою  
Інформаційні управляючі системи та  
технології  
спеціальності 126 Інформаційні  
системи та технології  
ступеня вищої освіти магістр  
групи 126ІСТ\_мд\_2023  
Кибкало Володимир Володимирович  
Керівник: Слюсарь Ігор Іванович  
Рецензент: Муравльов Володимир  
Вячеславович

**Полтава – 2024 року**

## ВСТУП

Актуальність теми кваліфікаційної роботи підтверджується необхідністю автоматизації вирішення завдання опису зображень за рахунок використання великих мовних моделей на основі архітектури трансформерів. Це є важливою складовою інтелектуальних систем з розпізнавання об'єктів, аналізу візуального контенту та створення доступного середовища для людей із порушеннями зору та ін. Наразі існує багато моделей, які демонструють високу ефективність у цьому завданні. Однак питання проведення порівняльного аналізу мовних моделей з опису зображень на основі прозорої оцінки їх продуктивності є недостатньо дослідженим в теоретичному плані. Все це свідчить про актуальність теми роботи.

*Зв'язок роботи з науковими програмами, темами.* Робота відповідає дослідженням в межах науково-дослідної ініціативної тематики «Організаційно-методологічні аспекти впровадження інформаційно-комунікаційних систем і технологій в управлінні діяльністю сучасних організацій та підприємств за умов переходу до цифрової економіки» (ДРН 0123U105060, 2023-2028 рр.), що реалізується на кафедрі інформаційних систем та технологій, тематиці досліджень навчально-дослідної лабораторії інтелектуальних систем, комп'ютерних мереж та інтернет речей кафедри інформаційних систем та технологій Полтавського державного аграрного університету.

*Метою* кваліфікаційної роботи є підвищення ефективності текстових описів зображень за допомогою великих мовних моделей на основі розробленої методики їх порівняльного аналізу.

*Завданнями* кваліфікаційної роботи є:

- аналіз стратегій інтеграції комп'ютерного зору з мовними моделями;
- розробка варіантів оцінки великих мовних моделей з опису зображень;
- обґрунтування підходів щодо формування запитів у роботі з

мовними моделями;

– розробка рекомендацій щодо використання методики порівняльного аналізу великих мовних моделей з опису зображень.

*Об'єктом дослідження* є процес генерування текстових описів зображень за допомогою великих мовних моделей.

*Предметом дослідження* є великі мовні моделі з обробки зображень.

*Методами* дослідження є аналітичний, інформаційно-пошуковий, методи оптимізації мовних моделей, робота з нейронними мережами Vision Transformer.

*Інформаційна база* кваліфікаційної роботи базується на ресурсах про великі мовні моделі, візуальні асистенти до мовних моделей, інструментарій для їх порівняльного аналізу.

*Елементи наукової новизни* роботи полягають в розробці методики порівняльного аналізу великих мовних моделей з опису зображень.

*Практична значущість* роботи полягає в обґрунтуванні рекомендацій щодо використання великих мовних моделей для опису зображень – можуть бути використані для подальших досліджень за даною тематикою та при проектуванні хмарних сервісів.

*Апробація результатів* відбувалася в рамках VII Міжнародної студентської наукової конференції «Наука сьогодення: від досліджень до стратегічних рішень» (листопад 2024 р., м. Кривий Ріг) та VII Міжнародної студентської наукової конференції «Розвиток суспільства та науки в умовах цифрової трансформації» (листопад 2024 р., м. Тернопіль).

*Структура кваліфікаційної роботи* логічно пов'язана з завданнями досліджень і містить вступ, три розділи основної частини, висновки, список використаних джерел, додатки. Загальний обсяг пояснювальної записки кваліфікаційної роботи складає 68 сторінок формату А4. Вона містить 25 рисунків.

# РОЗДІЛ 1

## АНАЛІЗ ОСОБЛИВОСТЕЙ РЕАЛІЗАЦІЇ ВІЗУАЛЬНИХ АСИСТЕНТІВ МОВНИХ МОДЕЛЕЙ

### 1.1 Загальні відомості про візуальні асистенти мовних моделей

Великі мовні моделі (Large Language Models, LLM) [1] вже стали частиною сучасного світу. Їх застосовують, щоб спростити сучасну рутину, а також використовуємо для вирішення бізнес-завдань. У міру розвитку застосування штучного інтелекту зростає попит на моделі, здатні розуміти та виробляти як текстову, так і візуальну інформацію. Ця тенденція призвела до появи мультимодального штучного інтелекту (Artificial Intelligence, AI), який інтегрує обробку природної мови (Natural Language Processing, NLP) [2] з функціями комп'ютерного зору (CV) [3]. Це злиття покращує традиційні завдання CV та відкриває шляхи для інноваційних додатків у різних сферах. Моделі можуть надавати більш детальні та релевантні відповіді. Інтеграція візуальної інформації дозволяє моделі краще розуміти ситуацію або запит. Відкриваються можливості в галузях, де текстової інформації недостатньо. При цьому стали окремо виділяти розширення або компонент LLM, який має назву візуального асистенту. Він дозволяє їй приймати візуальні дані як вхідні, розпізнавати об'єкти, сцени, текст на зображеннях тощо з метою аналізу цих даних та їх інтерпретації, а також описувати зображення, відповідати на питання щодо візуального контенту, надавати пояснення. В якості прикладу візуальних асистентів для LLM можна вказати моделі типу Visual Question Answering (VQA) [4] – системи, які відповідають на питання про вміст зображення; Contrastive Language-Image Pre-training (CLIP) [5] – модель від OpenAI [6], яка навчається на парах «зображення-текст» для встановлення зв'язку між ними, а також деякі версії GPT-4 [7] можуть приймати зображення як вхідні дані та генерувати відповіді на основі їх аналізу. По суті, в даному випадку, утворюється візуальна мовна модель або

ще використовується синонім модель візуальної мови (Vision-Language Model, VLM). Спрощено, роботу візуального асистенту для LLM можна представити наступним чином. По-перше, здійснюється обробка зображення, для якої використовується нейронна мережа, що перетворює зображення у векторне представлення. Такий енкодер зображень реалізується на основі архітектури згорткових мереж (Convolutional Neural Network, CNN) або Vision Transformer. Зображення аналізується для виявлення ключових особливостей (екстракція ознак), таких як об'єкти, їхні атрибути та взаємодія між ними. По-друге, повинна бути реалізована інтеграція з мовною моделлю. Векторне представлення зображення поєднується з мовною моделлю, що дозволяє їй «розуміти» візуальну інформацію. Тобто виконується об'єднання модальностей. При цьому мультимодальні трансформери здатні обробляти як текстові, так і візуальні дані одночасно. Нарешті, генерується відповідь. Модель враховує як візуальний контекст, так і текстовий запит користувача. На основі інтегрованої інформації генерується текстова відповідь, яка може описувати зображення, відповідати на питання або надавати іншу релевантну інформацію. Одним з яскравих прикладів мультимодальної LLM є Generative Pre-trained Transformer 4 with Vision (GPT-4V). Ця версія GPT-4 здатна працювати як із текстовими, так і з візуальними даними. GPT-4V збудовано на основі трансформерів із можливістю інтеграції тексту й візуальних даних. Модель навчалася на великому обсязі текстових і візуальних даних, що дозволяє їй розуміти як загальні концепції, так і вузькоспеціалізовані теми. Незважаючи на свої потужності, модель може помилятися в складних або нетипових сценаріях, а також мати обмеження в розпізнаванні дуже специфічного візуального контенту. Вона є продовженням розвитку попередніх моделей OpenAI і надає можливість аналізувати зображення разом із текстом, що відкриває нові напрями для використання AI. Модель здатна аналізувати візуальний контент (зображення, фотографії, графіки тощо), може відповідати на запитання, пов'язані із зображенням, описувати його вміст, визначати деталі чи контекст, а також генерувати пояснення до

побаченого. Поєднання текстового й візуального аналізу дозволяє працювати із завданнями, які включають інтеракцію між цими двома типами даних. Наприклад, можна подати на вхід текстовий запит разом із зображенням для детального аналізу чи створення опису. GPT-4V може враховувати контекст на основі тексту та візуальних елементів, що робить її більш ефективною в складних задачах, таких як діагностика медичних зображень або аналіз наукових ілюстрацій. Модель підтримує інтерактивну взаємодію, де користувач може уточнювати свої запити чи додавати нові дані до попередніх. Всі ці можливості можна використовувати для аналізу рентгенівських знімків або медичних зображень у поєднанні з текстовою інформацією про стан пацієнта; створення інтерактивних навчальних матеріалів, які включають текст, зображення та навіть схеми; генерації описів або ідей на основі візуальних даних для рекламних кампаній або проєктів у сфері мистецтва; аналізу візуальних доказів у поєднанні з текстовою документацією, а також для технічної підтримки в якості інструментів, які можуть аналізувати фото обладнання, щоб надавати рекомендації з налаштувань або ремонту.

## **1.2 Аналіз ключових стратегій інтеграції комп'ютерного зору з мовними моделями**

Згідно [8], інтеграція LLM з CV поєднує їхні сильні сторони для створення синергетичного мультимодального AI для глибшого розуміння візуальних даних. Інтегруючи ці можливості в VLM, мультимодальний AI може виконувати завдання, що виходять за рамки простого маркування або ідентифікації. Вони можуть генерувати описові текстові інтерпретації візуальних сцен, надаючи контекстуально релевантні ідеї, які імітують людське розуміння. Вони також можуть генерувати точні підписи, анотації або навіть відповідати на запитання, пов'язані з візуальними даними.

Наприклад, VLM може проаналізувати фотографію міської вулиці та згенерувати підпис, який не лише ідентифікує місце події («жвава вулиця міста в годину пік»), але й надає контекст («пішоходи, що поспішають тротуарами, вздовж яких розташовані магазини та кафе»). Він може анотувати зображення мітками для ключових елементів, таких як «пішохідний перехід», «світлофори» та «автобусна зупинка», а також відповідати на запитання про сцену, наприклад, «Який зараз час доби?».

VLM потребують великих наборів даних пар «зображення-текст» для навчання. Навчання мультимодальному представленню передбачає навчання моделей розуміти та представляти інформацію як з тексту (мова), так і з візуальних даних (зображення, відео). Попереднє навчання LLM на великомасштабному тексті з подальшим їх тонким налаштуванням на мультимодальних наборах даних значно покращує їхню здатність розуміти та генерувати текстові описи візуального контенту.

Для попередньо навчені LLM (VLPM) використовуються масивні текстові набори даних, адаптуються до візуальних завдань за допомогою додаткового навчання на розмічених візуальних даних, продемонстрували значний успіх. Цей метод використовує вже існуючі лінгвістичні знання, закодовані в LLM, для підвищення продуктивності при виконанні завдань CV з відносно невеликими обсягами анотованих даних.

Контрастне навчання [9] попередньо тренує VLM, використовуючи великі набори даних пар «зображення-підпис» для спільного навчання окремих кодерів зображення та тексту. Ці кодери відображають зображення та текст у спільному просторі функцій, мінімізуючи відстань між парами, що збігаються, і максимізуючи її між парами, що не збігаються, допомагаючи VLM вивчати подібності та відмінності між точками даних.

CLIP (рис. 1.1) – популярна система VLM, використовує контрастне навчання для досягнення можливостей прогнозування нульового знімка. Спочатку він попередньо тренує кодери тексту та зображень на парах зображення-текст. Під час прогнозування нульового кадру CLIP порівнює

невидимі дані (зображення або текст) із вивченими представленнями та оцінює найбільш релевантний підпис або зображення на основі їх найближчого збігу в просторі функцій.

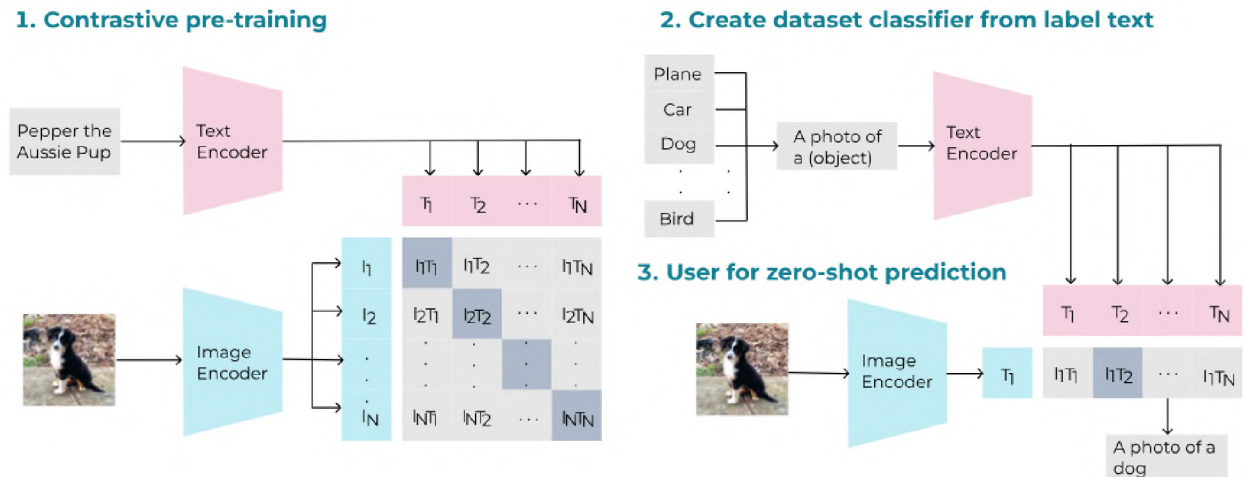


Рисунок 1.1 – Архітектура CLIP

CLIP, незважаючи на свою вражаючу продуктивність, має обмеження, такі як відсутність можливості інтерпретації, що ускладнює розуміння процесу прийняття рішень. Він також бореться з дрібнозернистими деталями, відносинами та нюансами емоцій та може зробити упередження з даних перед навчанням, викликаючи етичні проблеми в системах прийняття рішень.

Багато базових моделей CV залишаються обмеженими заздалегідь визначеними завданнями, не маючи відкритих можливостей LLM. Візуальні LLM (VisionLLM) [10] вирішує цю проблему, розглядаючи зображення як іноземну мову, узгоджуючи завдання зору з гнучкими мовними інструкціями (рис. 1.2). Потім декодер на основі LLM робить прогнози для відкритих завдань на основі цих інструкцій. Ця інтеграція дозволяє краще налаштовувати завдання та глибше розуміти візуальні дані, потенційно долаючи проблеми CLIP завдяки детальним деталям, складним зв'язкам та можливості інтерпретації. VisionLLM може налаштовувати завдання за допомогою мовних інструкцій, від дрібнозернистого рівня об'єкта до грубого рівня завдань. Він досягає понад 60 % середньої точності (mAP) на наборі

даних COCO [11], маючи на меті встановити новий стандарт для моделей загального профілю, що інтегрують зір і мову. Розглянемо приклад, що відповідає рис. 1.3. Людина задає prompt: «З предметами всередині, якщо тобі стане жарко, що ти зробиш?». VisionLLM надає таку відповідь: «По-перше, Ви можете спробувати вимкнути комп'ютер та інші електроприлади в кімнаті, щоб зменшити тепло, що виділяється під час їхньої роботи. В якості альтернативи можна використовувати наданий на зображенні настільний вентилятор для створення ефекту охолодження в приміщенні. Ви можете підключити вентилятор до мережі та налаштувати, щоб переконалися в тому, що видуває повітря в напрямку, який ефективно досягає людей у кімнаті».

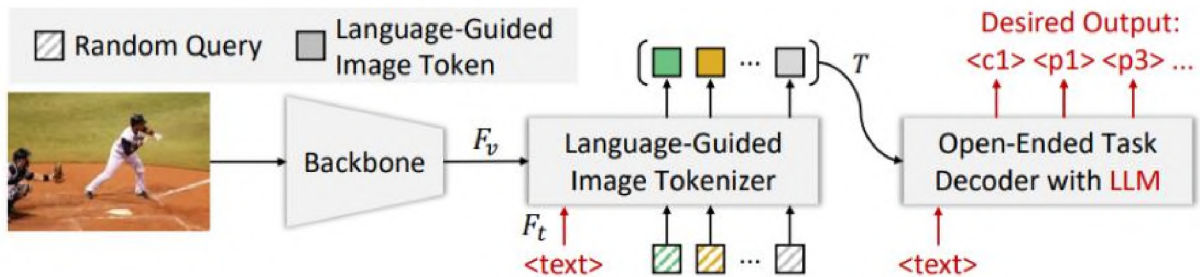


Рисунок 1.2 – Архітектура VisionLLM



Рисунок 1.3 – Приклад роботи VisionLLM

Однак VisionLLM стикається з такими проблемами, як внутрішні розбіжності між модальностями та форматами завдань, конфлікти

багатозадачності, а також потенційні проблеми з інтерпретацією та прозорістю складних процесів прийняття рішень.

Уніфікований інтерфейс для завдань на мові бачення [12]. Розглянемо мультимодальну LLM MiniGPT-v2 (рис. 1.4). Вона призначена для уніфікації різних завдань із мовою бачення, використовуючи різні ідентифікатори завдань для підвищення ефективності навчання та продуктивності.

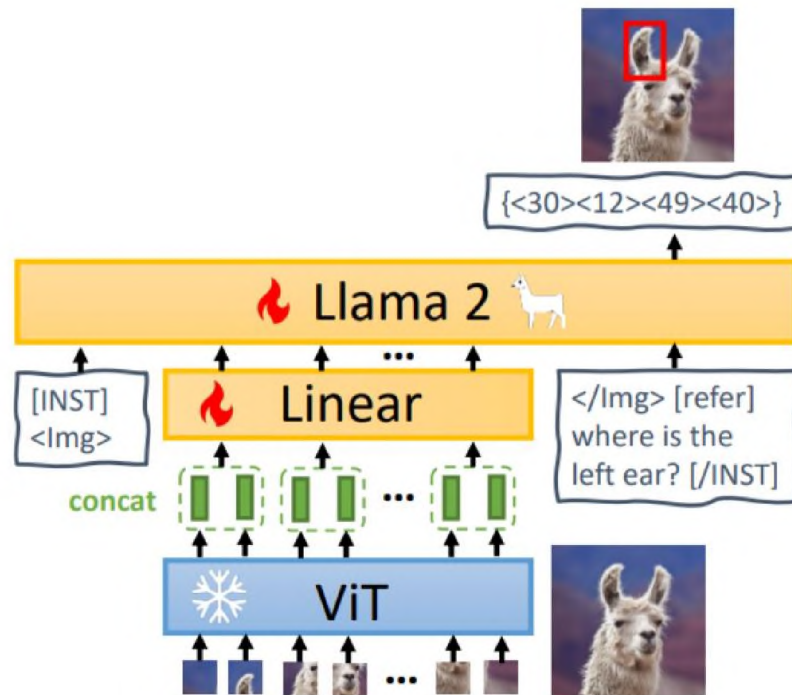


Рисунок 1.4 – MiniGPT-v2 в ролі уніфікований інтерфейс для багатозадачного навчання мови бачення

MiniGPT-v2 спрямована на вирішення проблем інтеграції візії та мови, потенційно вдосконалюючи CLIP шляхом підвищення адаптивності та продуктивності завдань у різноманітних візуальних та текстових завданнях. Вона також може подолати обмеження в інтерпретації, тонкому розумінні та кастомізації завдань, що властиві як моделям CLIP, так і VisionLLM. Модель поєднує в собі візуальні токени з візуального кодера ViT за допомогою трансформерів і самоувагу для обробки ділянок зображення. Вона використовує 3-ступеневу стратегію навчання на слабо позначених наборах даних «зображення-текст» і точних наборах даних «зображення-текст». Це

покращує його здатність виконувати такі завдання, як опис зображень, візуальні відповіді на запитання та підписи до зображень. Модель перевершила MiniGPT-4, LLaVA та InstructBLIP у бенчмарках і чудово продемонструвала гарну адаптованість до нових завдань. Розглянемо приклад, що відповідає рис. 1.5. Користувач надає запит: «Опишіть об'єкти на цьому зображенні». MiniGPT-v2 надає наступну відповідь. «Промислові машини – у приміщенні знаходиться кілька промислових машин, які використовуються для виготовлення або обробки матеріалів. Верстаки – верстаки забезпечують поверхні для таких завдань, як складання, ремонт або тестування. Висока стеля зі сталевими балками – кімната має високу стелю з видимими сталевими балками. Захисні шоломи: щонайменше двоє осіб на зображенні одягнені в захисні шоломи».



Рисунок 1.5 – MiniGPT-v2 в ролі уніфікований інтерфейс для багатозадачного навчання мови бачення

Проблеми цього мультимодального AI полягають у тому, що він іноді демонструє галюцинації [13] під час створення описів зображень. Галюцинації становлять серйозну проблему, особливо в критичних сферах, таких як медицина чи наука, де поширення неправдивої інформації може мати значні наслідки. Тобто, це випадки, коли моделі AI створюють відповіді, що містять неправдиву або оманливу інформацію, подану як факт. Наприклад,

дослідження показали, що моделі, такі як ChatGPT, можуть надавати вигадані посилання на наукові статті або неправильно інтерпретувати існуючі дані. Крім того, він може описувати неіснуючі візуальні об'єкти або неточно визначати місцезнаходження заземлених об'єктів.

Система з покращеною мовою (Language-Enabled Neural System, LENS) використовує мовні моделі, наприклад, GPT, у поєднанні з іншими компонентами AI для розв'язання складних завдань (рис. 1.6). Вони можуть конкретизувати візуальні концепції за допомогою зовнішніх словників, але не справляються з нульовими або невеликою кількістю завдань і вимагають ретельного тонкого налаштування для більш широких застосувань.

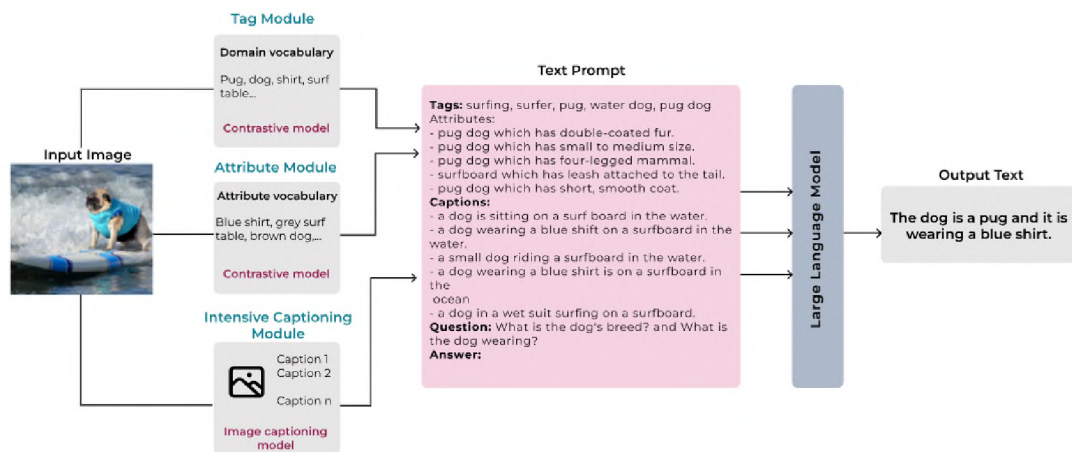


Рисунок 1.6 – Архітектура LENS

Щоб вирішити цю проблему, модель LENS інтегрує контрастне навчання зі словником з відкритим вихідним кодом для тегування зображень у поєднанні із замороженими LLM (попередньо навчена модель, яка використовується без подальших налаштувань) [14]. Модель LENS починається з вилучення функцій із зображень за допомогою трансформерів зору, таких як ViT і CLIP. Ці візуальні функції інтегровані з текстовою інформацією, обробленою LLM, такими як GPT-4, що дозволяє виконувати такі завдання, як створення описів, відповіді на запитання та виконання візуального мислення. Завдяки багатоетапному тренувальному процесу LENS поєднує візуальні та текстові дані за допомогою крос-модальних механізмів

уваги. Цей підхід підвищує продуктивність у завданнях розпізнавання об'єктів і завдання на VLM, без детального тонкого налаштування. Структуроване бачення та мовні концепції (SVLC) включають атрибути, відносини та стани, які зустрічаються як у текстових описах, так і в зображеннях [15]. Сучасні VLM мають проблеми з розумінням структурованого бачення. Щоб вирішити цю проблему, було запроваджено підхід, заснований на даних, який спрямований на покращення розуміння SVLC без необхідності використання додаткових спеціалізованих наборів даних. Цей підхід передбачав маніпулювання текстовими компонентами в існуючих наборах даних перед навчанням Vision And Language (VL), щоб підкреслити SVLC. Його методи включають розбір на основі правил і генерацію альтернативних текстів за допомогою мовних моделей. Експериментальні результати на кількох наборах даних продемонстрували значне покращення до 15 % у розумінні SVLC, забезпечуючи при цьому надійну продуктивність у завданнях розпізнавання об'єктів. Метод був спрямований на пом'якшення «зміщення об'єкта», яке, зазвичай, спостерігається в моделях VL, навчених з контрастними втратами, тим самим підвищуючи застосовність у таких завданнях, як виявлення об'єктів і сегментація зображення. В цілому, інтеграція LLM з CV за допомогою мультимодального AI, такого як VLM, є трансформаційним прогресом у галузі AI. Поєднуючи розуміння природної мови з візуальним сприйняттям, ці моделі гарно справляються з підписами до зображень і візуальні відповіді на запитання.

### **1.3 Проблемні аспекти застосування візуальних асистентів**

Візуальний асистент для LLM є важливим кроком у розвитку AI, який об'єднує обробку природної мови та візуальної інформації. Ця інтеграція дозволяє мовним моделям працювати не лише з текстом, але й аналізувати та

взаємодіяти з візуальними даними, що значно розширює їх функціональність. Завдяки цьому такі системи можуть забезпечувати більш повні, адаптивні та контекстуально обґрунтовані рішення для широкого спектра завдань, включаючи особисте асистування, автоматизацію виробничих процесів та покращення систем управління. Однак розвиток таких систем стикається з низкою викликів. Одним із головних є технічні складнощі, пов'язані з навчанням моделей. Для досягнення високої точності та узгодженості в обробці тексту і зображень моделі повинні тренуватися на великих наборах даних, що містять пари «зображення-текст». Це вимагає значних обчислювальних ресурсів та часу. Крім того, обробка візуальної інформації є надзвичайно ресурсозатратною, що ускладнює використання таких систем на пристроях із обмеженими можливостями, наприклад, смартфонах чи мікроконтролерах. Ще однією проблемою є етичні та правові аспекти. Використання зображень для навчання може порушувати права на конфіденційність, особливо якщо ці зображення містять особисті дані або використовуються без згоди їхніх власників. Крім того, мовні моделі та їхні візуальні компоненти часто успадковують упередження, що містяться в навчальних даних, що може призводити до неточностей або дискримінаційних рішень. Наприклад, система може неправильно інтерпретувати зображення людей через гендерні, расові чи інші стереотипи. Існує також ризик помилок в аналізі зображень, особливо у випадках, коли зображення є надто складними або містять незвичні для моделі елементи. Попри ці виклики, напрямки подальшого розвитку цього сегмента AI є перспективними. До ключових стратегій належать наступні положення.

1. Розробка моделей, що краще інтегрують текстову та візуальну інформацію. Такі системи будуть здатні одночасно аналізувати текст та зображення для формування більш точних відповідей і рекомендацій.

2. Оптимізація для пристроїв із обмеженими ресурсами. Це дозволить використовувати візуальних асистентів на компактних пристроях, таких як смартфони, IoT-гаджети чи портативні комп'ютери.

3. Розробка етичних норм та стандартів. Важливо забезпечити прозорість використання моделей, щоб мінімізувати ризики зловживання та упередженості. Це також включає розробку системи відповідальності за використання таких технологій.

4. Забезпечення зрозумілості роботи моделей для користувачів. Наприклад, пояснення, як система дійшла до певного висновку, може підвищити довіру користувачів та розширити їх можливості.

5. Інтеграція мультимодальних інтерфейсів. Візуальні асистенти можуть використовувати текст, голос та зображення для кращої комунікації з користувачами, що робить їх більш універсальними.

6. Адаптація для різних груп користувачів. Це передбачає створення рішень, які враховують індивідуальні потреби та можливості користувачів, включаючи людей із обмеженими можливостями.

Незважаючи на наявність технічних, етичних та правових бар'єрів, потенціал візуальних асистентів для LLM є значним. Вони мають змогу трансформувати різні аспекти життя, включаючи особисте асистування, освіту, медицину та промислові процеси. Інвестиції в їхній розвиток та вдосконалення є ключем до розкриття цього потенціалу.

## **Висновки до розділу 1**

Візуальні асистенти мовних моделей є одним із ключових напрямів сучасного розвитку AI, що інтегрує NLP та CV. Вони дозволяють розширити функціональність LLM, забезпечуючи можливість аналізу зображень, розпізнавання об'єктів, текстів та сцен, а також генерацію відповідей на основі візуального контексту. Інноваційні рішення, такі як GPT-4V, демонструють перспективи мультимодальних систем у завданнях медичної діагностики, створенні інтерактивних навчальних матеріалів, аналізі технічних об'єктів і творчих проєктів. Незважаючи на потужності, дані моделі

мають певні обмеження, пов'язані зі специфічністю контенту та нетиповими сценаріями. Подальший розвиток цього напрямку відкриває нові горизонти для інтеграції AI у повсякденне життя та різноманітні галузі науки й бізнесу.

Інтеграція CV з LLM є революційним кроком у розвитку мультимодального AI. Завдяки такому поєднанню, моделі набувають здатності виконувати завдання, що виходять за межі окремого текстового або візуального аналізу, забезпечуючи глибоке розуміння візуального контексту та створення описових текстів, релевантних до візуальних даних. Популярні підходи, такі як контрастне навчання та системи на кшталт CLIP, демонструють значний потенціал у сфері прогнозування нульового знімка, хоча й мають обмеження в деталізації та інтерпретації.

Нові рішення, такі як VisionLLM, MiniGPT-v2 та LENS, долають ці обмеження, дозволяючи точніше інтерпретувати візуальні концепції, узгоджувати їх із текстовими інструкціями та виконувати багатозадачні завдання з високою ефективністю. Однак, ці моделі все ще стикаються з проблемами, зокрема галюцинаціями, неточним відображенням об'єктів і обмеженнями інтерпретаційної прозорості, що залишається ключовим викликом для застосування у критичних сферах, таких як медицина.

Загалом, інтеграція LLM з CV відкриває нові горизонти для створення універсальних систем, здатних розуміти та обробляти як текстові, так і візуальні дані. Подальші дослідження спрямовані на усунення обмежень і вдосконалення здатності моделей до інтерпретації, що є основою для їх успішного впровадження в різноманітні галузі науки, техніки та бізнесу. Для подолання цих бар'єрів необхідно зосередитися на таких стратегіях: оптимізація моделей для роботи на пристроях з обмеженими ресурсами, розробка етичних стандартів і забезпечення прозорості систем, адаптація під потреби різних груп користувачів та інтеграція мультимодальних інтерфейсів. Ці заходи сприятимуть підвищенню довіри користувачів та розширенню сфер застосування візуальних асистентів. Попри наявні проблеми, потенціал цих технологій є значним, особливо в таких сферах, як освіта, медицина,

промисловість та особисте асистування. Подальші дослідження, інвестиції у вдосконалення моделей та їхню адаптацію до сучасних вимог сприятимуть успішному розкриттю цього потенціалу, забезпечуючи трансформацію різних аспектів життя та діяльності.

## РОЗДІЛ 2

### РОЗРОБКА МЕТОДИКИ ПОРІВНЯЛЬНОГО АНАЛІЗУ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ З ОПИСУ ЗОБРАЖЕНЬ

#### **2.1 Оцінювання великих мовних моделей з опису зображень шляхом опитування експертів на основі методу аналізу ієрархій**

Щоб забезпечити більш точний та всебічний порівняльний аналіз моделей для автоматичного опису зображень, необхідно виконати їх оцінку. В якості критерію можна застосовувати людську оцінку, наприклад, задіяти опитування користувачів та експертів. Оцінювання LLM, які генерують текстові описи зображень, є складним завданням, що вимагає врахування багатьох якісних і кількісних аспектів. Серед можливих підходів до оцінювання однією з найбільш обґрунтованих методик є опитування експертів. Цей метод передбачає залучення спеціалістів у відповідних галузях (наприклад, у сфері CV, лінгвістики, когнітивних наук) для оцінювання релевантності, точності, зрозумілості та повноти описів, які генерують мовні моделі. Для забезпечення надійності та репрезентативності оцінки обираються експерти, які мають значний досвід у відповідній галузі, зокрема в оцінюванні моделей AI або у використанні текстових описів для аналізу зображень. Для аналізу використовується стандартизований набір зображень із заданими параметрами, наприклад, популярні бази даних (MS COCO [11], ImageNet [16] та ін.) або спеціально підібрані зображення з урахуванням потреб дослідження. Для кожного зображення експерт отримує згенерований мовною моделлю опис та оцінює його за шкалою (наприклад, від 1 до 5 балів) за кожним критерієм. Додатково експерт може залишати коментарі щодо недоліків чи переваг опису. Отримані дані підлягають статистичному аналізу для визначення середніх показників, відхилень, а також кореляцій між оцінками різних критеріїв. Зокрема, порівнюються результати різних моделей або версій однієї моделі для виявлення їхніх

сильних та слабких сторін. Експертам можуть пропонуватись конкретні критерії. Одним із існуючих підходів є використання методу аналізу ієрархій (MAI) [17] в поєднанні з експертними оцінками. Такий підхід дозволяє систематично структурувати процес оцінювання та приймати рішення на основі комплексного аналізу факторів. Метод аналізу ієрархій передбачає побудову багаторівневої моделі, яка розділяє основну мету оцінювання (наприклад, визначення якості мовної моделі) на декілька рівнів критеріїв і підкритеріїв. Для завдання оцінки моделей опису зображень критеріями може бути кілька.

1. Точність відповідності опису контенту зображенням. Сюди відноситься релевантність опису до зображення та здатність моделі розпізнавати деталі (об'єкти, дії, сцени).

2. Лінгвістична якість тексту може містити граматичну правильність; логічну послідовність та зв'язність; відсутність повторень і тавтології.

3. Інноваційність і гнучкість моделі – можливість генерації нетривіальних описів та здатність адаптуватися до специфічних доменів.

Опитування експертів здійснюється для отримання вагових коефіцієнтів, які відображають важливість кожного критерію, а також для порівняння моделей на основі парних оцінок. Експерти порівнюють моделі за кожним критерієм, що дозволяє визначити їх відносну перевагу. Вагові коефіцієнти обчислюються на основі узгоджених парних порівнянь за допомогою спеціальних математичних методів, зокрема через розрахунок власних векторів матриць порівнянь.

Результатом такого аналізу є кількісне ранжування мовних моделей за їхньою здатністю ефективно генерувати описи зображень. MAI дозволяє не лише зробити оцінювання більш об'єктивним, а й врахувати суб'єктивні аспекти експертних суджень. Це робить метод особливо цінним для оцінювання LLM, де комплексність і багатовимірність проблеми ускладнюють застосування суто об'єктивних метрик. MAI було розроблено з ціллю об'єднати раціональність та інтуїтивність задля обрання оптимальної

альтернативи із множини альтернатив по відношенню до певних критеріїв. В ході цього процесу, той хто приймає рішення здійснює попарні порівняння суджень, на основі яких отримує оцінки кожного варіанту вирішення поставленої проблеми, альтернативи. Хоч MAI і не гарантує узгодженості в судженнях, але надає ефективний інструментарій для перевірки поточного її рівня. Метод опитування експертів дозволяє отримати якісну оцінку моделей завдяки врахуванню людської інтерпретації, яка є критично важливою в задачах генерації тексту. Водночас, його обмеження включають суб'єктивність оцінок, необхідність значного часу та ресурсів для залучення фахівців, а також можливу упередженість експертів у разі недостатньо чітких критеріїв. Опитування експертів є ефективним підходом для оцінювання LLM у завданнях генерації описів зображень, особливо у випадках, коли автоматичні метрики (наприклад, BLEU чи ROUGE) не забезпечують повноцінної оцінки. Комбінування цього методу з автоматизованими підходами дозволяє досягти більш точного та комплексного розуміння якості роботи мовних моделей.

## **2.2 Оцінювання великих мовних моделей з опису зображень на основі метрик**

В свою чергу, для порівняльного аналізу LLM, які генерують описи зображень, використовуються різні автоматичні метрики. Серед найбільш поширених є BLEU, ROUGE [18] та CIDEr [19]. Вони дозволяють кількісно оцінити відповідність згенерованого опису до еталонних текстів. Розглянемо детальніше кожен з цих метрик та способи їх застосування.

Bilingual Evaluation Understudy (BLEU) [20] – це одна з перших і найпоширеніших метрик для оцінки якості машинного перекладу та автоматичної генерації тексту. Вона вимірює схожість між згенерованим текстом (кандидатом) і одним або кількома еталонними текстами шляхом

аналізу збігів  $n$ -грам. Метрика порівнює дані  $n$ -грами (послідовності з  $n$  слів) у згенерованому тексті з  $n$ -грамами в еталонних текстах для  $n$  від 1 до 4. При цьому враховується максимальна кількість збігів  $n$ -грам, щоб уникнути переоцінки через повторення слів. Щоб заохотити повноту опису, застосовується штраф за стислість (Brevity Penalty), якщо згенерований текст коротший за еталонний. BLEU розраховується за виразом:

$$BLEU = Brevity Penalty \cdot \exp\left(\sum_{n=1}^N \omega_n \log p_n\right), \quad (2.1)$$

де  $p_n$ , – модифікована прецесія для  $n$ -грам;

$\omega_n$  – вагові коефіцієнти (зазвичай, рівні).

BLEU дозволяє оцінити, наскільки добре модель відтворює еталонні описи з точки зору точності використання слів і фраз. Вищі значення BLEU свідчать про більшу схожість з еталонними описами.

Наступний варіант – Recall-Oriented Understudy for Gisting Evaluation (ROUGE). Це набір метрик, орієнтованих на відтворення (Recall), які широко використовуються для оцінки якості автоматичних рефератів та узагальнень (рис. 2.1). ROUGE метрики допомагають визначити, наскільки повно модель охоплює інформацію з еталонних описів. За допомогою різних варіантів ROUGE можна оцінити різні характеристики згенерованих описів (наприклад, точність або послідовність).

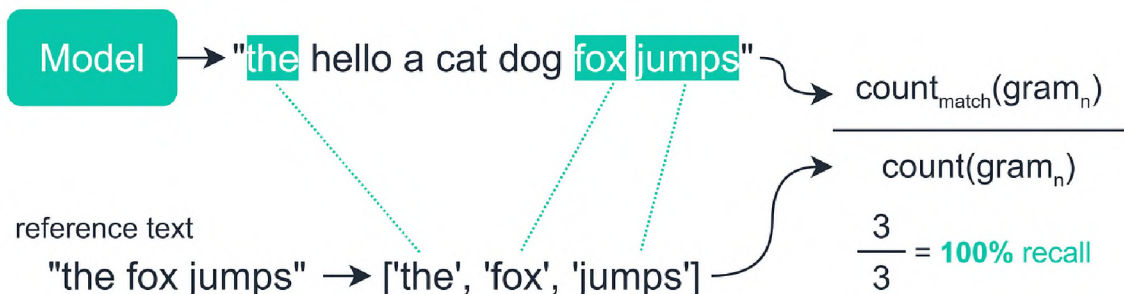


Рисунок 2.1 – Метрика ROUGE

Для опису зображень найбільш релевантними є: ROUGE-N – вимірює відсоток  $n$ -грам у еталонних текстах, які з'являються в згенерованому тексті;

ROUGE-L – базується на довжині найдовшої спільної підпоследовності (LCS) між згенерованим та еталонним текстами, що враховує як довжину збігів, так і їхню послідовність.

Ще може використовуватись Consensus-based Image Description Evaluation (CIDEr). Це спеціалізована метрика, розроблена для оцінки якості описів зображень. Вона враховує як схожість з еталонними описами, так і статистичну рідкість слів та фраз у корпусі текстів. Кожне слово або фраза зважується за допомогою методу Term Frequency-Inverse Document Frequency (TF-IDF), щоб надати більшої ваги унікальним та інформативним термінам. Частота термінів (TF) – це відношення кількості разів, коли слово зустрічається в документі, до загальної кількості слів у документах. Обернена частота даних (IDF) – призначає вищу вагу рідкісним словам у текстовому корпусі. По суті, TF-IDF – це метод пошуку інформації, який зважує частоту терміну (TF) та його обернену частоту документа (IDF). Кожне слово або термін має свою відповідну оцінку TF та IDF. Добуток оцінок TF та IDF терміну ( $TF \cdot IDF$ ) називається вагою цього терміну. Крім цього, обчислюється косинусна схожість між векторами згенерованого та еталонних описів, зважаючи на TF-IDF ваги (порівняння  $n$ -грам), а також агрегування результатів – результати для всіх  $n$ -грамів ( $n$  від 1 до 4) агрегуються для отримання фінального CIDEr значення. CIDEr краще враховує особливості описів зображень, ніж загальні метрики, завдяки TF-IDF зважуванню. Дослідження показують, що CIDEr має кращу кореляцію з оцінками людей щодо якості описів.

В цілому, вищі значення вказаних метрик, зазвичай, свідчать про кращу якість опису. Як наслідок, можна провести статистичні тести (наприклад,  $t$ -тест) для визначення значущості відмінностей між моделями.

Крім BLEU, ROUGE і CIDEr, для оцінювання якості генерації описів зображень LLM можуть застосовуватись й інші метрики. До основних з них відносять наступні. Semantic Propositional Image Caption Evaluation (SPICE) – фокусується на семантичному аналізі. Вона будує графи, що відображають

значення опису (семантичні трійки – суб’єкт, предикат, об’єкт) і порівнює їх із еталонними. Основна перевага: оцінює не просто схожість текстів, а їхнє семантичне наповнення. Metric for Evaluation of Translation with Explicit ORdering (METEOR) використовує лексичний збіг між еталонним та згенерованим описами, враховуючи синоніми, стовбурні форми слів і порядок слів. Вона також додатково враховує вагу різних збігів, що робить її більш чутливою до змістовних відмінностей. Semantic Propositional Image Description Evaluation (SPIDEr) – це комбінація метрик CIDEr і SPICE, яка об’єднує чутливість CIDEr до частотних збігів з глибоким семантичним аналізом SPICE. Така метрика рекомендується для більш збалансованої оцінки. Word Mover’s Distance (WMD) обчислює «відстань» між згенерованим і еталонним текстом у векторному просторі. В її основі – моделі Word2Vec або інші векторизації слів. Добре працює для оцінки семантичної близькості. BERTScore використовує попередньо натреновані мовні моделі (наприклад, BERT) для порівняння згенерованих і еталонних текстів на рівні ембеддингів. Вона дає змогу оцінювати семантичну близькість на рівні фраз і контексту. CLIPScore базується на моделі CLIP, яка порівнює текст із візуальним контекстом (зображенням). Вона використовується для оцінки, наскільки текстовий опис відповідає вхідному зображенню та є унікальною через те, що враховує візуальний контекст, а не лише текстовий збіг. Self-BLEU оцінює різноманітність згенерованих описів і застосовується для аналізу однотипності текстів, генерованих моделлю. Flicker Distance обчислює ступінь візуальної відповідності між описом і зображенням, аналізуючи об’єкти, дії, сцени тощо. Це частіше використовується для специфічних задач візуального контексту. Coverage Score вимірює, наскільки добре згенеровані описи охоплюють всі аспекти зображення (наприклад, кількість об’єктів, дій або деталей, що описуються). Метрика VIFIDEL (Visual Fidelity) спрямована на оцінку вірності опису щодо візуального змісту. Використовується для задач, де важливе точне відповідність візуальним даним. В цілому, для глибшого аналізу, зазвичай,

застосовують кілька метрик одночасно (комбінація метрик), наприклад: CIDEr + SPICE – для балансу між частотним збігом і семантичним аналізом; BERTScore + CLIPScore – для оцінки тексту з урахуванням зображення. Важливо підібрати метрику відповідно до цілей і специфіки вашої задачі.

### **2.3 Методи prompting у роботі з мовними моделями**

Сучасні мовні моделі, наприклад, GPT, активно використовуються для вирішення широкого спектру завдань. Однією з ключових методик їх ефективного використання є застосування різних методів prompting, що включають zero-shot prompt, Chain-of-Thought prompt, системний prompt та prompt-tuning та ін. Кожен з цих методів має особливості, переваги та сфери застосування, що робить їх незамінними інструментами в наукових та прикладних дослідженнях. Тому надалі розглянемо їх детальніше.

Zero-shot prompt передбачає використання мовної моделі без надання явних прикладів виконання завдання. Користувач формує запит у вигляді тексту, що містить завдання, яке модель повинна виконати, спираючись виключно на навчені уявлення. Zero-shot prompt підходить для широкого кола завдань та для нього не потрібна підготовка додаткових даних або прикладів. Тобто результат залежить від попереднього навчання моделі. При цьому, модель може видавати відповіді низької точності, особливо складних завдань. Відсутність контексту або прикладів може призвести до інтерпретації завдання не так, як очікує користувач. Наприклад, на prompt «Напиши вірш про природу.» модель надає у відповідь текст вірша створений без додаткового контексту.

Prompt з невеликою кількістю прикладів (Few-shot prompt). Незважаючи на те, що LLM демонструють чудові можливості Zero-shot, вони все одно не справляються з більш складними завданнями при використанні налаштування Zero-shot. Few-shot prompt може бути використано як техніка

для забезпечення навчання в контексті, де користувач надає демонстрації в prompt, щоб спрямувати модель до кращої продуктивності. Демонстрації служать обумовленням для наступних прикладів, де б хотіли, щоб модель генерувала відповідь. Можна спостерігати, що модель якимось чином навчилася виконувати завдання, надавши їй лише один приклад. Для більш складних завдань можна експериментувати зі збільшенням демонстрацій (наприклад, 3, 5, 10 прикладів). Згідно [21], формат, який використовується, також відіграє ключову роль у продуктивності, навіть якщо просто використовувати випадкові мітки, це набагато краще, ніж повна відсутність міток. Додаткові результати показують, що вибір випадкових міток із справжнього розподілу міток (замість рівномірного розподілу) також допомагає. Стандартний prompt з кількома прикладами добре працює для багатьох завдань, але все одно не є ідеальною технікою, особливо коли маємо справу з більш складними завданнями на міркування.

Коли Zero-shot і Few-shot недостатні, це може означати, що все, що було вивчено моделлю, недостатньо для успішного виконання завдання. Звідси рекомендується почати думати про тонке налаштування моделей або експериментувати з більш просунутими техніками створення prompt.

Chain-of-Thought prompt (COT) спрямований на підвищення якості виконання складних завдань за рахунок поетапного міркування (рис. 2.2).

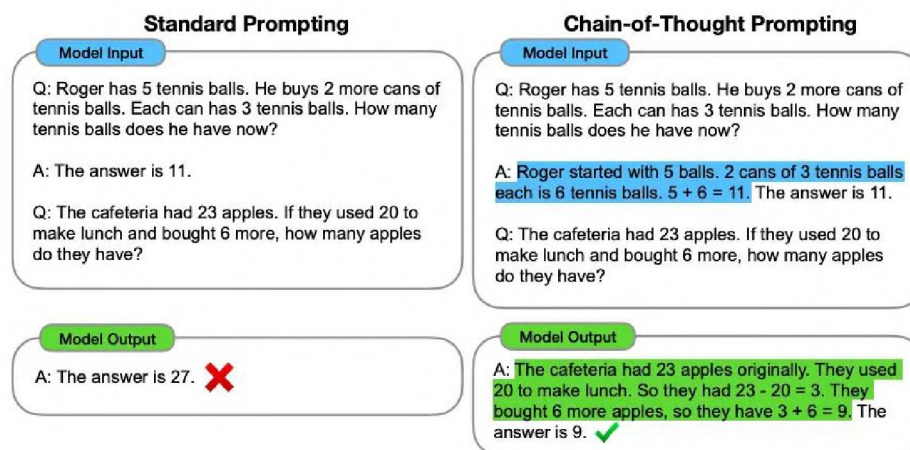


Рисунок 2.2 – Техніка COT

Цей варіант передбачає явну вказівку моделі міркувати покроково, що дозволяє краще структурувати процес рішення. Він дозволяє отримати доступ до логіки міркувань моделі. COT особливо ефективний для завдань, що потребують багатокомпонентних обчислень, наприклад математичних завдань або логічних висновків. По суті, він стимулює модель формулювати ідеї, ґрунтуючись на проміжних кроках. Однак, недостатньо навчені моделі можуть видавати неправильні проміжні кроки, а покрокова обробка збільшує обчислювальне навантаження. Наприклад, на prompt «Якщо підприємство має верстати, і воно придбало ще 5, а потім списало 2, скільки в нього залишилося? Будь ласка, поясни свої дії крок за кроком.» модель надає у відповідь поетапне обчислення із поясненнями на кожному етапі.

Системний (System) prompt – це підхід, при якому формується структурований запит, що задає контекст або роль моделі. Він включає опис моделі, її функцій та очікувань від її роботи. System prompt допомагає адаптувати відповіді під певні ролі та стилі спілкування та підходить як для наукових, так і для прикладних завдань, а за рахунок уточнення ролі зменшує ймовірність створення нерелевантного тексту. При цьому, він вимагає ретельного продумування формулювання prompt, а надмірно вузький контекст може обмежити можливості моделі.

Prompt-tuning є методом адаптації мовної моделі шляхом оптимізації параметрів prompt. Цей підхід може включати як ручне налаштування тексту prompt, так і автоматичну генерацію prompts через спеціальні алгоритми навчання. Це дозволяє досягти оптимальних результатів для вузькоспеціалізованих завдань та можливе використання алгоритмів ML для генерації та покращення prompts. Також він не потребує зміни параметрів моделі, що спрощує процес адаптації. До обмежень такого підходу відноситься більша складність у реалізації порівняно з іншими техніками, а також вимоги наявності даних для навчання. Прикладом застосування prompt-tuning є навчання моделі, де prompts оптимізуються до виконання конкретного завдання – аналізу тональності тексту.

Meta prompt – це просунута техніка підказок, яка зосереджується на структурних і синтаксичних аспектах завдань і проблем, а не на їхніх конкретних деталях змісту. Ця мета за допомогою Meta prompt полягає в тому, щоб побудувати більш абстрактний, структурований спосіб взаємодії з LLM, акцентуючи увагу на формі та шаблоні інформації над традиційними методами, орієнтованими на контент.

Згідно [22], ключові характеристики можна узагальнити наступним чином: надає перевагу формату та шаблону проблем і рішень над конкретним контентом; використовує синтаксис як керівний шаблон для очікуваної відповіді або рішення; використовує абстрактні приклади як рамки, ілюструючи структуру проблем і рішень, не зосереджуючись на конкретних деталях; застосовується в різних областях, здатна надавати структуровані відповіді на широкий спектр проблем; спирається на теорію типів, щоб підкреслити категоризацію та логічне розташування компонентів у prompt.

До переваг Meta prompt перед можна віднести: ефективність токенів (зменшує кількість необхідних токенів, зосереджуючись на структурі, а не на детальному контенті); справедливе порівняння (забезпечує більш справедливий підхід для порівняння різних моделей вирішення проблем шляхом мінімізації впливу конкретних прикладів); ефективність Zero-shot (можна розглядати як форму prompt з Zero-shot, де вплив конкретних прикладів зводиться до мінімуму).

Зосереджуючись на структурних моделях вирішення проблем, Meta prompt пропонує чітку дорожню карту для навігації складними темами, покращуючи можливості міркування LLM у різних сферах. Важливо зазначити, що Meta prompt також передбачає, що LLM має вроджені знання про конкретне завдання чи проблему, яка вирішується. Оскільки LLM можуть узагальнюватися до невидимих завдань, цілком можливо, що їх можна використовувати за допомогою Meta prompt, але продуктивність може погіршитися при більш унікальних і нових завданнях, як у випадку з технікою Zero-shot.

Самостійна послідовність (Self-Consistency) [233] Можливо є одним з найбільш просунутих методів швидкого проектування. Запропонований у [24], він спрямований на те, щоб «замінити наївне жадібне декодування, яке використовується в prompt CoT». Ідея полягає в тому, щоб взяти зразки з кількох різноманітних шляхів міркувань за допомогою кількох пострілів CoT і використовувати покоління для вибору найбільш послідовної відповіді. Це допомагає підвищити продуктивність підказок CoT у завданнях, що включають арифметику та міркування здорового глузду.

Prompt згенерованих знань (Generate Knowledge prompt). LLM продовжують удосконалюватися, і одна з популярних технік включає можливість включення знань або інформації від мовної моделі, щоб допомогти моделі робити більш точні прогнози. Тобто, згенерованні знання будуть використовуватися як частина prompt (рис. 2.3).

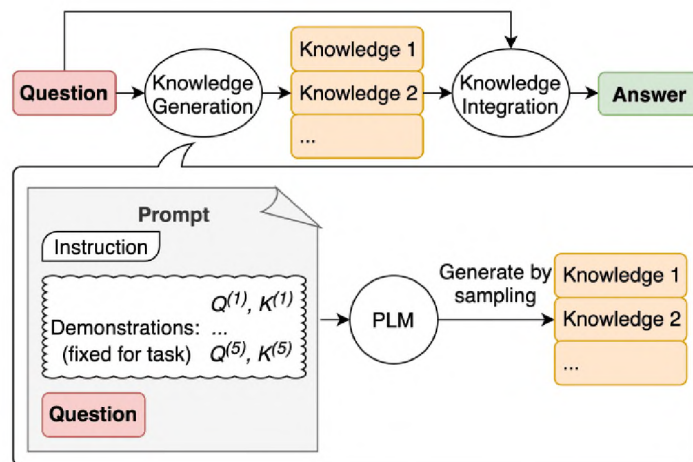


Рисунок 2.3 – Generate Knowledge prompt

Ланцюжок команд (Prompt Chaining). Для підвищення надійності та продуктивності LLM одним із важливих методів є розбиття завдань на підзавдання. Після того, як ці підзадачі були визначені, LLM отримує запит з підзавданням, а потім його відповідь використовується як вхідні дані для іншого запиту. Це так званий prompt chaining, коли завдання розбивається на підзадачі з ідеєю створити ланцюжок швидких операцій. Ланцюжок команд

корисний для виконання складних завдань, які LLM може бути важко вирішити, якщо йому буде запропоновано дуже детальний запит. У швидкому ланцюжку ланцюгові prompt виконують перетворення або додаткові процеси на згенерованих відповідях, перш ніж досягти кінцевого бажаного стану. Окрім досягнення кращої продуктивності, швидке зчеплення допомагає підвищити прозорість вашої програми LLM, підвищує керованість і надійність. Це означає, що користувачу може набагато легше налагоджувати проблеми з відповідями моделі, а також аналізувати та покращувати продуктивність на різних етапах, які потребують вдосконалення. Ланцюжок команд особливо корисний при створенні голосових помічників на базі LLM і покращенні персоналізації та користувацького досвіду програм. Ланцюжок може використовуватися в різних сценаріях, які можуть включати кілька операцій або перетворень. Наприклад, один із поширених випадків використання LLM передбачає відповіді на запитання про великий текстовий документ. Це допоможе, якщо розробите два різні prompt, де перший запит відповідає за вилучення релевантних цитат для відповіді на запитання, а другий запит приймає як вхідні дані цитати та оригінальний документ для відповіді на задане запитання. Іншими словами, користувач створює два різні prompt для виконання завдання відповіді на питання, яке задане в документі.

Дерево думок (Tree of Thoughts, ToT). Для складних завдань, які вимагають дослідження або стратегічного огляду вперед, традиційні або прості методи prompt не підходять. В [25, 26] запропоновано ToT, який узагальнює спонукання до ланцюга думок і заохочує до дослідження думок, які служать проміжними кроками для вирішення загальних проблем за допомогою мовних моделей. ToT підтримує дерево думок, де думки представляють зв'язні мовні послідовності, які служать проміжними кроками на шляху до вирішення проблеми (рис. 2.4). Цей підхід дозволяє LLM самооцінювати прогрес за допомогою проміжних думок, висловлених у напрямку до вирішення проблеми за допомогою обдуманого процесу міркування. Здатність LLM генерувати та оцінювати думки потім

поєднується з алгоритмами пошуку (наприклад, пошук по ширині та пошук за глибиною), щоб забезпечити систематичне дослідження думок з переглядом вперед і зворотним відстеженням.

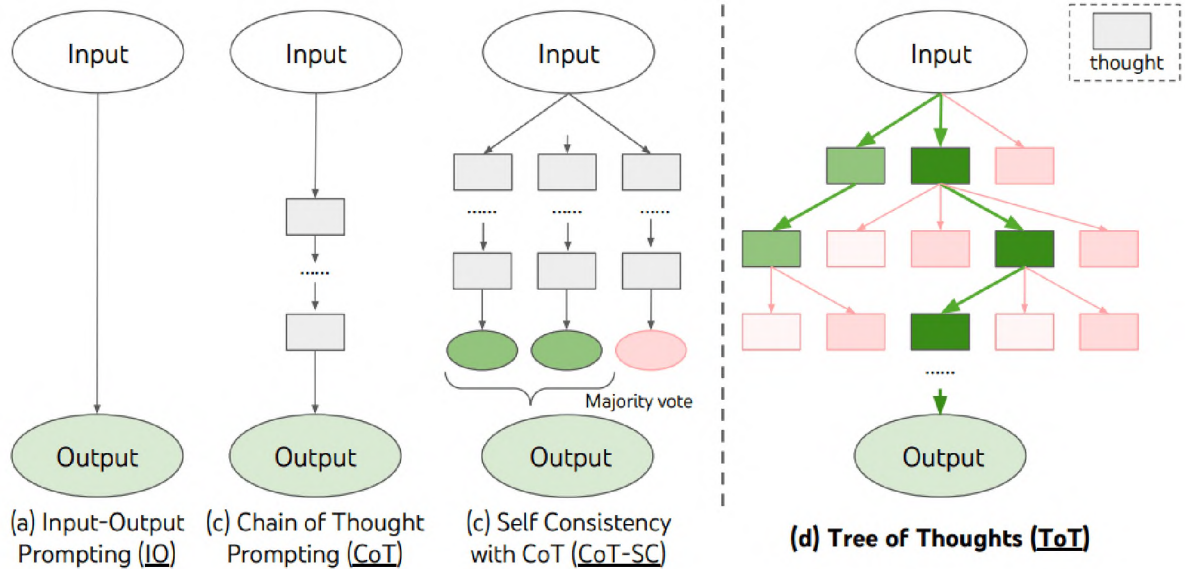


Рисунок 2.4 – Структура ToT

При використанні ToT для різних завдань потрібно визначити кількість кандидатів та кількість думок/кроків. В [27] запропоновано Tree-of-Thought Prompting, який застосовує основну концепцію з фреймворків ToT як просту техніку prompt, змушуючи LLM оцінювати проміжні думки в одному prompt.

Отримання доповненої генерації (Retrieval Augmented Generation, RAG). LLM загального призначення можуть бути точно налаштовані для досягнення кількох загальних завдань, таких як аналіз настроїв і розпізнавання іменованих сутностей. Ці завдання, як правило, не вимагають додаткових базових знань. Для більш складних і наукомістких завдань можна побудувати систему на основі мовної моделі, яка звертається до зовнішніх джерел знань для виконання завдань. Це забезпечує більшу фактичну узгодженість, підвищує надійність згенерованих відповідей і допомагає пом'якшити проблему «галюцинацій». Дослідники Meta AI представили метод RAG [28] для вирішення таких наукомістких завдань. RAG поєднує в

собі компонент пошуку інформації з моделлю генератора тексту. RAG можна точно налаштувати, а його внутрішні знання можна модифікувати ефективним чином і без необхідності перенавчання всієї моделі. RAG приймає вхідні дані та отримує набір релевантних/підтверджуючих документів із зазначенням джерела (наприклад, Вікіпедія). Документи об'єднуються як контекст з оригінальним вхідним запитом і подаються до генератора тексту, який виробляє остаточний результат. Це робить RAG адаптивним для ситуацій, коли факти можуть змінюватися з часом. Це дуже корисно, оскільки параметричні знання LLM є статичними. RAG дозволяє мовним моделям обходити перенавчання, забезпечуючи доступ до найновішої інформації для генерації надійних результатів за допомогою генерації на основі пошуку. RAG генерує відповіді, які є більш фактичними, конкретними та різноманітними при тестуванні на запитання MS-MARCO та Jeopardy. RAG також покращує результати з перевірки фактів FEVER. Це показує потенціал RAG як життєздатного варіанту для покращення виходів мовних моделей у вимогливих до знань завданнях. Останнім часом ці підходи стали більш популярними та поєднуються з популярними LLM, такими як ChatGPT, для покращення можливостей і узгодженості фактів.

Автоматичне міркування та використання інструментів (Automatic Reasoning and Tool-use, ART). Поєднання підказок та інструментів CoT у взаємопов'язаному вигляді показало себе як сильний та надійний підхід до вирішення багатьох завдань за допомогою LLM. Ці підходи, як правило, вимагають демонстрації ручного виготовлення конкретних завдань і ретельно прописаного чергування поколінь моделей з використанням інструментів. В [29] запропонувати новий фреймворк, який використовує заморожений LLM для автоматичної генерації проміжних кроків міркування як програма. ART працює наступним чином (рис. 2.5). Отримавши нове завдання, він вибирає демонстрації багатоетапного міркування та використання інструментів із бібліотеки завдань. Під час тестування він призупиняє генерацію щоразу, коли викликаються зовнішні інструменти, та інтегрує їх вивід перед

відновленням генерації. ART заохочує модель до узагальнення з демонстрацій для розкладання нового завдання і Використовуйте інструменти у відповідних місцях, без пострілу. Крім того, ART є розширюваним, оскільки він також дозволяє людям виправляти помилки на етапах міркування або додавати нові інструменти, просто оновлюючи завдання та бібліотеки інструментів.

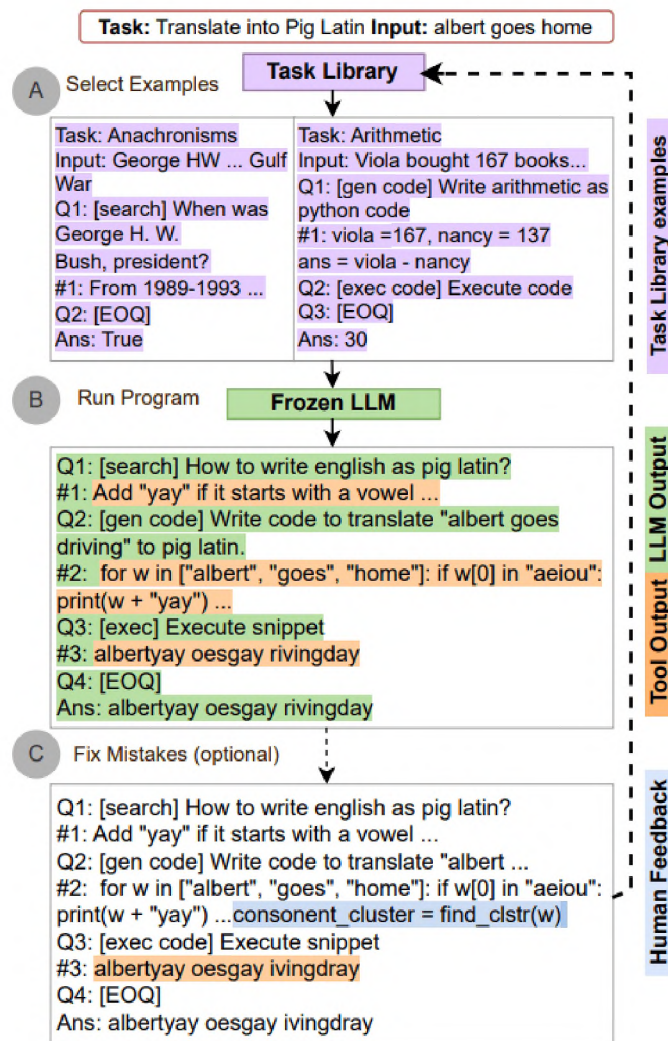


Рисунок 2.5 – Процес роботи ART [29]

ART значно покращує запит порівняно з кількома prompt та автоматичним COT на невидимих завданнях у тестах BigBench та MMLU, а також перевершує продуктивність prompt COT, створених вручну, коли враховуються відгуки людей.

Інженер автоматичних підказок (Automatic Prompt Engineer, APE) – фреймворк для автоматичної генерації та вибору команд запропонований в [30], механізм його функціонування наведений на рис. 2.6.

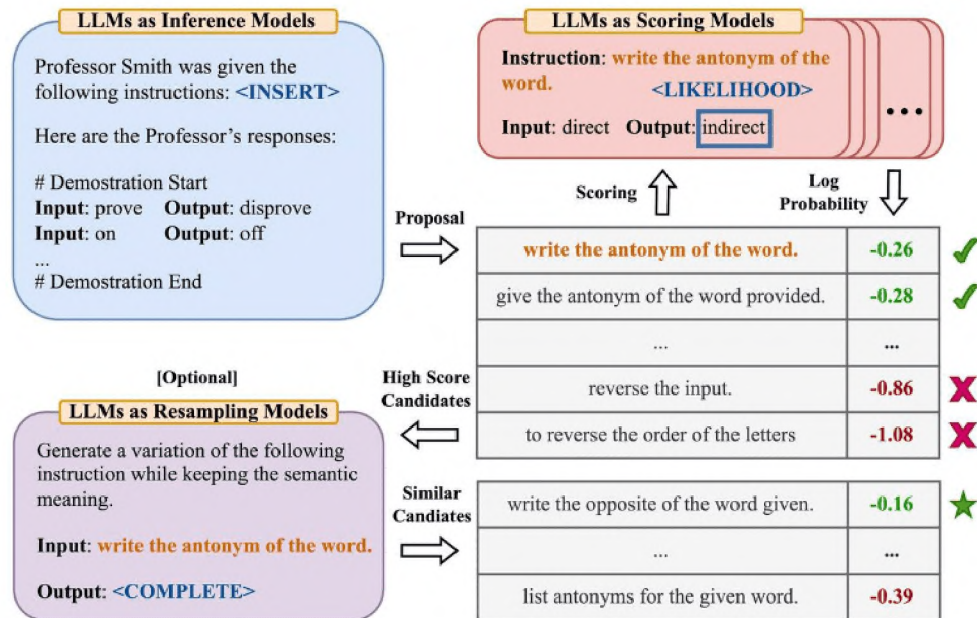


Рисунок 2.6 – Automatic Prompt Engineer [30]

Проблема генерації інструкцій представлена у вигляді синтезу природної мови, що розглядається як задача оптимізації «чорного ящика» з використанням LLM для генерації та пошуку кандидатських рішень. Перший крок включає велику мовну модель (як модель логічного висновку), яка отримує вихідні демонстрації для генерації кандидатів у інструкції для завдання. Ці рішення для кандидатів будуть орієнтиром у процедурі пошуку. Інструкції виконуються за допомогою цільової моделі, а потім вибирається найбільш відповідна інструкція на основі обчислених оціночних балів. APE виявляє кращий prompt COT з нульовим пострілом, ніж сконструйований людиною prompt «Давайте подумаємо крок за кроком» [31]. Prompt «Давайте розберемося з цим крок за кроком, щоб переконатися, що у нас є правильна відповідь» викликає міркування на основі COT і покращує продуктивність за тестами MultiArith і GSM8K.

Активний запит (Active-prompt). Методи CoT спираються на фіксований набір прикладів, анотованих людиною (рис. 2.7). Проблема з цим полягає в тому, що приклади можуть бути не найефективнішими прикладами для різних завдань.

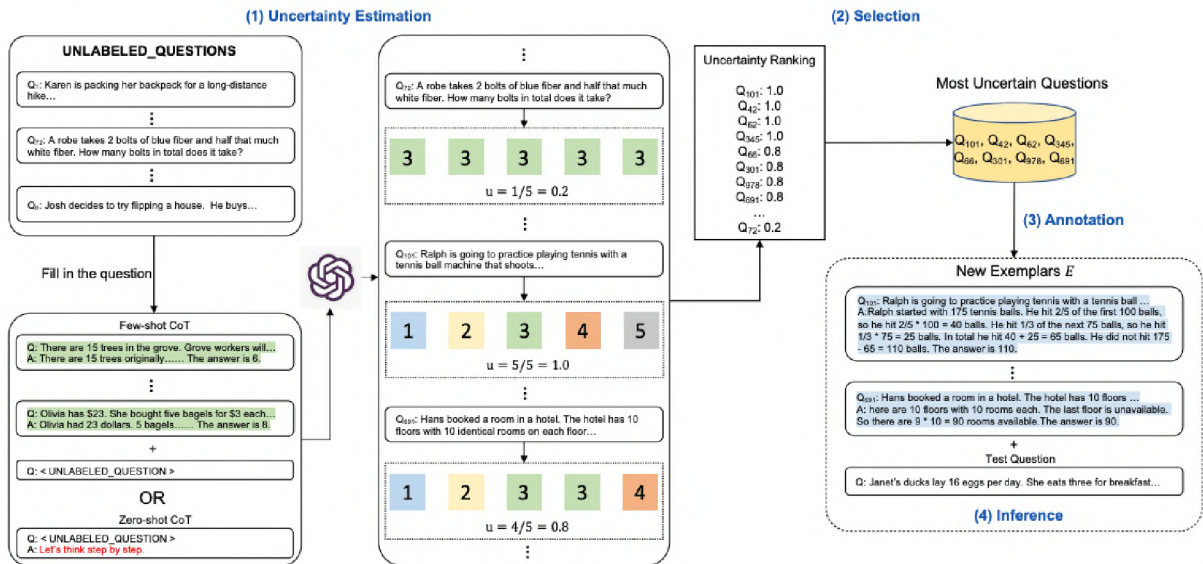


Рисунок 2.7 – Active-Prompt [32]

Щоб вирішити цю проблему, в [32] запропоновано новий підхід до prompt під назвою Active-Prompt для адаптації LLM до різних прикладів prompt для конкретного завдання (анотованих міркуваннями CoT, розробленими людиною). Першим кроком є запит до LLM з кількома прикладами CoT або без них. Для набору навчальних питань генеруються можливі відповіді. Метрика невизначеності розраховується на основі  $k$  відповідей (використовується незгода). Найбільш невизначені питання вибираються для анотації людиною. Потім для висновку про кожне питання використовуються нові анотовані приклади.

Спонування до спрямованого стимулу (Directional Stimulus Prompting) – це техніка для кращого керівництва LLM у генерації бажаного резюме [33]. Налаштована політика LLM навчена генерувати стимул/prompt. Спостерігається все більше застосування методів навчання з підкріпленням

(RL) для оптимізації LLM. На рис. 2.8 показано, як метод Directional Stimulus Prompting порівнюється зі стандартним prompt. Політика LM може бути компактною моделлю, оптимізованою для генерації prompts, направляючи заморожену LLM у режимі «чорного ящика».

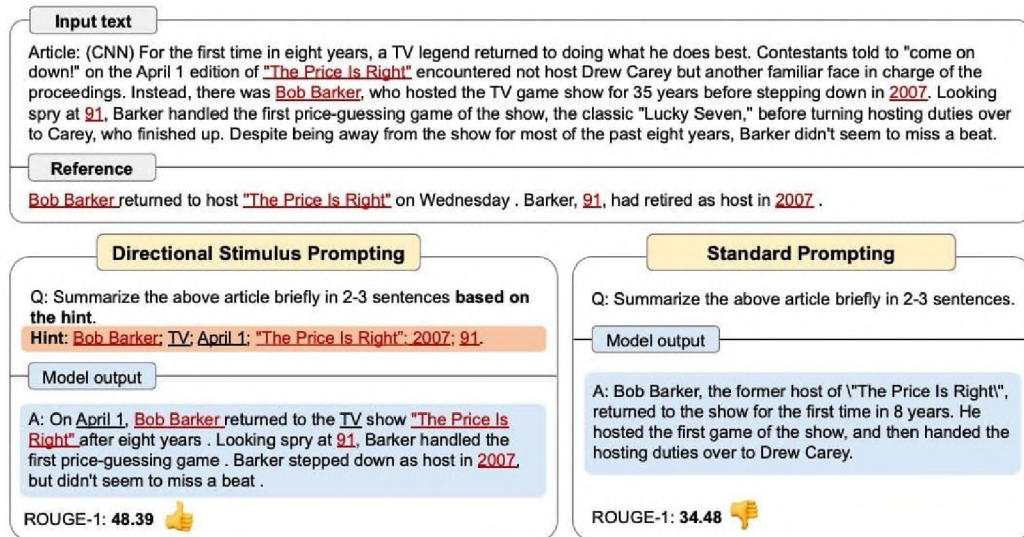


Рисунок 2.8 – Directional Stimulus Prompting

Програмно-орієнтовані мовні моделі (Program-Aided Language Models, PAL) – це метод, який використовує LLM для читання задач на природній мові та генерації програм як проміжних етапів міркування [34]. PAL відрізняються від prompt COT тим, що замість використання тексту у довільній формі для отримання рішення він розвантажує крок рішення до програмного середовища виконання, такого як інтерпретатор Python (рис. 2.9).

ReAct prompt – концепція, в якій LLMs використовуються для генерації як COT, так і дій, специфічних для завдання, в порядку, що чергується [35]. Генерація логічних ланцюжків дозволяє моделі формувати, відстежувати та оновлювати плани дій, а також обробляти винятки. Етап дій забезпечує взаємодію із зовнішніми джерелами, такими як бази знань чи середовища, для збирання додаткової інформації.

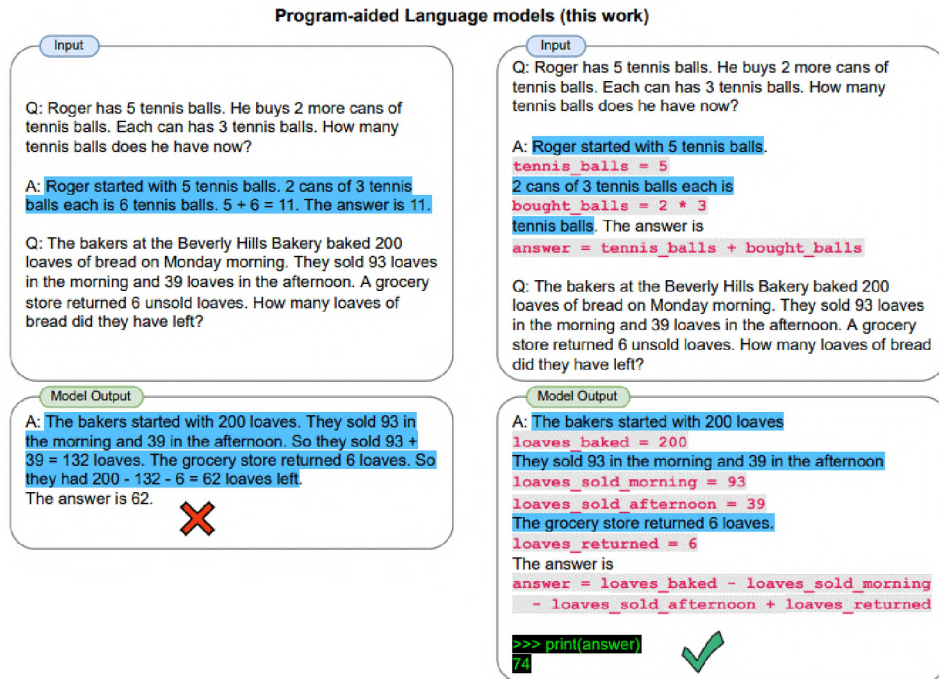


Рисунок 2.9 – Метод Program-Aided Language Models

Результати досліджень показують, що ReAct перевершує кілька сучасних базових підходів у завданнях обробки мови та прийняття рішень. Крім того, ReAct сприяє покращенню інтерпретованості та надійності LLM з точки зору людини. ReAct може дозволити LLM взаємодіяти із зовнішніми інструментами для отримання додаткової інформації, яка призводить до більш надійних і фактичних відповідей. Prompt CoT показав здатність LLM виконувати трасування міркувань для отримання відповідей на питання, що включають арифметичні міркування та здоровий глузд. Але відсутність доступу до зовнішнього світу або нездатність оновити свої знання може призвести до таких проблем, як галюцинації фактів і поширення помилок. ReAct – це загальна парадигма, яка поєднує міркування та дію з LLM. ReAct спонукає LLM генерувати вербальні міркування, сліди та дії для завдання. Це дозволяє системі виконувати динамічні міркування для створення, підтримки та коригування планів дій, а також дозволяє взаємодіяти із зовнішнім середовищем (наприклад, Вікіпедією) для включення додаткової інформації в міркування. На рис. 2.10 показаний приклад ReAct та різні кроки, що необхідні для виконання відповідей на запитання.

Рефлексія (Reflexion) – це метод посилення мовних агентів з допомогою лінгвістичного зворотного зв'язку (рис. 2.11). Згідно [36], Reflexion – це нова парадигма «вербального» підкріплення, яка параметризує політику як кодування пам'яті агента в поєднанні з вибором параметрів LLM.

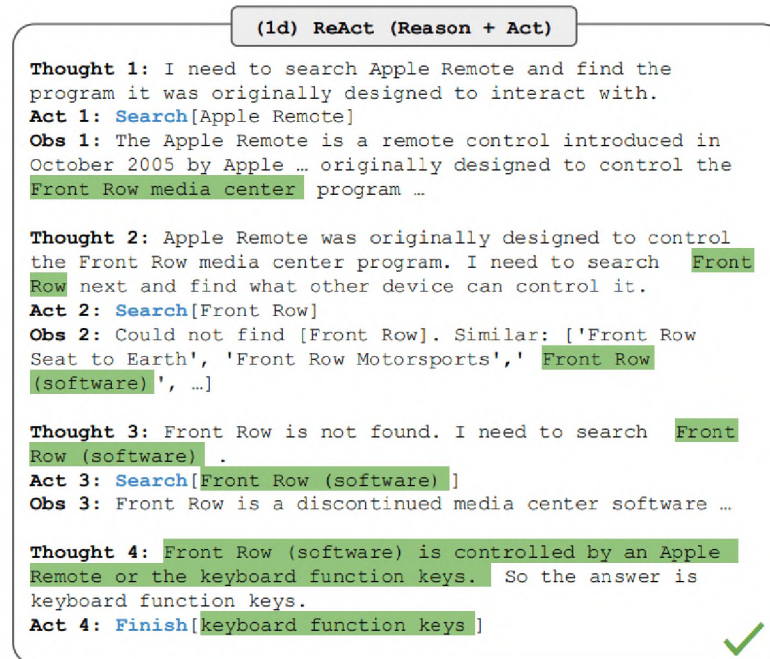


Рисунок 2.10 – Prompt ReAct [35]

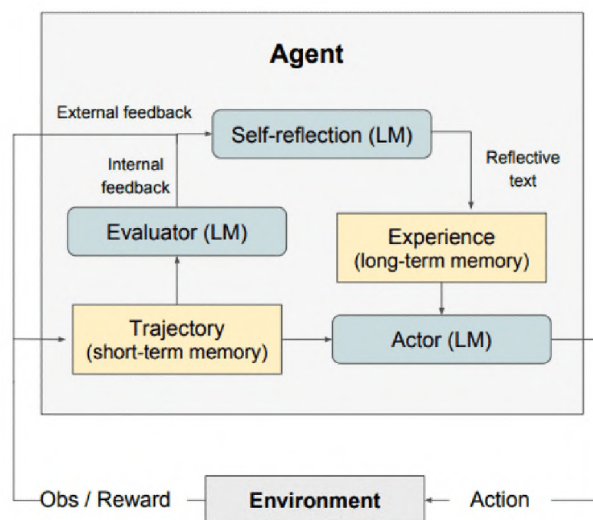


Рисунок 2.11 – Структура Reflexion

На високому рівні Reflexion перетворює зворотний зв'язок (будь то у вільній формі або скалярний) з навколишнього середовища в лінгвістичний

зворотний зв'язок, також званий саморефлексією, який надається як контекст агента LLM в наступному епізоді. Це допомагає агенту швидко та ефективно навчатися на попередніх помилках, що призводить до покращення продуктивності у вирішенні множини складних завдань. Як показано на рис. 2.11, Reflexion складається з 3-ох різних моделей. Actor генерує текст і дії на основі спостережень за станом. Actor виконує дію в обстановці і отримує спостереження, яке призводить до траєкторії. COT та використовуються в якості акторських моделей. Також додається компонент пам'яті для надання додаткового контексту агенту. Evaluator оцінює результати, вироблені Actor. Конкретно, він бере на вхід згенеровану траєкторію (також позначається як короткочасна пам'ять) і виводить оцінку винагороди. Залежно від завдання використовуються різні функції винагороди (для завдань на прийняття рішень використовуються LLM та евристика на основі правил). Self-Reflection генерує вербальні підкріплювальні сигнали, щоб допомогти Актору в самовдосконаленні. Ця роль досягається LLM і забезпечує цінний зворотний зв'язок для майбутніх випробувань. Для генерації конкретного та актуального зворотного зв'язку, який також зберігається в пам'яті, модель саморефлексії використовує сигнал винагороди, поточну траєкторію та його постійну пам'ять. Цей досвід (що зберігається в довготривалій пам'яті) використовується агентом для швидкого поліпшення прийняття рішень. Таким чином, ключовими етапами процесу рефлексії є (рис. 2.12): визначення завдання, генерація траєкторії, оцінка, виконання рефлексії та генерація наступної траєкторії. Рефлексія розширює рамки ReAct, вводячи компоненти самооцінки, саморефлексії та пам'яті. Експериментальні результати демонструють, що рефлексивні агенти значно покращують продуктивність при виконанні завдань AlfWorld щодо прийняття рішень, питаннях міркування в HotPotQA та завданнях програмування на Python на HumanEval. Рефлексія найкраще підходить для кількох випадків. Агент повинен вчитися методом проб і помилок: рефлексія призначена для того, щоб допомогти агентам поліпшити свою роботу, розмірковуючи про

минулі помилки і враховуючи ці знання в майбутніх рішеннях. Це робить його добре придатним для завдань, де агенту потрібно вчитися шляхом проб і помилок, таких як прийняття рішень, міркування та програмування.

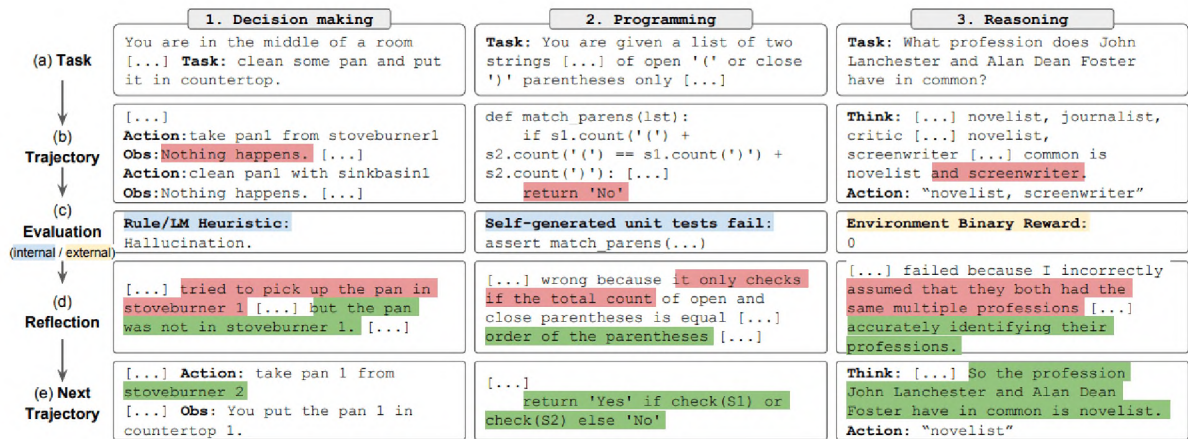


Рисунок 2.12 – Приклад того, як рефлексивний агент може навчитися ітеративно оптимізувати свою поведінку для вирішення різних завдань, таких як прийняття рішень, програмування та міркування

Традиційні методи навчання з підкріпленням непрактичні: традиційні методи навчання з підкріпленням (RL) часто вимагають великих навчальних даних і дорогого доопрацювання моделі. Reflexion пропонує легку альтернативу, яка не вимагає тонкого налаштування базової моделі мови, що робить її ефективною відносно даних та обчислювальних ресурсів. Reflexion використовує вербальний зворотний зв'язок, який може бути більш тонким і специфічним, ніж скалярні винагороди, що використовуються в традиційному RL. Це дозволяє агенту краще розуміти свої помилки і робити більш цілеспрямовані поліпшення в наступних випробуваннях. Рефлексія забезпечує більш інтерпретовану та експліцитну форму епізодичної пам'яті порівняно з традиційними методами RL. Саморефлексія агента зберігається в його пам'яті, що дозволяє легше аналізувати і розуміти процес його навчання.

Reflexion може бути ефективною в наступних завданнях. Послідовне прийняття рішень – агенти Reflexion покращують свою продуктивність у

завданнях AlfWorld, які передбачають навігацію в різних середовищах та виконання багатоетапних завдань; обґрунтування – Reflexion покращило продуктивність агентів на HotPotQA, наборі даних для відповідей на запитання, який вимагає обґрунтування кількох документів; програмування – Агенти Reflexion пишуть кращий код на таких тестах, як HumanEval і MBPP, досягаючи в деяких випадках найсучасніших результатів.

Однак існують деякі обмеження для використання Reflexion. Вона спирається на здатність агента точно оцінювати свою діяльність і генерувати корисні саморефлексії. Це може бути складним завданням, але очікується, що Reflexion з часом стане кращим, оскільки можливості моделей постійно вдосконалюються. Reflexion використовує ковзне вікно з максимальною ємністю, але для складних завдань вигідно використовувати складні структури, такі як векторне вбудовування або бази даних SQL. Існують обмеження для розробки на основі тестування у визначенні точних відображень входів-виходів (наприклад, недетермінована функція генератора та виходи функцій, на які впливає апаратне забезпечення).

Prompt Multimodal COT [37]. Традиційний COT фокусується на мовній модальності. На противагу цьому, Multimodal COT включає текст і бачення в двоетапну структуру (рис. 2.13).

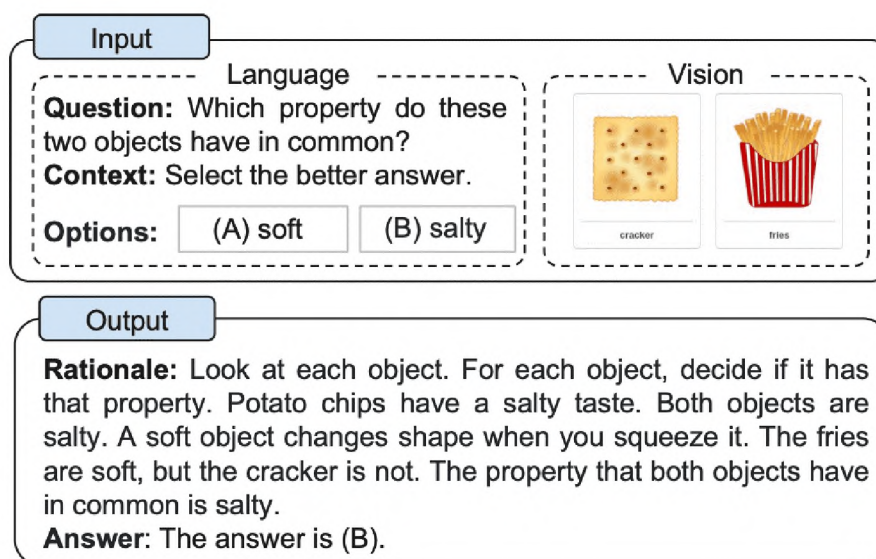


Рисунок 2.13 – Приклад мультимодальної задачі COT [37]

Перший крок передбачає генерацію обґрунтувань на основі мультимодальної інформації. Після цього слідує друга фаза – вивід відповідей, яка використовує інформативно згенеровані обґрунтування. Мультимодальна модель CoT (1B) перевершує GPT-3.5 у бенчмарку ScienceQA.

Графічні (Graph) prompts [38] – структура підказок для графіків для підвищення продуктивності під час виконання наступних завдань.

В цілому, кожна з описаних технік prompting надає унікальні можливості взаємодії з мовними моделями. Проведені дослідження свідчать, що найбільш поширеними техніками формування prompt є Zero-shot prompt, CoT та prompt-tuning. Zero-shot prompt зручний для швидкого тестування, Chain-of-Thought prompt покращує здатність моделі до міркування, системний промптинг задає контекст і роль, а prompt-tuning націлений на адаптацію моделі до конкретних завдань. Вибір техніки залежить від характеру завдання, доступних ресурсів та рівня точності, якого потрібно досягти. Ці підходи є не лише інструментами, а й основою для подальших досліджень та розвитку методів взаємодії зі AI.

#### **2.4 Структуризація методики порівняльного аналізу великих мовних моделей з опису зображень**

Під час проектування та тестування prompt, зазвичай, взаємодіють з LLM через API [39]. Можна налаштувати кілька параметрів, щоб отримати різні результати для prompt. Налаштування цих параметрів важливе для підвищення надійності та бажаності відповідей, і потрібно трохи поекспериментувати, щоб визначити правильні налаштування для користувацьких випадків використання. Розглянемо загальні налаштування, з якими користувач може зіткнутись під час використання різних постачальників LLM. Чим нижча температура, тим більш детерміновані

результати в тому сенсі, що завжди вибирається найімовірніший наступний токен. Підвищення температури може призвести до більшої випадковості, що заохоче більш різноманітні або творчі виходи. По суті, користувач збільшує вагу інших можливих токенів. З точки зору застосування, можна використовувати нижчий температурний показник для таких завдань, як QA на основі фактів, щоб заохотити більш фактичні та лаконічні відповіді. Можна багато чого досягти за допомогою простих prompts, але якість результатів залежить від того, скільки інформації містить та наскільки добре сформульований prompt. Він може містити певну інформацію та складатись з наступних елементів [40]. Instruction – це конкретне завдання або інструкція, яку необхідно виконати моделі. Context – зовнішня інформація або додатковий контекст, який може спрямувати модель до кращих відповідей. Input Data – вхідні дані або питання, на які користувачу цікаво знайти відповідь. Output Indicator – тип або формат виводу. Можна використовувати ці елементи для більш ефективного інструктування моделі для покращення якості результатів. Слід зазначити, що при використанні моделей чату OpenAI можна структурувати свій запит, використовуючи три різні ролі: system, user, і assistant. System повідомлення не є обов'язковим, але допомагає налаштувати загальну поведінку помічника. Для простоти використовують лише повідомлення user для prompt LLM. Повідомлення assistant в наведеному вище прикладі відповідає моделі відповіді. Також можна визначити повідомлення assistant, щоб передати приклади бажаної поведінки.

Розглянемо приклад prompt «Classify the text into neutral, negative, or positive Text: I think the food was okay. Sentiment». У наведеному вище прикладі інструкції відповідають завдання класифікації: «Класифікуйте текст на нейтральний, негативний або позитивний». Вхідні дані відповідають частині «Я думаю, їжа була нормальною», а індикатор на виході використовується «Настрій:». Зверніть увагу, що в цьому базовому прикладі не використовується контекст, але він також може бути наданий як частина запиту. Наприклад, контекстом для цього текстового запиту класифікації

можуть бути додаткові приклади, надані як частина `prompt`, щоб допомогти моделі краще зрозуміти завдання та керувати типом виходів, який очікує користувач. Коли користувач починає розробляти `prompt`, він повинен мати на увазі, що це дійсно ітеративний процес, який вимагає багато експериментів для отримання оптимальних результатів. Можна почати з простих `prompts` і продовжувати додавати більше елементів і контексту, прагнучи до кращих результатів. З цієї причини повторення `prompt` є життєво важливим. Компонент `Instruction` дозволяє створювати ефективні `prompts` для різних простих завдань, використовуючи команди для вказівки моделі, чого бажано досягти, наприклад, «Записати», «Класифікувати», «Підсумувати», «Перекласти», «Замовити». Чим конкретніший і релевантніший контекст завдання, тим краще. Іноді, рекомендується розміщувати інструкції на початку `prompt`. Іншою рекомендацією є використання якогось чіткого роздільника, наприклад «###», щоб відокремити інструкцію від контексту. Чим більш описовим і детальним буде `prompt`, тим кращими будуть результати. Не існує конкретних токенів, що призводять до кращих результатів. Важливіше мати хороший формат і описовий `prompt`. Надання прикладів у підказці є ефективним для отримання результату в певних форматах. Під час проектування `prompt` також треба пам'ятати про довжину запити. Включення великої кількості непотрібних деталей не завжди є вдалим підходом. Вони повинні бути актуальними і сприяти виконанню поставленого завдання, що досягається проведенням налаштувань. Врахування цих рекомендацій може привести до потенційно неточного опису, тобто треба формулювати конкретно та прямо. Також варто не говорити, що не слід робити, а говорити, що робити замість цього. Це заохочує до більшої конкретики та зосереджується на деталях, які призводять до хороших відгуків моделі. Таким чином, порівняльний аналіз LLM з опису зображень можна виконати за допомогою структурованої методики, що забезпечує об'єктивність та повторюваність результатів. Її структуру можна описати у вигляді кроків, які доцільно визначити наступним чином.

На першому кроці визначаються цілі аналізу, наприклад, Оцінити якість описів, що генерують моделі, для конкретних задач (наприклад, точність опису, релевантність контексту, використання мови), а також критерії порівняння, наприклад, лексичне багатство, синтаксична правильність, адекватність змісту, швидкість генерації.

Другий крок відповідає вибір набору тестових зображень. Достатній для розмір статистичної значущості рекомендований на рівні щонайменше 500 - 1000 зображень. Зображення мають охоплювати широкий спектр контенту (об'єкти, сцени, абстракції, різні умови освітлення тощо). В якості джерел можна використовувати відкриті датасети (OpenImages, COCO Dataset, Flickr8k) або набори даних, що відповідають конкретному завданню користувачу.

На третьому кроці вибираються моделі для порівняльного аналізу. В критеріїв вибору можуть виступати: розмір моделі (велика, середня, компактна), технологічна база (наприклад, GPT, LLaMA, BLOOM тощо), наявність попередньої оптимізації для опису зображень.

Четвертий крок відповідає процесу генерації описів. Здійснюється завантаження зображень у кожну модель. Потім встановлюються однакові параметри (наприклад, максимальна довжина опису, температурний режим, кількість варіантів). Виконуються різні варіанти prompts, спираючись на розглянуті в п. 2.3 техніки. Після цього проводиться збір результатів.

П'ятий крок присвячений оцінці якості описів. Згідно п. 2.1, для іноді застосовується краудсорсинг – залучення великої групи користувачів для суб'єктивної оцінки за шкалою (1 - 5 балів). Але, зазвичай, проводять експертну оцінку. Експерти аналізують опис за кількома параметрами, наприклад, точність (опис відповідає зображенню), чіткість (зрозумілість і простота опису), релевантність (наскільки опис є доречним для задачі). З іншого боку, можна застосовувати автоматизовані метрики. Оскільки кожна метрика має свої обмеження, їхнє комбіноване використання дає більш об'єктивну оцінку. Слід розглядати приклади з низькими значеннями метрик,

щоб зрозуміти, де моделі можуть покращитись. Якщо певні деталі є критично важливими, можливо, варто розглянути додаткові або спеціалізовані метрики. Під час інтерпретації отриманих результатів варто пам'ятати, що високі значення метрик не завжди гарантують високу якість з точки зору людини. Метрики можуть не враховувати різні способи вираження тієї ж самої ідеї. Автоматичні метрики не завжди адекватно оцінюють граматичну правильність та плавність тексту. Можуть пропускати нюанси значення або контексту, важливі для розуміння опису. Тому слід використовувати метрики в поєднанні з людською оцінкою для більш повної картини, а також розглянути можливість використання додаткових метрик, таких як METEOR або SPICE, що враховують семантичну інформацію.

На шостому кроці відбувається порівняння продуктивності. Швидкість генерації описів, обсяг ресурсів (GPU/CPU, RAM) визначають обчислювальна ефективність. Можливість налаштування моделі для нових доменів або мов визначає адаптивність.

На сьомому кроці проводиться візуалізація результатів. Використовуються графіки, таблиці, приклади (показ згенерованих описів поряд із референтними), що містять розподіли метрик, порівняння моделей, середні значення кожної метрики для моделей.

На восьмому кроці формуються висновки та рекомендації. Аналізуються отримані результати із врахуванням мети дослідження. Визначаються найкращі моделі для заданих умов. Обґрунтовуються рекомендації для покращення якості описів (наприклад, донавчання моделей, вибір оптимальних гіперпараметрів).

## **Висновки до розділу 2**

При проведенні порівняльного аналізу LLM для автоматичного опису зображень можна використовувати експертні оцінки на основі MAI. В якості

критерієві можуть виступати точність відповідності контенту зображення, лінгвістична якість тексту, інноваційність і гнучкість моделі. Ключовими обмеженнями методу є значні часові та ресурсні витрати на залучення експертів, а також можливість упередженості оцінок за відсутності чітких критеріїв.

Крім цього для оцінки можна застосовувати автоматизовані метрики. Використання метрик BLEU, ROUGE, CIDEr, SPICE, METEOR, BERTScore та ін. дозволяє оцінити текстові описи з різних аспектів: від точності збігів n-грам до глибокого семантичного аналізу. Це забезпечує багатовимірну оцінку якості моделей. Поєднання таких метрик, як CIDEr + SPICE чи BERTScore + CLIPScore, дозволяє отримати більш збалансовану оцінку, яка враховує як текстову, так і візуальну відповідність описів. Автоматичні метрики значно спрощують процес оцінювання моделей, забезпечуючи об'єктивність і повторюваність результатів. Проте для найбільш повної оцінки рекомендується доповнювати їх експертним аналізом.

У роботі проведено дослідження існуючих методів prompting, що використовуються для підвищення ефективності роботи LLM. Кожен із них демонструє унікальні переваги, недоліки та можливості для вирішення специфічних задач. Найбільш поширеними вважаються zero-shot, few-shot, chain-of-thought.

Попри численні переваги, методи prompting часто залежать від якості вихідних даних, складності завдань і можливостей моделей. Точний вибір техніки потребує врахування цих факторів.

У роботі запропоновано методика порівняльного аналізу LLM для генерації описів зображень, що забезпечує об'єктивність, повторюваність результатів та оптимізацію процесу оцінювання. Вона передбачає: визначання цілі аналізу, вибір набору тестових зображень, вибір моделі для порівняльного аналізу, генерацію описів, оцінку якості описів, візуалізацію результатів та формування висновків та рекомендацій.

## РОЗДІЛ 3

### РЕКОМЕНДАЦІЇ ЩОДО ВИКОРИСТАННЯ МЕТОДИКИ ПОРІВНЯЛЬНОГО АНАЛІЗУ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ З ОПИСУ ЗОБРАЖЕНЬ

#### 3.1 Локальний OCR-додаток Ollama на базі Llama 3.2-Vision

Згідно [41], оптичне розпізнавання символів (Optical Character Recognition, OCR) стало важливим інструментом для оцифровки друкованого тексту та вилучення інформації із зображень. Завдяки прогресу в галузі AI такі моделі, як Llama 3.2-Vision, надають потужні можливості створення власної програми OCR, використовуючи Python як мову програмування. В роботі розглядається Python версії 3.7 або вище. Ollama – це платформа, яка дозволяє запускати мультимодальні моделі локально. Щоб встановити Ollama завантажуюмо інсталяційний пакет, що підходить для відповідно ОС [42]. Після того, як встановлена Ollama, можна встановити модель Llama 3.2-Vision, виконавши наступну команду у своєму терміналі: `ollama run llama3.2-vision`. Ця команда завантажує та налаштовує модель для локального використання. Далі налаштовуємо середовище Python для проекту OCR. Для цього створюється новий каталог для проекту: `mkdir llama-ocr && cd llama-ocr`. Потім створюється віртуальне середовище (необов'язково, але рекомендується):

```
python -m venv venv
source venv/bin/activate # On Windows use `venv\\Scripts\\activate`
```

Встановлюються необхідні бібліотеки, серед яких є бібліотека для обробки зображень та кодування – `base64`. Вона встановлюється за допомогою пакету `pip`: `pip install requests Pillow`. Далі пишемо свій сценарій OCR на основі скрипта на Python, який виконуватиме OCR за допомогою Llama 3.2-Vision. Створюємо новий файл з іменем і додаємо наступний код: `ollama_ocr.py`. Після цього формуємо `prompt`: «Виконуйте

функції помічника OCR. Проаналізуйте подане зображення та: 1. Якомога точніше розпізнайте весь видимий текст на зображенні. 2. Зберігайте оригінальну структуру та форматування тексту. 3. Якщо якісь слова чи фрази незрозумілі, позначте це «[незрозуміло]» у вашій транскрипції. Надати лише транскрипцію без додаткових коментарів». Функція `encode_image_to_base64` читає файл зображення та перетворює його на рядок base64, який потрібен для надсилання зображень через запити HTTP. Функція `perform_ocr` надсилає запит POST до локальної служби Ollama із системним запитом і зображенням у кодуванні base64. Сценарій перевіряє, чи запит був успішним, і отримує розпізнаний текст із відповіді JSON. Розглянутий програмний код наведений у Додатку А. Щоб запустити свій сценарій OCR, треба замінити «`path/to/your/image.jpg`» на фактичний шлях до файлу зображення, який необхідно проаналізувати. Потім виконується сценарій у терміналі: `python ollama_ocr.py`. Користувач повинен побачити результат, подібний до цього:

```
OCR Recognition Result:  
The text recognized from your image will be displayed here.
```

Якщо буде виявлено, що результати OCR незадовільні, то необхідно продумати налаштування змінної `SYSTEM_PROMPT` у сценарії, щоб він краще відповідав конкретному випадку використання або покращити ясність `prompt`, наданих для Llama 3.2-Vision.

Створення OCR-додатку за допомогою Llama 3.2-Vision за допомогою Ollama є простим і потужним завдяки його мультимодальним можливостям.

### **3.2 Порівняльний аналіз мовних моделей з опису зображень**

В рамках проведених досліджень виконаний порівняльний аналіз мультимодальних мовних моделей, що дозволяють генерувати опис зображень. Для прикладу розглядалися GPT-4o від к. OpenAI [7] та Llama 3.2 від к. Meta AI [43]. Llama 3.2 – перші мультимодальні моделі Meta AI, і тести

свідчать про те, що вони є сильними конкурентами малих і середніх пропріетарних альтернатив. При цьому розглядалось кілька тестових завдань.

Загальне розуміння образів, підрахунок та ідентифікація предметів. Для початку візьмемо образ Фрірена, що їсть бургер (рис. 3.1) [44] та будемо порівнювати відповіді GPT-4o і Llama 3.2. Був сформований prompt «Explain the image» (поясніть зображення).



Рисунок 3.1 – Порівняння відповідей GPT4o та Llama 3.2

Проаналізуємо отримані відповіді. Відповідь GPT4o: «Зображення є ілюстрацією в стилі аніме персонажа зі світло-фіолетовим волоссям, довгими кісками та ельфійськими вухами, який насолоджується їжею в ресторані швидкого харчування. Людина тримає великий гамбургер і виглядає задоволеною із закритими очима та невеликим рум'янцем на щоках. Стіл перед ними заставлений продуктами з McDonald's, включаючи кілька пачок картоплі фрі, два бургери та два напої. Здається, що обстановка знаходиться всередині ресторану, на задньому плані видно столи та стільці. Загальний стиль милий і химерний». Відповідь Llama3.2: «На зображенні зображено мультяшну ілюстрацію персонажа аніме, який насолоджується їжею в ресторані швидкого харчування. Персонаж, аніме-дівчина з довгим білим волоссям і загостреними вухами, одягнена в біло-золотий одяг і сидить за столом із підносом із їжею перед нею. Вона тримає в руках великий бургер і, здається, відкушує. Піднос містить різноманітні продукти швидкого

харчування, зокрема картоплю фрі, напої та салат. На задньому плані зображено інтер'єр ресторану швидкого харчування, де видно столи та стільці. Загальна атмосфера зображення – це невимушена трапеза та відпочинок». Обидві відповіді були однаково хорошими, але GPT4o зміг правильно відтворити логотип McDonald's.

Далі проаналізуємо правильність підрахунку об'єктів на зображенні. Почнемо з простого зображення (рис. 3.2) [45]. Задаємо prompt «Sort the number of balls as per their colour» (розсортуйте кількість кульок за кольором). Обидві моделі змогли відповісти на нього правильно.

Надалі дещо ускладнимо завдання. Зробимо запит порахувати кількість виделок на зображенні (рис. 3.3) [46], тобто створимо prompt «Can you count the number of forks in the image?». В даному випадку, Llama 3.2 змогла відповісти на цей запит правильно, тоді як GPT4o не помітив виделку на столі, яку не було видно відразу.



Рисунок 3.2 – Зображення на підрахунок кількості предметів



Рисунок 3.3 – Ускладнене завдання на підрахунок кількості предметів

Надалі зробимо запит – порахувати бокали та пояснити їх форму (prompt «How many glasses are there? Explain the shapes of the glasses.»). Обидві моделі правильно визначили кількість. Опис GPT4o був набагато кращим, а форми бокалів були правильно пояснені. З іншого боку, опис Llama 3.2 був частково правильним.

Одним з поширених випадків використання мовних моделей є ідентифікація будь-якого інструменту на полиці та прохання пояснити його функцію, наприклад, рис. 3.4 [47]. Для цього створимо prompt «What tool is it, and how can it be used?». Обидві впоралися добре, але модель GPT-4o була більш детальною та інформативною.

Надалі спробуємо виконати діагностику хвороб листя. Зробимо запит визначити хвороби рослин за фотографією (рис. 3.5) [48] – prompt «Can you explain the leaf disease in the image?». Обидві моделі правильно впоралися з хворобою, яку визначали як несправжня борошниста роса.



Рисунок 3.4 – Ідентифікація інструменту



Рисунок 3.5 – Зображення хворого листя

В роботі розглядалась ще одна ситуація – розуміння лікарських рецептів (рис. 3.6) [49]. GPT 4o була набагато кращою. Вона розуміла ім'я пацієнта і призначені ліки. Llama 3.2 не доклала жодних зусиль.

**Name:** Lolita Alvarez  
**Address:** Bagong Bag, Pasig City  
**Age:** 39 **Sex:** F **Date:** 2-5-13

**Rx**  
 FeSO<sub>4</sub> tab # 30  
 Sig: A.D.

---

Ascorbic Acid # 30  
 500mg tab  
 Sig: once a day

Physician's Sig: J. Dela Cruz  
 Lic. No. 12345  
 PTR No. 1234567  
 S2 No.

**GPT4o response**

The prescription is for a patient named Lolita Alvarez, who is 39 years old and female. The prescription is dated February 5, 2013. The doctor's name appears to be J. Dela Cruz, with license and PTR numbers given.

The prescription includes two medications:

1. FeSO<sub>4</sub> tab (Ferrous Sulfate) 30 tablets  
 - Sig: A.D. (Latin for "auri dextra", meaning "right ear" or could be "as directed" depending on context)
2. Ascorbic Acid 500 mg tab (Vitamin C) 30 tablets  
 - Sig: Once a day

The instructions for the Ferrous Sulfate are unclear ("A.D." is not a standard dosing instruction). It is advisable to confirm the dosing instructions with the prescribing doctor or pharmacist for the correct administration directions.

**Llama response**

The prescription is written in a foreign language, but it appears to be a prescription for a medication. The prescription includes the patient's name, address, and date of birth, as well as the name of the medication, the dosage, and the frequency of administration. It also includes the signature of the prescribing physician.

Рисунок 3.6 – Обробка лікарського рецепту

Крім розглянутих завдань розглядалися ще декілька. Одне з них є виділення важливих текстів із зображень, табличних даних у визначеному форматі, опис та виділення практичних деталей із рахунків-фактур та ін. В даному випадку, Llama 3.2 – показала себе як потужна модель. Технічно, вона може витягувати тексти із зображень. Однак, GPT-4o краще дотримується prompt (отже, легше витягнути необхідні дані за її допомогою). При формуванні резюме аналізу фінансових графіків GPT-4o все ж таки краще підходить для складних графіків та аналізу. Llama 3.2 галюцинує більше, ніж GPT4o, і вигадує щось самостійно. В цілому, модель Llama 3.2 Vision гарно розуміє та аналізує загальні образи. Враховуючи переваги вартості та конфіденційності, має сенс використовувати її локально в завданнях, що не вимагають складного аналізу або глибоких знань. Обидві моделі добре показують себе в загальному розумінні образу. У той час як GPT-4o залишається кращою моделлю, Llama 3.2 краща, якщо врахувати співвідношення вартості та корисності. Розуміння лікарського призначення та звіту – GPT4o все ж таки кращий з точки зору аналізу медичного звіту.

### 3.3 Економічне обґрунтування прийнятих рішень

Фінансовий аналіз включає оцінку прямих та непрямих витрат на впровадження локалізованих LLM. Прямі витрати включають вартість ліцензій на програмне забезпечення, придбання апаратного забезпечення та оплату послуг спеціалістів. Непрямі витрати включають навчання персоналу та технічну підтримку. Очікувані фінансові вигоди включають зниження витрат на обслуговування, підвищення продуктивності та зростання доходів за рахунок покращення якості обслуговування клієнтів. Наприклад, при використанні LM Studio та open source LLM рівня GPT4, в тому числі з візуальним асистентом, не має потреби сплачувати абонементу ChatGPT4о к. OpenAI ( $C_1 \approx 850$  грн/місяць), що складає на рік:

$$V = C_1 \cdot 12 = 10200 \text{ грн.} \quad (3.1)$$

Ідентифікація та управління ризиками є важливим етапом процесу впровадження локалізованих LLM [50]. Основні ризики включають технічні ризики, пов'язані з інтеграцією моделей, фінансові ризики, пов'язані з перевищенням бюджету, ризики людського фактору, такі як недостатня кваліфікація персоналу. Стратегії управління ризиками включають регулярний моніторинг проекту, створення резервних планів та проведення навчальних програм для співробітників. План впровадження включає наступні етапи:

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

Запропонований в роботі підхід до локалізації LLM дозволяє забезпечити: конфіденційність даних; низьку затримку та високу доступність;

налаштування та адаптація під конкретні завдання; економічну ефективність; незалежність від зовнішніх послуг; відповідність законодавству та нормативним вимогам; підтримку локальних мов та особливостей. Такий підхід є привабливим для організацій, які прагнуть досягти максимальної ефективності та надійності своїх IT-рішень. Локалізація LLM є доцільним та економічно обґрунтованим рішенням для організацій, які прагнуть підвищити ефективність та надійність своїх IT-рішень. Впровадження локалізованих LLM дозволить покращити взаємодію з користувачами, знизити витрати на обслуговування та підвищити продуктивність.

### **Висновки до розділу 3**

Розробка локального OCR-додатку на основі моделі Llama 3.2-Vision демонструє значні переваги у завданнях OCR. Використання інструкцій з налаштування та готових команд значно спрощує створення функціонального додатку. Це дозволяє підвищити точність розпізнавання у складних або нестандартних умовах.

Функціональність сценарію може бути адаптована до конкретних випадків використання шляхом зміни системного промпту або налаштувань моделі. Порівняльний аналіз моделей GPT-4o і Llama 3.2 показав, що обидві здатні успішно виконувати широкий спектр завдань з аналізу зображень. Вони демонструють високий рівень розуміння контексту зображень, однак GPT-4o є більш точною та інформативною, особливо для завдань, що вимагають детального опису або аналізу. Llama 3.2 має меншу схильність до точного опису деталей, але показує добру продуктивність у загальних завданнях. Натомість GPT-4o краще підходить для складного аналізу, завдяки кращій адаптації до системних запитів. Враховуючи переваги вартості та конфіденційності, має сенс використовувати Llama 3.2 в завданнях, які не вимагають складного аналізу або глибоких знань.

Впровадження локалізованих мовних моделей дозволяє значно скоротити витрати на використання хмарних рішень, таких як GPT-4o. Економія від зменшення ліцензійних витрат і абонплат сприяє рентабельності проекту, що підтверджується розрахунками щорічної економії.

## ВИСНОВКИ

Візуальні асистенти мовних моделей є ключовим напрямом розвитку сучасного AI, який інтегрує NLP та CV. Вони дозволяють суттєво розширити функціональність LLM, забезпечуючи аналіз візуального контексту, розпізнавання об'єктів та генерацію релевантних текстів. Інноваційні розробки, такі як GPT-4V та VisionLLM, демонструють значний потенціал для використання в медицині, освіті, промисловості та інших галузях. Проте зберігаються обмеження, зокрема галюцинації, неточності інтерпретації та недостатня прозорість роботи моделей, які потребують подальшого вдосконалення. Інтеграція моделей LLM та CV дозволяє досягти нового рівня мультимодального аналізу, об'єднуючи текстові та візуальні дані. Нові методи навчання та системи (CLIP, MiniGPT-v2 та LENS) сприяють точнішому розумінню візуальних концепцій та текстових інструкцій. Однак, для критичних сфер важливо усунути проблеми з прозорістю алгоритмів та підвищити їхню точність.

Дослідження підходів щодо оцінки LLM для опису зображень підкреслило важливість поєднання експертних оцінок і автоматизованих метрик. Експертний підхід забезпечує глибокий контекстуальний аналіз текстів, враховуючи точність відповідності контенту зображення, лінгвістичну якість, інноваційність та гнучкість моделей. Водночас автоматизовані метрики, такі як BLEU, ROUGE, CIDEr, SPICE, METEOR і BERTScore, дозволяють отримати об'єктивні та повторювані результати, оцінюючи текстові описи з різних аспектів. Оптимальним рішенням є поєднання обох підходів, що сприяє комплексному аналізу якості моделей.

Методи prompting демонструють значний потенціал у підвищенні ефективності LLM, однак їхня успішність залежить від якості вихідних даних, складності задач і специфіки моделей. Точний вибір техніки потребує врахування цих факторів для досягнення найкращих результатів. Розроблена методика порівняльного аналізу LLM для генерації описів зображень

забезпечує системний і послідовний підхід до оцінювання. Вона охоплює всі ключові етапи, включаючи визначення цілі аналізу, вибір тестового набору зображень, генерацію описів, оцінювання якості текстів, візуалізацію результатів та формулювання висновків і рекомендацій. Такий підхід дозволяє оптимізувати процес оцінювання, підвищуючи його об'єктивність і ефективність. Загалом, впровадження комплексних методик оцінювання та використання вдосконалених технік prompting сприяє підвищенню точності й гнучкості мовних моделей, що відкриває нові можливості для їхнього застосування в різноманітних галузях науки, бізнесу й технологій.

Розробка локалізованих моделей LLM відкриває перспективи економії ресурсів та підвищення конфіденційності даних. Наприклад, використання локального OCR-додатку на базі моделі Llama 3.2-Vision демонструє високий рівень адаптивності та точності в завданнях OCR. Порівняльний аналіз моделей GPT-4o та Llama 3.2 підтверджує доцільність використання локалізованих рішень у завданнях із помірною складністю.

GPT-4o краще справляється з ідентифікацією специфічних об'єктів, таких як логотипи чи елементи на малопомітних ділянках зображення, таких як підрахунок об'єктів чи базовий опис. Llama 3.2 показала себе як модель, придатна для виконання нескладних задач локально, що робить її ефективним вибором у контексті збереження конфіденційності даних. Модель Llama 3.2 частіше галюцинує, ніж GPT-4o, що свідчить про потребу у вдосконаленні її алгоритмів. Однак, враховуючи її нижчу вартість та можливість локального використання, вона є хорошим варіантом для завдань із помірною складністю. GPT-4o демонструє переваги у розумінні складних медичних текстів та фінансових даних, що робить її придатною для професійного використання. Llama 3.2 є конкурентоспроможною альтернативою в задачах, де критичним є збереження приватності та економія ресурсів. Хоча GPT-4o перевершує Llama 3.2 за якістю аналізу, остання має значну перевагу у співвідношенні вартості та ефективності. Це робить її кращим вибором для загальних або обмежених за ресурсами завдань.

Підчас проведення порівняльного аналізу доцільно використовувати різнопланові тестові завдання, наприклад: базове розуміння образу, аналіз лікарських призначень та звітів, виділення тексту з зображень, тлумачення фінансового графіка. Подальші дослідження та вдосконалення мультимодальних мовних моделей сприятимуть їх впровадженню в наукові, технічні та бізнес-процеси. Зосередженість на оптимізації для пристроїв з обмеженими ресурсами, етичних стандартах, прозорості алгоритмів та адаптації до потреб користувачів є ключовими факторами для успішного розвитку цієї технології.

Таким чином, результатами роботи є методика порівняльного аналізу великих мовних моделей з опису зображень; рекомендації щодо використання великих мовних моделей для опису зображень. Вони можуть бути використані для подальших досліджень за даною тематикою та при проектуванні хмарних сервісів.