

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

**Кваліфікаційна робота
на здобуття ступеня магістра**

за спеціальністю 121 Інженерія програмного забезпечення

на тему:

**РОЗРОБКА ЗАСТОСУНКУ ДЛЯ СТВОРЕННЯ ІДЕЙ
В ТЕХНІЧНИХ ГАЛУЗЯХ З ВИКОРИСТАННЯМ
ЗАСОБІВ ШТУЧНОГО ІНТЕЛЕКТУ**

Виконав студент 2-го курсу магістратури
Гліб ПИЛИПЕЦЬ

(підпис)

Науковий керівник:
асистент, кандидат фіз.-мат наук
Костянтин ЖЕРЕБ

(підпис)

Засвідчую, що в цій роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент

(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри інтелектуальних
програмних систем

« ____ » _____ 2023 р.,

протокол № ____

Завідувач кафедри

Олександр ПРОВОТАР

(підпис)

РЕФЕРАТ

Обсяг роботи: 50 сторінок, 11 ілюстрацій, 57 джерел посилань.

ГЕНЕРАЦІЯ ІДЕЙ, ІННОВАЦІЇ, ШТУЧНИЙ ІНТЕЛЕКТ, ОБРОБКА ПРИРОДНОГО МОВЛЕННЯ, OPENAI, ВЕБ-ПРОГРАМУВАННЯ, NLP, AI

В роботі було проведено дослідження з аналізом різних методів генерації ідей з використанням штучного інтелекту. Даний аналіз було прийнято до уваги при реалізації технічного-орієнтованого веб-застосунку, що надає готові пропозиції ідей, а також релевантну інформацію для створення ідей.

Об'єктом дослідження є процес створення ідей в технічних галузях.

Предметом дослідження є аналіз можливості створення ідей за допомогою методів штучного інтелекту.

Мета роботи: проектування та імплементація застосунку для створення та обробки ідей методами штучного інтелекту, а також аналізу наукових досліджень, новин, тенденцій та інших джерел інформації.

Методи розроблення: глибинне навчання, архітектури трансформерів GPT-2 і text-davinci-003 від OpenAI для генерації ідей, веб-програмування, API-запити до різних джерел інформації.

Інструменти розроблення: мова програмування Python, JavaScript-фреймворк Vue.JS.

Результат роботи: розглянуто підходи по створенню ідей, визначено основні інформаційні ресурси та потоки для створення ідей, проаналізовані існуючі додатки, що генерують ідеї, спроектований і реалізований варіант додатку для створення ідей з використанням OpenAI API, hugging face inference моделей, arxiv, patents_view, arxivexplorer, Google News/Trends та інших NLP підходів.

ЗМІСТ

| | |
|---|----|
| СКРОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ | 4 |
| ВСТУП | 5 |
| РОЗДІЛ 1. КОНЦЕПТИ КРЕАТИВНОСТІ ТА ІДЕЙ | 8 |
| 1.1. Визначення креативності | 9 |
| 1.2. Ролі і побудова | 10 |
| 1.2.1. Креативність у суспільстві | 10 |
| 1.2.2. Еволюція і побудова | 11 |
| РОЗДІЛ 2. ДОСЛІДЖЕННЯ СПОСОБІВ ГЕНЕРАЦІЇ ІДЕЙ | 13 |
| 2.1. Варіаційні автокодери | 13 |
| 2.2. Трансформерні моделі | 15 |
| 2.2.1. Основні концепти | 15 |
| 2.2.2. Приклади моделей | 16 |
| 2.2.3. Застосування | 16 |
| 2.3. Методи на основі вхідних даних | 17 |
| 2.3.1. Основні концепти | 17 |
| 2.3.2. Застосування | 18 |
| 2.4. NLP inference | 18 |
| 2.5. Існуючі реалізації генерації ідей | 21 |
| 2.6. Основні недоліки існуючих реалізацій | 27 |
| 2.7. Методологія розробки застосунку для генерації ідей | 27 |
| РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ ДЛЯ СТВОРЕННЯ ІДЕЙ | 35 |
| 3.1. Макет застосунку | 35 |
| 3.2. Архітектура системи | 36 |
| 3.3. Імплементация основної частини додатку | 36 |
| 3.4. Інфраструктура додатку (фронтенд та бекенд) | 39 |
| РОЗДІЛ 4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ | 41 |
| 4.1. Робота системи для створення ідей | 41 |
| 4.2. Можливі покращення системи | 43 |
| ВИСНОВКИ | 45 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ | 46 |

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

| | |
|------|---|
| ГН | Глибинне навчання |
| ШІ | Штучний інтелект |
| API | Application programming interface |
| BERT | Bidirectional encoder representations from transformers |
| CNN | Convolutional neural networks |
| GAN | Generative adversarial network |
| GPT | Generative pre-trained transformer |
| GRU | Gated recurrent unit |
| JSON | JavaScript object notation |
| LSTM | Long short-term memory |
| MNLI | MultiNLI |
| NLI | Natural language inference |
| NLP | Natural language processing |
| OUP | Oxford university press |
| RNN | Recurrent neural networks |
| SPA | Single-page application |
| SNLI | Stanford natural language inference |
| VAE | Variational autoencoder |

ВСТУП

Оцінка сучасного стану об'єкта дослідження. За останні роки технології штучного інтелекту зазнали значного прогресу, особливо в галузі ГН та нейромереж [1]. Багато компаній та організацій використовують ці технології для розробки різноманітних програм та інструментів, які забезпечують автоматизацію процесів та поліпшення продуктивності. У галузі створення ідей застосування штучного інтелекту все ще вважається відносно новим напрямком, однак існують деякі застосування, які знаходяться в розробці або вже використовуються [2]. Такі інструменти, як генеративні моделі та машинне навчання, можуть бути застосовані для автоматичної генерації ідей та пропозицій у різних сферах, наприклад, в маркетингу, рекламі, дизайні, наукових дослідженнях та інших. Однак, існують також деякі виклики, пов'язані з застосуванням штучного інтелекту в цій галузі, такі як проблема інноваційності та творчості, а також етичні питання, пов'язані зі збереженням приватності та конфіденційності [3].

Актуальність роботи та підстави для її виконання. Розвиток технологій штучного інтелекту дозволяє застосовувати його в різних галузях, включаючи креативність і творчість. Застосування штучного інтелекту для створення ідей є новим напрямком досліджень, що має великий потенціал в різних галузях, таких як маркетинг, реклама, дизайн, музика, кіно і т.д. Підставою для виконання даної магістерської роботи є потреба розробити новий технічно-орієнтований застосунок для створення ідей з використанням штучного інтелекту, який міг би допомогти вирішувати складні творчі завдання. В рамках роботи були проаналізовані існуючі методи та алгоритми штучного інтелекту для створення ідей, їх переваги та недоліки, а також розроблений новий технічно-орієнтований застосунок, який може бути використаний в різних галузях.

Мета й завдання роботи. Метою магістерської роботи є проектування та імплементація застосунку для створення та обробки ідей методами штучного

інтелекту, а також аналізу наукових досліджень, новин, тенденцій та інших джерел інформації. Для її досягнення були поставлені наступні завдання, як:

1. Визначення можливості застосування методів штучного інтелекту для створення ідей.
2. Вивчення наукових праць та розробок у галузі штучного інтелекту та створення ідей.
3. Проектування методології розробки технічного застосунку для створення ідей з використанням штучного інтелекту.
4. Розроблення архітектури технічного застосунку та його реалізація.
5. Аналіз результатів використання застосунку та розроблення рекомендації щодо його вдосконалення.

Результати подібної роботи можуть мати практичне значення для бізнесу та науково-дослідних організацій, які займаються розробкою нових ідей та інноваційних продуктів.

Об'єкт, методи й засоби розроблення. Об'єктом пізнання є процес створення ідей в технічних галузях. Для виконання дослідження були обрані математичні та програмні засоби розробки. В якості математичних засобів були застосовані архітектури моделей ГН для генерації ідей від OpenAI, такі як GPT-2 і text-davinci-003. В якості програмних засобів були використані фреймворк Flask для забезпечення API доступу до різних ресурсів (патентів, новин, трендів, публікацій, AI моделей), а також Vue.JS для побудови веб-застосунку.

Можливі сфери застосування. Можливими сферами застосування результатів магістерської роботи є:

1. Інноваційна діяльність в галузі науки, техніки та технологій – розробка нових ідей та продуктів з використанням методів штучного інтелекту.
2. Маркетинг та реклама – створення рекламних кампаній та ідей з використанням штучного інтелекту, які допоможуть залучити більше клієнтів.
3. Індустрія креативу – розробка креативних ідей для мистецьких проєктів, рекламних кампаній, фільмів, відеоігор тощо.

4. Освіта – застосування штучного інтелекту для створення ідей у навчальному процесі та розробка нових методик навчання та тестування.

5. Медіа-індустрія – розробка нових форматів контенту з використанням методів штучного інтелекту.

6. Індустрія туризму – розробка нових туристичних ідей та продуктів з використанням штучного інтелекту, які допоможуть привернути більше туристів.

7. Бізнес-консалтинг – використання штучного інтелекту для створення ідей та розробка стратегій розвитку бізнесу.

РОЗДІЛ 1. КОНЦЕПТИ КРЕАТИВНОСТІ ТА ІДЕЙ

Зв'язок між креативністю та машинами такий же давній, як і сама комп'ютерна наука. У 1842 році леді Лавлейс, англійський математик і письменниця, яку багато хто визнає першим програмістом, висунула так зване «заперечення Лавлейс» [4]: Аналітичний двигун (цифрова програмована машина, запропонована Чарльзом Беббіджем [5]) «не претендує на те, щоб щось породжувати, оскільки він може робити лише те, що ми знаємо, як йому наказати» [6]. Дійсно, в наступні століття було здійснено кілька проектів і досліджень, пов'язаних з розробкою машин, здатних «створювати щось» [7-12]. Це призвело до появи спеціалізованої галузі в комп'ютерних науках, а саме обчислювальної креативності [13], яка займається вивченням взаємозв'язку між креативністю та штучними системами [14-15].

У цьому контексті застосування методів ГН призвело до значного прориву в останні роки. Величезна обчислювальна потужність і дуже великі обсяги доступних даних лежать в основі зростаючого успіху глибинних генеративних моделей (тобто генеративних моделей, заснованих на ГН [16]). Дійсно, генеративні технології ГН використовуються для написання газетних статей, генерування людських облич [17] і голосів [18], проектування ліків і білків [19] і навіть для створення творів мистецтва, які продаються за кілька сотень тисяч доларів. Однак, хоча очевидно, що сучасні технології здатні генерувати вражаючі результати, водночас можна стверджувати, що їх не можна вважати креативними в цілому [20]. Насправді, метою генеративного ГН є отримання синтетичних даних, які якомога ближче нагадують реальні дані, подані на вході [16]. З іншого боку, креативність передбачає новизну та різноманітність [21]. Хоча для деяких завдань може бути достатньо простого генерування контенту [22], для інших завдань, таких як, наприклад, у мистецтві, важлива здатність створювати різні (але все одно цінні) результати.

Мета цього огляду – представити та критично обговорити сучасний стан генеративного ГН з точки зору машинної креативності.

1.1. Визначення креативності

Креативність вивчається десятиліттями, але досі немає єдиної думки щодо її визначення. Наразі існує понад сто визначень [23-24], і їхня кількість продовжує зростати. Іншими словами, можна сказати, що креативність – це багатозначне слово, тобто люди вкладають у нього безліч значень [25]. Тим не менш, деякі концепції на сьогоднішній день є загальноприйнятими. Однією з них є можливість вивчення креативності з чотирьох різних точок зору: людина, місце, процес і продукт. Вони вивчалися і в комп'ютерній креативності [26]. Однак традиційно основна увага приділяється виміру продукту. Дійсно, ідея полягає в тому, що ми вивчаємо креативність, не беручи до уваги внутрішню сутність творця (людини), зв'язок із середовищем (пресою) або процес. Навіть якщо людина, місце і процес є важливими, ми зосереджуємося на аспектах творчої роботи у зв'язку з самим результатом (продуктом).

З цієї причини можна розглянути три критерії Бодена для вивчення машинної креативності, яка визначається як «здатність генерувати ідеї або артефакти, які є новими, дивовижними та цінними» [27]. Цінність і новизна завжди розглядаються як критерії креативності продукту; також часто використовується поняття «несподіванка» [28]. Ці поняття можна визначити по-різному. Даний аналіз спирається на три критерії Бодена, оскільки вони отримали широке розповсюдження. Боден також пропонує виокремити три форми креативності [27], щоб описати, як створюється новий і несподіваний продукт, пов'язуючи таким чином процес і продукт. Три форми креативності – це комбінаторна, дослідницька та трансформаційна. Комбінаторна креативність полягає у створенні незнайомих комбінацій зі знайомих ідей. Дослідницька креативність передбачає вивчення концептуального простору, визначеного відповідним культурним контекстом.

Трансформаційна креативність передбачає зміну цього простору таким чином що уможлиблює появу нових, раніше немислимих думок.

1.2. Ролі і побудова

Основне питання полягає в тому, яку роль відіграє креативність у певному суспільстві і чому вона може бути такою важливою. Це окреслює сприйманий вплив креативності, а також те, які аспекти цінуються у креативності. Як зазначає Гарднер [29], креативність можна розглядати як тимчасову невідповідність між ідеєю та суспільством, в якому вона розвивається. Зрештою, оскільки деякі люди готові йти на більший ризик і сприймати нові ідеї, вони поступово інтегруються в соціальну структуру і, зрештою, стають прийнятною і стандартизованою частиною суспільства.

1.2.1. Креативність у суспільстві

Моран [30] запропонував дослідити функціональні аспекти ролі, яку креативність може відігравати в суспільстві, припускаючи, що вона здебільшого виконує функції вдосконалення або самовираження. Роль удосконалення – це вплив, який творчий об'єкт може мати на суспільство (як технологічний артефакт), тоді як роль самовираження зосереджена на ролі, яку креативність відіграє в житті людини. Хоча ці дві ролі здаються дещо дихотомічними, їх пропонується розглядати як взаємодоповнюючі, а не конкурентні.

Бачення креативності як вдосконалення – це те, що часто прославляється політичними чи промисловими лідерами, як обов'язковий інструмент для прогресу суспільства і людства. У цьому сенсі суспільство розглядається як система, яка постійно рухається вгору до гіпотетичного благословенного стану (незважаючи на те, що ми не маємо ані найменшого уявлення про те, де це може бути). Таким чином, креативність є інструментом, який транспортує суспільство через кордони сьогодення до ідеалізованого майбутнього, переміщуючи норми на більш високу позицію [31]. У той час як практики розробляють норми, креативність тягне суспільство вперед, неминуче поступаючись місцем стандартизації [30]. У цьому

погляді на креативність як на підсилювача суспільства, її важливість полягає в тому, що вона є метою і що вона дозволяє нам рухатися до неї.

Інша роль креативності, запропонована Мораном [30] – це роль самовираження, яку можна інтерпретувати по-різному залежно від суспільства, в якому вона розгортається. У цьому аспекті креативність можна розглядати як засіб самовираження та індивідуальності, тоді як точний характер цього самовираження залежить від вседозволеності навколишнього суспільства. Такі форми креативності в суспільстві менш зосереджені на результатах, а радше допускають соліпсичні та індивідуалістичні підходи. Таким чином, креативність як самовираження може розглядатися як катарсична діяльність, яка може менше цінуватися суспільством.

Обидві ролі креативності в суспільстві можуть взаємодіяти і призводити, наприклад, до покращення в результаті самовираження. У будь-якому випадку, важливо зазначити, що креативність – це, перш за все, трансформаційна діяльність. Існує загально визнаний консенсус, що креативність можлива лише на основі наявних знань, які суттєво впливають на характер і якість творчих результатів. Як зазначалося раніше, невизначеність креативності не є безмежною, оскільки ніщо не виникає з концептуальної порожнечі. Коли Ньютон сказав: «Якщо я бачив далі, то тільки стоячи на плечах велетнів», він визнав, що його власні відкриття стали можливими лише завдяки всім знанням, накопиченим попередніми дослідниками до нього. Отже, креативність – це результат накопиченого спільнотою досвіду, який згодом трансформується в інкрементальний спосіб.

1.2.2. Еволюція і побудова

Цей трансформаційний погляд втілюється в еволюційній теорії креативності, яка пропонує цікаву паралель між генетичною еволюцією та розвитком творчих ідей. Оригінальна дарвінівська модель Саймонтона [32] мала на меті описати більш розвивальний аспект творчого процесу, але вона природно поширюється на більші (суспільні) масштаби того, як креативність розвивалася в часі і як соціальні фактори можуть вступати в гру. Дійсно, креативні ідеї будуються на попередніх в адаптивний і відкритий спосіб [33]. Цей погляд на творчі ідеї, що розвиваються в часі через культуру, можна зобразити в дарвінівських термінах [34], як на форму

успадкування ідей, які поступово пристосовуються до тимчасових обмежень їхнього соціального середовища, коли вони переходять від однієї людини до іншої. Ця концепція називається теорією подвійного успадкування [33], яка підкреслює той факт, що ми успадковуємо як біологічну, так і культурну інформацію. Ця теорія розглядає культуру як дискретні елементи, які піддаються процесу адаптації, що складається як з випадкових мутацій, так і з випадкового процесу, коли ідеї відбираються під впливом зовнішнього середовища [35].

У цьому сенсі досліджується невідомий простір ідей (варіація) і обирається одна з них, а інші – ні (відбір), перетворюючи креативність на алгоритм відбору варіацій, що ґрунтується на досвіді [36]. Тут соціальний контекст становить найбільший інтерес, оскільки креативні ідеї розглядаються однаково як за їхнім змістом, так і за соціальними рамками, в яких вони продукуються. Оскільки ідеї стають актуальними лише тоді, коли навколо подібних думок формулюється критична маса знань, це ще більше підкреслює повсюдну потребу в знаннях, які трансформуються через творчі процеси. Однак це ускладнює завдання чіткого розмежування інтелекту і креативності. Дійсно, інтелект можна описати як процес асоціювання та трансформації існуючих знань. Проте, оскільки соціальні науковці понад століття дискутували щодо розмежування цих двох типів поведінки, це свідчить про існування аналогічної потреби у зміщенні наших міркувань від штучного інтелекту до штучної креативності.

РОЗДІЛ 2. ДОСЛІДЖЕННЯ СПОСОБІВ ГЕНЕРАЦІЇ ІДЕЙ

Ключовими методами генерації ідей [2] є:

1. Нейронне перенесення стилю – метод, який дозволяє переносити стиль одного зображення на інше, в результаті чого створюється нове зображення, яке поєднує зміст одного зображення зі стилем іншого. Існує кілька варіацій цього методу, включаючи використання глибинних нейронних мереж та використання змагальних мереж для покращення якості отриманих результатів.

2. Генерація тексту. Є кілька методів генерації тексту, включаючи мовні моделі, такі як GPT-3, і моделі нейронного машинного перекладу. Можна також використовувати навчання з підкріпленням для покращення якості згенерованого тексту, а також механізми уваги для покращення зв'язності згенерованого тексту.

3. Генерація музики. Є кілька методів генерації музики, включаючи системи на основі правил, генеративні змагальні мережі та глибинні нейронні мережі. Можна також використовувати навчання з підкріпленням для покращення якості генерованої музики, а також механізми уваги для покращення зв'язності та структури генерованої музики.

4. Генерація візуального мистецтва. Є кілька методів генерації візуального мистецтва, включаючи використання глибинних нейронних мереж і використання генеративних змагальних мереж. Можна також використовувати перенесення стилю, а також навчання з підкріпленням для покращення якості згенерованого мистецтва.

Ці методи характеризуються здатністю генерувати нові та цінні результати, які неможливо передбачити на основі вхідних даних.

2.1. Варіаційні автокодер

Варіаційний автокодер [37-38] – це архітектура навчання, що складається з двох моделей: кодера (або моделі розпізнавання) і декодера (або генеративної

моделі). Перший стискає високорозмірні вхідні дані в низькорозмірне представлення, а другий розпаковує вектор представлення назад у вихідну область [16]. Класичні автокодери безпосередньо вивчають латентний вектор представлення. І навпаки, варіаційні автокодери вивчають (гаусівський) розподіл над можливими значеннями латентного представлення, тобто кодер вивчає середнє значення і (логарифм) дисперсії розподілу. Потім декодер вчиться реконструювати дані, приймаючи на вхід вектор, вибраний з розподілу.

Варіаційні автокодери навчаються шляхом оптимізації двох втрат: втрат при реконструкції та втрат при регуляризації. Перша – це лог-вірогідність реальних даних x , отриманих декодером з урахуванням їхніх латентних векторів z , тобто це помилка декодера в реконструкції x . Друга – це KL-розбіжність між розподілом, отриманим кодером, і попереднім розподілом, наприклад, гаусівським. Слід зазначити, що латентний вектор z на вході декодера отримується за допомогою так званого трюку репараметризації, тобто шляхом вибірки з розподілу, визначеного середнім значенням і дисперсією. Без цього вибірка призвела б до шуму в градієнтах, необхідних для навчання [39].

Математичне виведення повних втрат має коріння у варіаційному виведенні [40]. Дійсно, варіативні автокодери можна розглядати як ефективний і стохастичний метод варіаційного виведення, в якому нейронні мережі та стохастичний градієнтний спуск використовуються для вивчення наближення (тобто кодера) істинної апостеріорної оцінки [41].

У варіаційних автокодерів подібні високорозмірні дані відображаються на подібні розподіли. Це дає змогу вибрати випадкову точку z у цьому просторі і все одно отримати зрозумілу реконструкцію [16].

З іншого боку, варіаційний автокодер має тенденцію створювати розмиті зображення [42]. Також може трапитися так, що області з високою щільністю за попередніми даними мають низьку щільність за наближеними апостеріорними даними, тобто ці області не декодуються до зразків, подібних до даних [43]. Нарешті, мета може призвести до надмірно спрощеного представлення без

використання всієї потужності, отримуючи лише субоптимальну генеративну модель [44].

2.2. Трансформерні моделі

2.2.1. Основні концепти

Трансформерні моделі – це нейронні мережі, засновані на архітектурі трансформера [45]. Іноді їх називають фундаментальними моделями [46] через провідну роль, яку вони відіграють у мові, зорі та робототехніці. Трансформер – це архітектура для послідовного моделювання, яка не потребує рекурентних або згорткових шарів. Натомість вона покладається лише на механізм самоуваги [47], який моделює далекий контекст без послідовної залежності. Кожен шар складається з multi-head уваги (тобто декількох механізмів самоуваги, що працюють паралельно), мережі прямого поширення та залишкових зв'язків. Оскільки самоувага є агностичною до порядку токенів, для фіксації порядку використовується техніка, яка називається позиційним вбудовуванням [45].

В принципі, трансформер – це не що інше, як модель авторегресії: він працює, передбачаючи поточний токен на основі попередніх. Однак існує кілька принципових відмінностей.

Трансформер також можна навчити за допомогою маскованого моделювання: деякі вхідні лексеми випадковим чином маскуються, і модель повинна навчитися відновлювати їх з усього контексту, а не тільки з попередніх частин [48]. Можливість мати справу з дуже довгими послідовностями дозволяє підказка. Надаючи на вході підказку природною мовою, модель здатна генерувати бажаний вихід, наприклад, відповідь на питання, клас між набором класів для даного тексту або вірш у певному стилі [49]. Це робиться шляхом простої передачі підказки на вході у вигляді тексту, а потім використання моделі для прогнозування того, що буде далі (наприклад, відповідь на питання).

Ці переваги, разом з величезною кількістю доступних даних, зростаючою обчислювальною потужністю та паралелізмом, індукованим їхньою архітектурою,

призвели до того, що моделі на основі трансформерів стали найсучаснішими для низки задач. Тим не менш, обчислювальні витрати архітектури з [4] зростають квадратично зі збільшенням розміру вхідних даних.

2.2.2. Приклади моделей

За останні роки було запропоновано кілька підходів на основі трансформерів [48]. Сферою, в якій трансформери найбільше впливають на обробку тексту, є NLP. BERT – це кодер на основі трансформерів, навчений як передбачати наступне речення (в авторегресійному режимі), так і реконструювати замасковані лексеми з контексту. Було запропоновано кілька вдосконалених варіацій оригінальної моделі. Наприклад, RoBERTa здатна досягти більш якісних результатів лише за допомогою маскованого моделювання з більшим часом навчання та більшою кількістю даних; ALBERT отримує конкурентні результати з меншими параметрами, використовуючи міжреченнєву зв'язність як додаткову втрату; а DistilBERT використовує дистиляцію для навчання меншої моделі, щоб відповідати результатам BERT. Іншим основним підходом є той, що використовується в сімействі GPT. Тут декодер на основі трансформера навчається авторегресійним способом, додатково подаючи на вхід відносну задачу. Після навчання його можна використовувати для виконання широкого спектру завдань, надаючи опис або кілька демонстрацій завдання. Ефективність цього генеративного підходу «текст до тексту» була досліджена в T5. Було запропоновано багато інших великих мовних моделей для досягнення кращих результатів за допомогою більшої кількості параметрів і обчислень або даних. Mixture of Experts також можна використовувати замість мережі прямого поширення для навчання більшої, але легшої моделі (оскільки для кожного завдання використовуються лише її частини), як це зроблено в GLaM. Нарешті, BART ідеально поєднує в собі BERT-кодер (навчений на пошкодженому тексті з довільною функцією шуму) і GPT-декодер (навчений відновлювати вихідний текст авторегресійно). Така архітектура кодера-декодера здатна досягти найсучасніших результатів у машинному перекладі, а також в інших завданнях перетворення тексту в текст.

2.2.3. Застосування

Трансформерні моделі великих мов можуть бути використані майже для будь-якої задачі NLP, включаючи узагальнення, генерацію та взаємодію з текстом. Для цього модель можна використовувати як «заморожену» (тобто, надаючи латентні репрезентації у вхідних даних іншим моделям); можна точно налаштувати під конкретну мету; можна експлуатувати з нульовим, однократним або кількома спробами за допомогою підказки завдання або кількох демонстрацій у вхідних даних. Замість цього можна використовувати навчання з перенесенням для класифікації зображень за допомогою моделей на основі трансформерів, навчених на зображеннях. Також можуть бути використані інші специфічні для конкретної галузі методи: наприклад, PlotMachines навчається писати розповідні абзаци, не отримуючи підказок, а отримуючи контури сюжету і зображення попередніх абзаців.

З точки зору генеративного навчання Трансформери продемонстрували вражаючу продуктивність у створенні довгих послідовностей текстів і музики, а також у генеруванні зображень на основі вхідного тексту. Однак вони успішно застосовуються не лише до цих джерел даних. Наприклад, AlphaFold використовує архітектуру трансформерів для прогнозування структури білків; RecipeGPT – для генерації рецептів; а GitHub Copilot – для написання коду.

2.3. Методи на основі вхідних даних

2.3.1. Основні концепти

Останній клас методів, який розглядається в аналізі, не стосується іншої генеративної моделі. Навпаки, йдеться про інший спосіб вибірки результатів з (попередньо навченої) генеративної моделі, а саме за допомогою її вхідних даних. Можна використовувати два різні підходи. Перший полягає в ретельному виборі або оптимізації вхідних даних для генеративної моделі (наприклад, латентного вектора або текстової підказки), щоб отримати бажаний результат. Другий підхід полягає в оптимізації входу так, щоб він безпосередньо став бажаним виходом.

Втрати, що використовуються, ґрунтуються на особливостях, вивчених нейронними мережами.

2.3.2. Застосування

Методи, засновані на вхідних даних, можна використовувати з будь-якою генеративною моделлю для отримання бажаного результату. З мовними моделями вони можуть використовувати їхню загальність у кількох конкретних завданнях без їхнього доопрацювання. За допомогою генераторів зображень можна отримати малюнки, що відповідають заданим описам, або високоякісні, але все ж таки своєрідні картини, такі як роботи колористів, абстрактні чи інопланетні картини. Обидва типи методів, заснованих на вхідних даних, можна використовувати не лише для отримання бажаних результатів або передачі стилів; їх також можна використовувати для кращого аналізу того, що відбувається всередині мережі.

2.4. NLP inference

NLP inference, або natural language inference (виведення природньою мовою) – це завдання визначення того, чи логічно впливає задана «гіпотеза» з «передумови». Простіше кажучи, необхідно зрозуміти, чи є гіпотеза правдивою, тоді як передумова – це єдине знання про предмет.

NLI можна розглядати як класифікацію гіпотез на три класи на основі передумов: наслідок, протиріччя або нейтральна гіпотеза.

Існує пов'язана з цим проблема: перевірка фактів. Проблема перевірки фактів дуже схожа на NLI. Єдина відмінність полягає в тому, що передумови немає. Таким чином, фактчекінг складається з двох проблем: проблеми пошуку та NLI.

Коли людина вступає в нову гру, першим кроком є вивчення правил, за якими вона буде грати. У машинному навчанні бенчмарки – це фактичні правила, за якими грають дослідники.

SNLI – це золота класика бенчмаркінгу NLI, тому бенчмарк широко використовується, поважається і, відверто кажучи, застарів.

Набір даних SNLI базується на підписах зображень з корпусу Flickr30k, де підписи зображень використовуються як передумови. Гіпотеза була створена вручну працівниками Mechanical Turk відповідно до наступної інструкції:

1. entailment: напишіть один альтернативний підпис, який, безумовно, є точним описом фотографії.

2. neutral: напишіть один альтернативний підпис, який може бути точним описом фотографії.

3. contradiction: напишіть один альтернативний підпис, який неправильно описує фотографію.

SNLI має два суттєві недоліки:

1. Передумови обмежуються короткими описами фотографій, а отже, не містять часових міркувань, переконань чи модальностей.

2. Прості та короткі передумови вимагають простих і коротких гіпотез, тому еталон не є достатньо складним, щоб моделі могли легко досягти людського рівня точності.

MNLI змодельований за схемою набору даних SNLI, але охоплює цілий ряд усних і письмових текстів. Тому MNLI можна використовувати разом із SNLI, пропонуючи текст з десяти різних жанрів.

Передумови MNLI базуються на десяти джерелах або жанрах (на основі Відкритого американського національного корпусу):

1. face-to-face: транскрипції розмов між двома особами з колекції розповідей і розмов Шарлотти (Charlotte Narrative and Conversation Collection).

2. government: звіти, промови, листи та прес-релізи з урядових веб-сайтів у відкритому доступі.

3. letters: листи з Індіанського центру міжкультурної комунікації дискурсу фандрейзингу в Індіані.

4. 9/11: публічний звіт Національної комісії з розслідування терористичних атак на Сполучені Штати.

5. OUP: п'ять нон-фікшн творів про текстильну промисловість та розвиток дітей, опублікованих видавництвом Оксфордського університету.

6. slate: статті про популярну культуру з архіву журналу Slate;
7. telephone: транскрипції двосторонніх телефонних розмов з корпусу Switchboard Консорціуму лінгвістичних даних Університету Пенсильванії.
8. travel: путівники, опубліковані видавництвом Berlitz Publishing.
9. verbatim: дописи про лінгвістику для нефахівців з архіву Verbatim.
10. fiction: кілька вільно доступних творів сучасної художньої літератури, написаних між 1912 і 2010 роками.

Процес створення гіпотези виглядає наступним чином: краудворкеру представили вихідні дані і попросили скласти три нових речення:

1. entailment: те, що обов'язково є істинним або доречним, коли істинним є засновок.
2. contradiction: те, що обов'язково є хибним або недоречним, коли передумова є істинною.
3. neutral: таке, де не застосовується жодна з умов.

Важливою особливістю набору даних є те, що лише п'ять з десяти жанрів з'являються в навчальній вибірці, що робить інші п'ять жанрів невидимими для моделі. Ці невидимі жанри можуть бути використані для оцінки того, наскільки добре модель може узагальнювати невидимі джерела тексту.

SuperGLUE - це набір з десяти бенчмарків, які вимірюють продуктивність NLP-моделі в трьох завданнях:

1. NLI
2. Відповіді на запитання
3. Роздільна здатність кореляції

На основі результатів виконання цих завдань SuperGLUE має на меті виставити єдиний бал, який підсумовує можливості моделі в розумінні природної мови. SuperGLUE є розширенням дуже популярного бенчмарку GLUE з більш складними завданнями.

У SuperGLUE є два бенчмарки NLI: RTE та CB.

RTE (Recognizing Textual Entailment – розпізнавання текстових зв'язків) походить від щорічних змагань з розпізнавання текстових зв'язків. RTE містить

набори даних RTE1, RTE2, RTE3 і RTE5. Самі дані взяті з Вікіпедії та новинних статей. Відмінною особливістю RTE є те, що він маркований для двокласової класифікації, а не трикласової, тому нейтральної мітки немає.

CB, або Банк зобов'язань, – це корпус коротких текстів, в яких принаймні одне речення містить вбудоване твердження. Кожне з цих вбудованих речень анотується ступенем вірогідності того, що людина, яка написала текст, вважає, що це речення є істинним. Підсумкове завдання складається з трьох класів текстових наслідків на прикладах, взятих з Wall Street Journal, художньої літератури з Британського національного корпусу та Switchboard. Кожен приклад складається з передумови, що містить вбудоване твердження, і відповідної гіпотези - вилучення цього твердження.

2.5. Існуючі реалізації генерації ідей

Наразі найбільш відомими прикладами додатків, які використовують штучний інтелект для генерації ідей є InfraNodus [50] та IdeasAI [51].

InfraNodus (рис. 2.1) – це онлайн-інструмент для візуалізації ідеї процесу, створення та розвитку концепцій, ідей та асоціацій. Для визначення зв'язків між словами та ідеями він використовує графову базу даних та аналіз тексту, що дозволяє швидко створювати нові ідеї та переосмислювати старі. Користувачі можуть вводити слова та терміни у вікно для створення графа, а потім використовувати інструменти візуалізації та аналізу графа для здійснення пошуку нових ідей. Одна з головних функцій InfraNodus – це генерація графів (рис. 2.1), що дозволяє візуалізувати зв'язки між словами та ідеями. Візуалізація допомагає користувачеві більш краще зрозуміти взаємозв'язки між різними ідеями і темами. InfraNodus може бути корисним інструментом для людей, які працюють над проектами, що вимагають творчого мислення, таких як письменники, маркетологи, дизайнери та інші.



Рисунок 2.1. Демонстрація роботи InfraNodus [50]

InfraNodus може працювати з текстами будь-якої мови, проте найкраще підходить для мов з латинським алфавітом. Інтерфейс користувача InfraNodus досить простий і зрозумілий, що дозволяє навіть не-технічним користувачам швидко зрозуміти, як користуватися програмою.

InfraNodus є комерційним продуктом, але існує безкоштовна версія програми з обмеженнями на кількість графів та обмеженнями в роботі з текстами певного обсягу.

InfraNodus користується методом «тематичної кластеризації» [52]. Суть його полягає в тому, що програма намагається виявити тематичні зв'язки між словами, використовуючи статистичні алгоритми. На основі цих зв'язків InfraNodus розташовує ідеї у вигляді кластерів на графі, де кожен кластер відображає тему, а кожен вузол (слово чи фраза) – ідею.

InfraNodus також може використовувати алгоритми машинного навчання, щоб навчитися виявляти тематичні зв'язки між словами на основі великої кількості текстових даних. Це дає можливість програмі відобразити більш складні зв'язки між ідеями, які можуть бути складні для розуміння людиною.

Після того, як граф зібраний, користувач може використовувати його, щоб знайти нові зв'язки між ідеями, розвинути вже наявні ідеї або створити нові концепції. Також InfraNodus має функцію «генерації ідей», яка дозволяє автоматично згенерувати нові ідеї на основі зібраних на графі даних.

InfraNodus також може бути використаний для аналізу соціальних мереж та збору даних з інтернету. Крім того, програма має ряд інструментів, які допомагають користувачеві візуалізувати та організувати ідеї, включаючи можливість додавати теги та мітки до графів.

InfraNodus використовує наступні методології машинного навчання для аналізу текстової інформації та генерації ідей:

1. Токенізація: цей алгоритм розбиває текст на окремі слова або токени. Це необхідно для подальшого аналізу та обробки тексту.

2. Лематизація: цей алгоритм зводить слова до їх базової форми або леми. Наприклад, слова «біг» та «бігти» будуть зведені до леми «бігти».

3. Кластеризація: цей алгоритм групує токени або ідеї в кластери на основі схожості між ними. Це дозволяє виділити теми та концепти, які можуть бути корисними для подальшого аналізу.

4. Аналіз тональності: цей алгоритм визначає, чи є даний текст позитивним, негативним або нейтральним за допомогою аналізу емоційних забарвлень слів (семантичного аналізу).

Для аналізу тексту також використовується RNN, яка може аналізувати послідовності даних, такі як речення. Крім того, використовуються такі алгоритми, як LSTM та GRU, які дозволяють вирішувати проблему зниклого градієнту в RNN.

Для генерації ідей InfraNodus використовує глибинні нейронні мережі, зокрема, автокодер (autoencoder). Автокодери навчаються відтворювати вхідні дані на виході, проте вони обмежені в розмірності внутрішнього представлення, що дозволяє вони використовуються для зменшення розмірності вхідних даних та виявлення складних залежностей у даних. У InfraNodus також використовуються глибинні генеративні моделі, такі як VAE та GAN, для створення нових ідей на основі вхідних даних.

Крім того, система використовує RL, щоб навчитися підлаштовувати результати аналізу на основі фідбеку від користувачів.

У загальному, InfraNodus – це потужний інструмент для створення мапи ідей та текстового аналізу, який може бути використаний у багатьох галузях, включаючи бізнес, науку, соціальні мережі та інтернет-маркетинг.

IdeasAI (рис. 2.2) – це інструмент генерації ідей, який використовує штучний інтелект для створення нових ідей. Він був розроблений командою OpenAI, однією з провідних компаній в галузі штучного інтелекту. Основним принципом роботи IdeasAI є створення зв'язків між різними ідеями та концепціями, що дозволяє згенерувати нові ідеї, які можуть бути корисними у різних сферах життя.

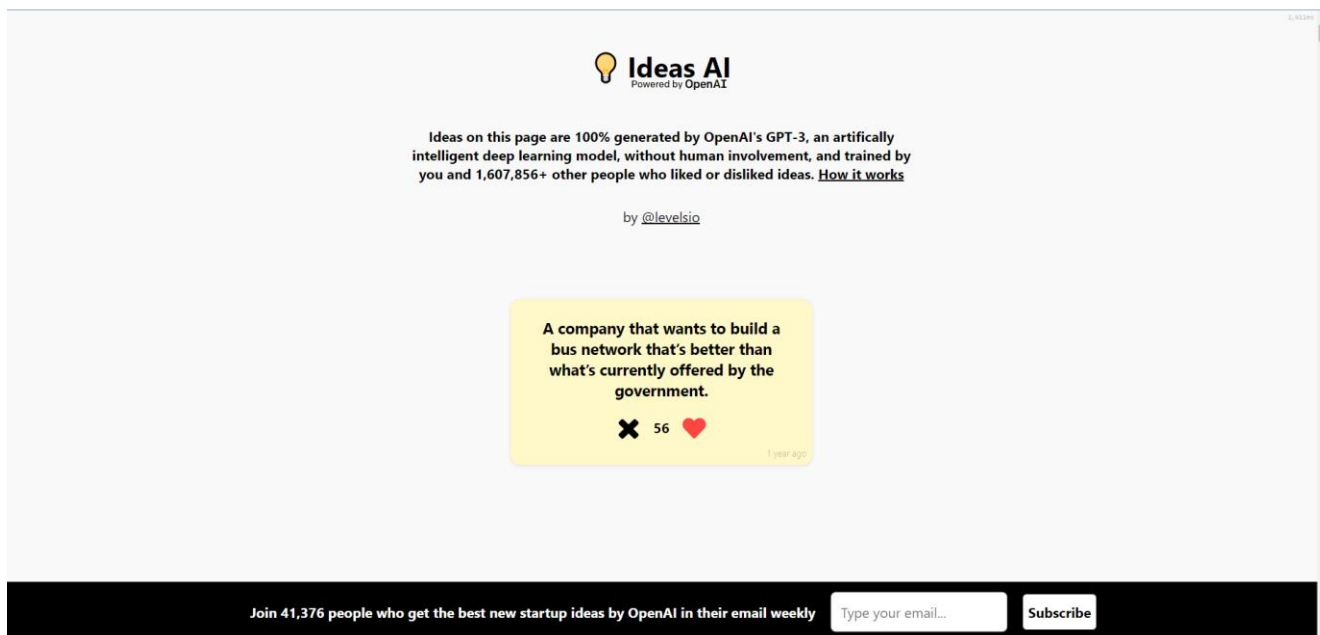


Рисунок 2.2. Вигляд головної сторінки IdeasAI [51]

Ідеї генеруються на основі вхідних даних, таких як ключові слова, теми, тексти або документи. Робота IdeasAI полягає у використанні глибинних нейронних мереж, які були навчені на великих обсягах текстових даних. Ці мережі аналізують текстову інформацію та шукають в ній зв'язки між словами та концепціями, що дозволяє створити нові ідеї.

Крім того, IdeasAI використовує генетичні алгоритми, які дозволяють відібрати найкращі ідеї та поєднати їх для створення нових концепцій. Також в

алгоритм вбудовані механізми контролю, що дозволяють забезпечити, що згенеровані ідеї будуть відповідати вхідним даним та вимогам користувача.

Його основна функція полягає в тому, щоб допомогти користувачам знайти нові ідеї для своїх проектів або бізнесу. Користувачі можуть почати роботу з IdeasAI, введенням своєї початкової ідеї або ключового слова в систему. IdeasAI потім аналізує цю інформацію та створює список пропозицій з новими ідеями, які можуть бути корисними для користувача. Додаток використовує набір алгоритмів машинного навчання, щоб вибрати найкращі ідеї та порекомендувати їх користувачам.

Одна з головних переваг IdeasAI полягає в тому, що додаток використовує методи машинного навчання для аналізу багатьох джерел інформації. Це дозволяє зібрати ідеї з різних джерел, таких як соціальні мережі, блоги та інші джерела Інтернету. Додаток також може аналізувати тексти користувача та з'ясовувати, які ідеї могли би бути корисними для нього. Крім того, IdeasAI має можливість аналізувати поточні тенденції та прогнозувати, які ідеї можуть бути популярними у майбутньому.

IdeasAI використовує різні алгоритми ГН, такі як генеративні моделі, RNN та CNN, для створення ідей з використанням текстових даних. Основна ідея полягає в тому, щоб навчити нейронну мережу прогнозувати наступний слово або фразу, використовуючи попередній текст як вхідні дані.

Для цього спочатку дані підготовлюються та векторизуються за допомогою алгоритмів обробки природньої мови. Потім створюється модель нейронної мережі, яка навчається на основі цих векторизованих даних. У випадку генеративної моделі, нейронна мережа навчається генерувати нові послідовності слів або фраз, які схожі на ті, які були навчені.

У випадку RNN, модель може використовуватись для прогнозування наступного слова в контексті попереднього тексту. Такі моделі зазвичай використовуються для генерації тексту на основі вхідних даних.

У випадку CNN, модель може використовуватись для виявлення певних шаблонів у тексті, які потім можуть бути використані для генерації ідей.

Генетичний алгоритм – це еволюційний алгоритм, який використовується для розв'язування оптимізаційних завдань. Генетичний алгоритм моделює процес природного відбору, використовуючи генетичні операції, такі як схрещування і мутація, для генерації нових і кращих рішень на основі попередніх.

У IdeasAI генетичний алгоритм використовується для генерації нових ідей. Алгоритм починається зі створення початкової популяції випадкових ідей. Кожна ідея кодується як хромосома, яка складається з генів. Гени представляють собою окремі аспекти ідей, такі як тема, ключові слова, переваги і недоліки.

Далі проводиться оцінка кожної ідеї за допомогою функції придатності. Ця функція відображає те, наскільки добре ідея відповідає вимогам користувача або завданню. Ідеї з вищими значеннями функції придатності мають більші шанси на виживання в наступному поколінні.

Далі застосовуються генетичні операції для генерації нових ідей у наступному поколінні. Схрещування дозволяє поєднати гени з різних хромосом, щоб створити нову хромосому, яка поєднує в собі риси обох батьківських хромосом. Мутація змінює випадковим чином деякі гени, щоб створити ще більше різноманітності в наступному поколінні.

Далі, в IdeasAI, проводиться оцінка кожної ідеї згідно зі значенням її фітнес-функції. Чим вище значення фітнес-функції, тим краще ідея. Згідно з результатами оцінки, ідеї сортуються у порядку спадання фітнес-функції.

Користувач може налаштувати кілька параметрів генетичного алгоритму, таких як розмір популяції, кількість поколінь, ймовірність мутації та інші, щоб підібрати оптимальні умови для генерації ідей.

Інтерфейс IdeasAI дозволяє користувачеві переглядати та оцінювати ідеї, що створюються алгоритмом, і дозволяє зберігати та експортувати їх у форматі текстових файлів. Це дозволяє використовувати згенеровані ідеї для подальшого аналізу та розробки проектів.

IdeasAI може бути корисним для різних галузей, включаючи науку, бізнес, маркетинг, мистецтво, літературу та інші. Він може бути використаний для

розробки нових продуктів, створення маркетингових кампаній, відкриття нових напрямків досліджень та багато іншого.

2.6. Основні недоліки існуючих реалізацій

Аналізуючи додатки з попередньому підрозділі, то можна виділити наступне:

1. У випадку IdeasAI ідеї вже є готовими, часто загальними та не мають можливості уточнень, модифікацій. Тобто немає інтерактивної взаємодії з користувачем, що дозволяла б розвивати певну ідею до задовільного рішення.
2. Окрім цього, використовувана модель не є самонавчаючою, тобто відстає від актуальної інформації та не містить підґрунтя згенерованих ідей, окрім кількості голосів користувачів.
3. Рішення InfraNodus вирішує частину з вищезазначених проблем, але додає нові — додаток є перенасичений інформацією та вимагає тривалої взаємодії для того, щоб прийти до якихось ідей чи рішень. Готові взаємозв'язки між інформацією накладають обмеження в креативності.
4. Скупість бази даних: InfraNodus та IdeasAI використовують лише GPT та Google пошук без необхідної технічної бази, такої як патенти, статті, тренди та актуальні новини.

Загалом, хоча InfraNodus та IdeasAI можуть бути потужними інструментами для генерації ідей з використанням штучного інтелекту, але вони мають специфіку використання, обмеження та недоліки, які можуть бути важливими для користувачів при виборі інструменту для своїх потреб.

2.7. Методологія розробки застосунку для генерації ідей

Основним фундаментом для генерації нових ідей є використання штучного інтелекту, або машинного навчання, що є підобластю штучного інтелекту. Щоб

отримати змогу генерувати ідеї (або розширювати існуючі), необхідно спочатку натренувати моделі машинного навчання на великому обсязі текстових даних. Краще за всі для навчання на великому масиві даних (Big Data) підходять нейронні мережі, що є моделями ГН. Наприклад, можна використовувати такі архітектури як LSTM [53] або GPT [49], які підходять для створення текстів зі значенням, включаючи генерацію ідей.

LSTM – це рекурентна нейронна мережа, яка здатна до вирішення проблеми зниклого градієнту в рекурентних мережах. Її архітектура складається з декількох повторюваних блоків, кожен з яких має три порти: вхідний, вихідний та забувальний порти. Кожен порт приймає на вхід вектори даних та здійснює різні операції для керування інформацією, що передається в мережу. Вхідний порт приймає нові дані та оброблює їх з урахуванням попередніх даних, що надійшли від попередніх блоків мережі. Забувальний порт відповідає за забування деякої частини інформації, переданої в мережу. Вихідний порт відповідає за відповідь мережі, яка передається наступним блоком. Таким чином, LSTM може зберігати довготривалу залежність вхідних даних та передавати цю інформацію наступним блоком мережі. Архітектура LSTM показана на рисунку 2.3.

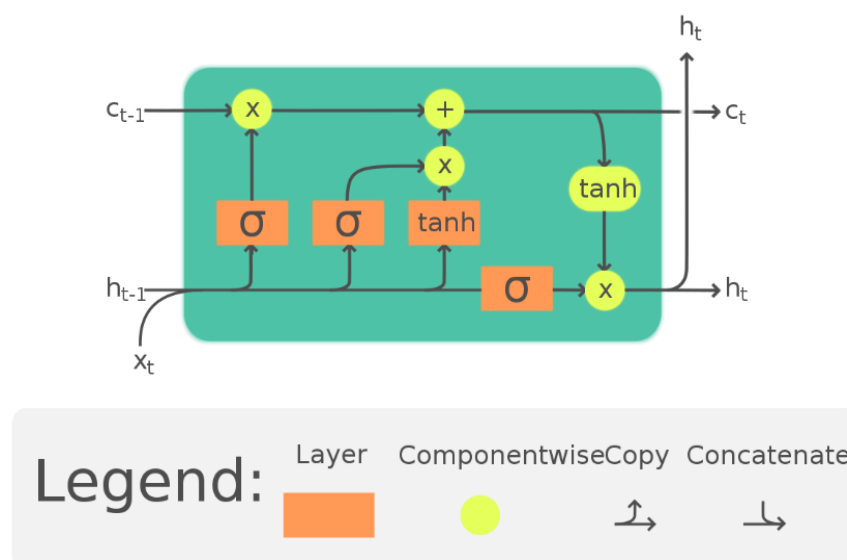


Рисунок 2.3. Архітектура моделі LSTM [54]

Архітектура ж GPT продемонстрована на рисунку 2.4.

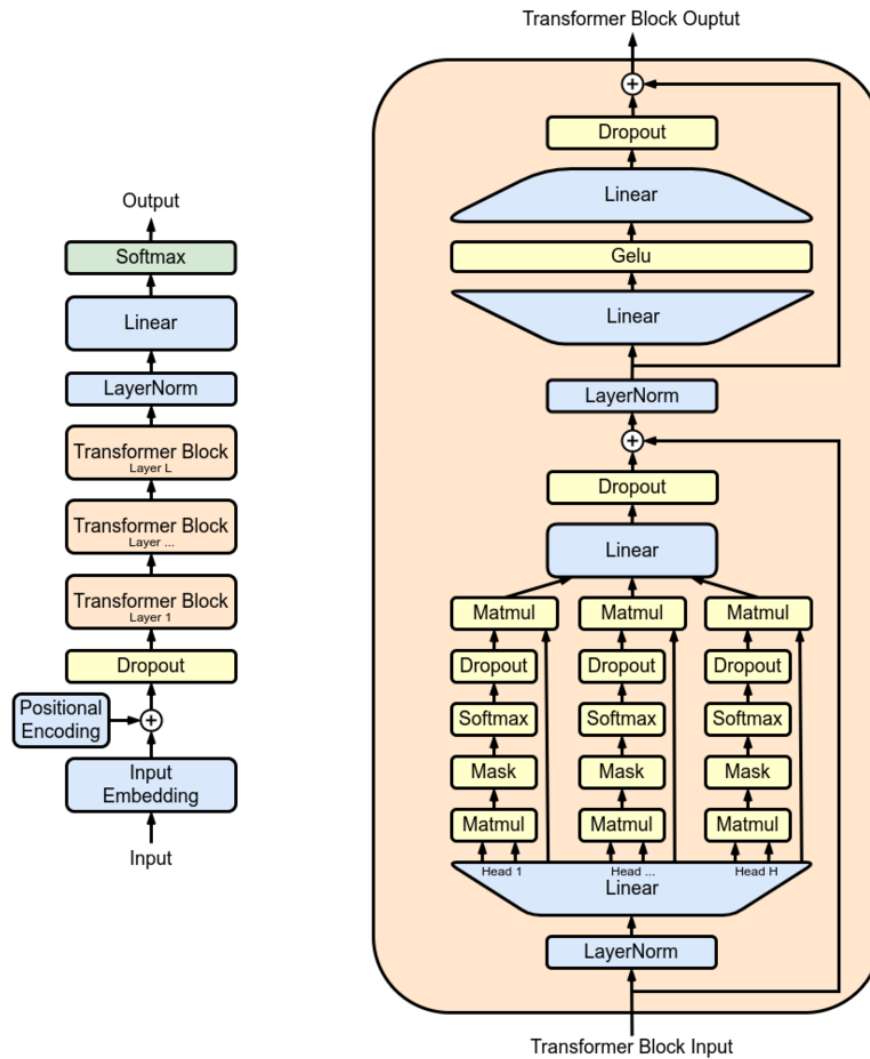


Рисунок 2.4. Архітектура моделі GPT [55]

Подібна архітектура заснована на трансформері. GPT складається з багатьох блоків трансформера, де кожний блок містить декілька шарів самоуваги та мережових прямих зв'язків. Одна з особливостей даної моделі – це те, що вона працює з авторегресійною моделлю, що означає, що вона генерує текст по одному слову або символу за раз, використовуючи заздалегідь навчену модель для передбачення наступного слова в залежності від попередніх.

Трансформер дозволяє моделювати довгострокові залежності в тексті завдяки кільком енкодерам і декодерам, що взаємодіють між собою для обробки та генерації тексту. Кожен енкодер/декодер складається з двох підрозділів: блоку

самоуваги та двох підключених до нього повнозв'язних шарів (шару прямого поширення).

Блок самоуваги дозволяє моделі зосередитися на певних частинах тексту та залежностях між ними. Шар прямого поширення отримує вихідний вектор з блоку самоуваги та застосовує до нього нелінійні функції, щоб зробити прогноз наступного слова. Для навчання GPT використовується велика кількість текстових даних, які використовуються для навчання моделі з використанням методу без учителя.

Подібні моделі можна використовувати для генерації нових ідей шляхом збільшення текстів, які генеруються на основі вхідних даних. Проте є ризик того, що створені моделі можуть бути недосконалими і генерувати недостатньо точні ідеї або не зовсім нові. Тому для досягнення кращих результатів краще поєднувати генеровані моделлю ідеї з ідеями, що вже існують, і проводити їх подальшу обробку та аналіз.

У даній роботі було вирішено зробити акцент на вже існуючих моделях від OpenAI на основі архітектури GPT. Отримавши відкритий доступ по API через «Hugging Face» до моделей ГН від OpenAI, були взяті GPT-2, архітектура якого показана на рисунку 2.5, а також text-davinci-003 (рис. 2.6), що є представником сімейства архітектур GPT-3.

GPT-2 є наслідком роботи над попередньою моделлю GPT. Це покращена версія архітектури GPT, яка відмінна більшою кількістю шарів, механізмів уваги, а також блоків для обробки тексту. Архітектура моделі складається з 12 шарів, що дозволяє моделі зрозуміти більш складні залежності в тексті. Вона має більше 1.5 мільярда параметрів, що робить її однією з найбільш потужних моделей для генерації тексту на даний момент. Усе це дозволяє GPT-2 працювати з довшими текстами, генерувати тексти вищої якості з меншою кількістю помилок, а також навчатись на меншій кількості даних, що робить дану модель більш гнучкою для використання в різних доменах і з різними типами даних. Крім того, GPT-2 має механізми для вирішення декількох проблем, пов'язаних з генерацією тексту, таких як згорткові шари, механізми уваги та відповідність навчанням з вчителем.

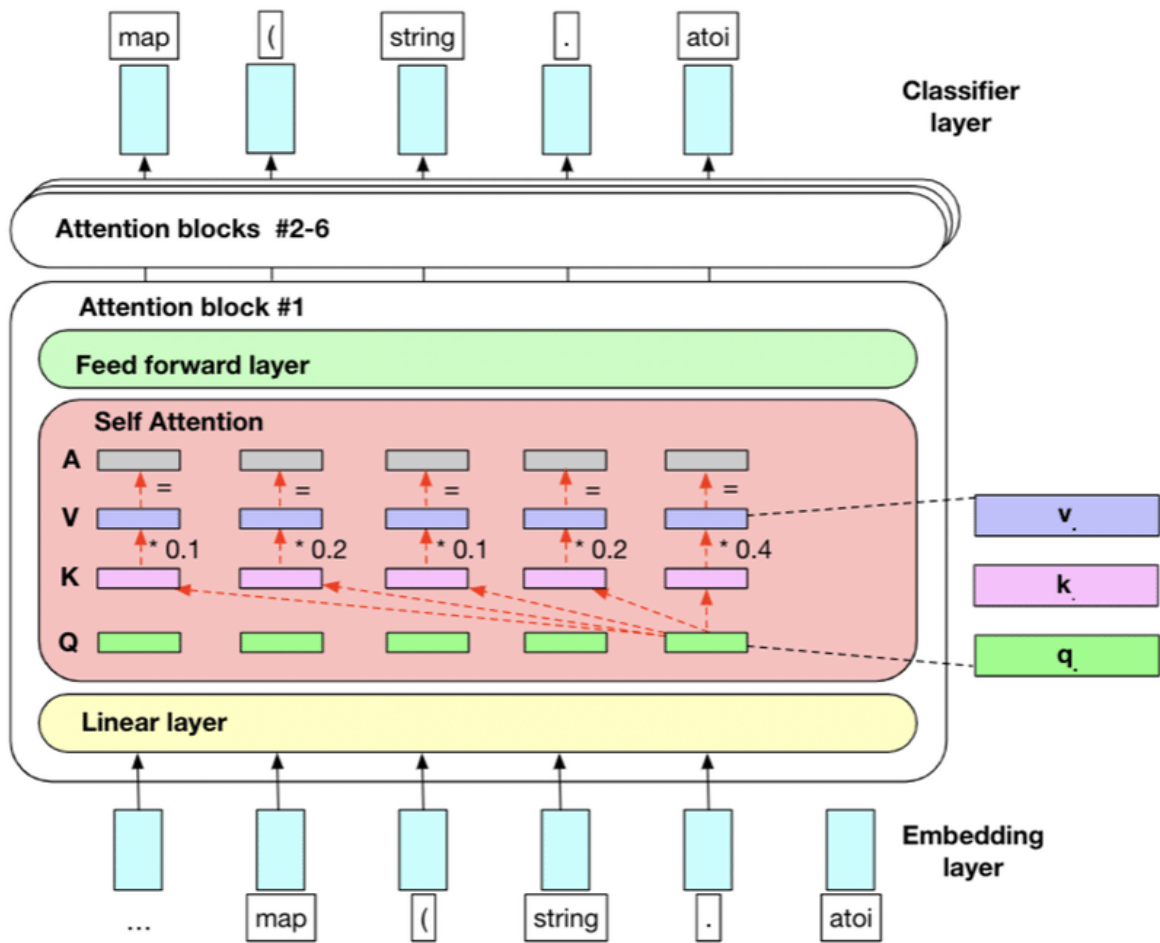


Рисунок 2.5. Архітектура моделі GPT-2 [56]

Також вона підтримує «нормалізацію інстанцій», що дозволяє зменшити перенавчання в моделі і зробити її більш стійкою.

GPT-3 містить 175 мільярдів параметрів, що є значною кількістю порівняно з її попередником – GPT-2, яка містить у майже 117 разів менше параметрів. GPT-3 складається з 96 шарів з 12 головками у кожному шарі. До особливостей моделі можна віднести наявність двох типів шарів: механізм самоуваги та повнозв'язний шар (fully connected layer), що дозволяє більш точно розуміти зв'язки між словами та реченнями.

Крім того, GPT-3 використовує механізм проміжної адаптації (intermediate fine-tuning), який дозволяє моделі адаптуватися до нових завдань шляхом тренування на невеликих даних.

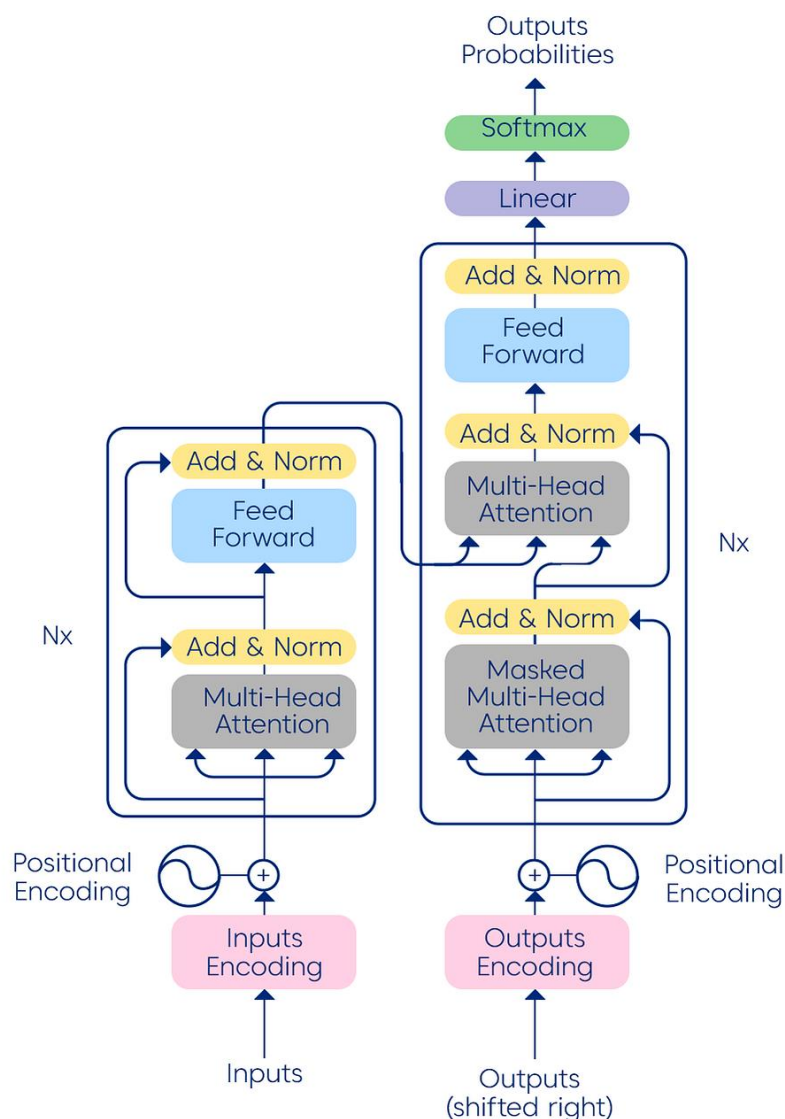


Рисунок 2.6. Архітектура моделі GPT-3 [57]

Однією з ключових особливостей text-davinci-003 є її здатність до здійснення зв'язку між різними областями знань, в тому числі тих, які не пов'язані зі словесним висловлюванням. Вона може поєднувати знання з різних джерел та використовувати їх для розв'язання складних завдань, таких як розробка нових продуктів, покращення існуючих технологій та створення медичних пристроїв.

Взаємодія між GPT-2 і text-davinci-003 відбувається наступним чином: спочатку GPT-2 генерує текст на основі заданого початку; потім цей згенерований текст використовується як вхід для моделі text-davinci-003, яка досконаліше оброблює вхідний текст і доповнює його більш точними деталями або виправляє помилки, допущені моделлю GPT-2.

Крім того, були використані бібліотеки мови програмування Python для обробки природнього мовлення (NLP) WordNet та NLTK. WordNet є лексичною базою даних, що зберігає слова англійської мови та їх семантичні відношення. Вона містить синоніми, антоніми, гіпероніми (батьківські терміни) та гіпоніми (дочірні терміни) для кожного слова. WordNet формує синоніми та антоніми шляхом аналізу відносин між словами, зокрема гіперонімів та гіпонімів.

Щоб отримати синоніми та антоніми слова, NLTK використовує методи WordNet, які дозволяють знайти семантично близькі слова за їхніми відносинами. Для прикладу, щоб отримати синоніми слова «happy», можна викликати метод `synsets()` з бібліотеки WordNet в NLTK та вибрати відповідний синонім з повернутих результатів.

В NLTK є токенізатор речень `unlkt`, який дозволяє розділяти текст на список речень за допомогою алгоритму навчання без вчителя для побудови моделі для слів-абревіатур, словосполучень і слів, що починають речення. Перед використанням його потрібно навчити на великій колекції відкритих текстів цільовою мовою. Проте є вже натренований токенізатор для англійської мови.

Для того, щоб була можливість візуально і з легкістю продемонструвати ідеї, які нагенерували моделі ГН, було прийнято рішення розробити веб-додаток за допомогою Vue.JS. Він є одним із найбільш популярних фреймворків для веб-програмування. Основними перевагами Vue.JS є:

1. Простота вивчення та використання. Vue.js надає дуже простий та логічний синтаксис, що дозволяє швидко засвоювати основні концепції та починати розробку.

2. Гнучкість. Vue.js дозволяє розробляти як малий інтерактивний компонент, так і повноцінний веб-додаток. Фреймворк підтримує модульну структуру та компонентний підхід, що дозволяє легко розбивати веб-додаток на окремі складові.

3. Висока продуктивність. Vue.js надає можливість для оптимізації швидкості завантаження сторінки та покращення продуктивності веб-додатка.

4. Широкі можливості. Vue.js має велику кількість додаткових бібліотек та розширень, що дозволяє легко додавати нові функціональність та забезпечує високу гнучкість у виборі інструментів для розробки.

Крім того, Vue.js має дуже гарну документацію та широку спільноту, що дозволяє швидко знайти рішення будь-яких проблем під час розробки.

РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ ДЛЯ СТВОРЕННЯ ІДЕЙ

3.1. Макет застосунку

На рисунку 3.1 наведена data flow діаграма застосунку для створення ідей.

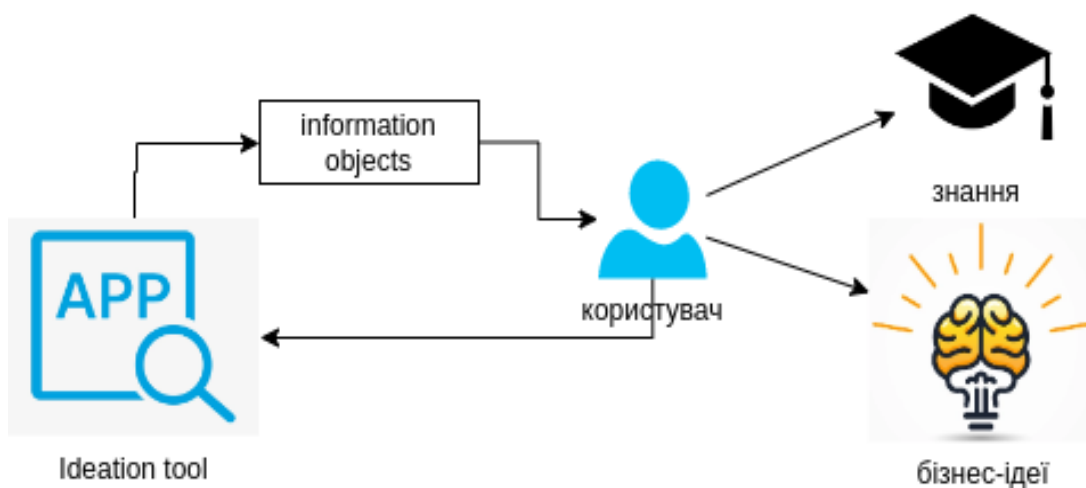


Рисунок 3.1. data flow діаграма

Як можна побачити з рисунку вище, основна ідея полягає у тому, що користувач користується додатком, який генерує різну інформацію (information objects), поступово набуваючи нових знань для генерування нових бізнес-ідей. Інформаційні ресурси будуть подаватись користувачеві у вигляді списку результатів, кожен з яких містить категорію, короткий опис та додаткові деталі. Подібний список можна повертати у JSON форматі.

Інформаційними ресурсами в даному випадку є:

1. Останні наукові публікації, інтелектуальна власність (патенти), новини. Доступ до них можна налаштувати за допомогою парсингу та API.
2. Факти, інсайти, ідеї. Надання доступу до подібної інформації виконується шляхом використання штучного інтелекту (архітектури моделей ГН для генерації ідей описані у минулому розділі).
3. Суміжні фрази, теми. Пропозиції щодо даної інформації демонструються за допомогою трендів (Google Trends API), а також штучного інтелекту.

3.2. Архітектура системи

На основі запропонованого макету (рис. 3.1) була спроектована архітектура системи для створення ідей, що показана на рисунку 3.2.

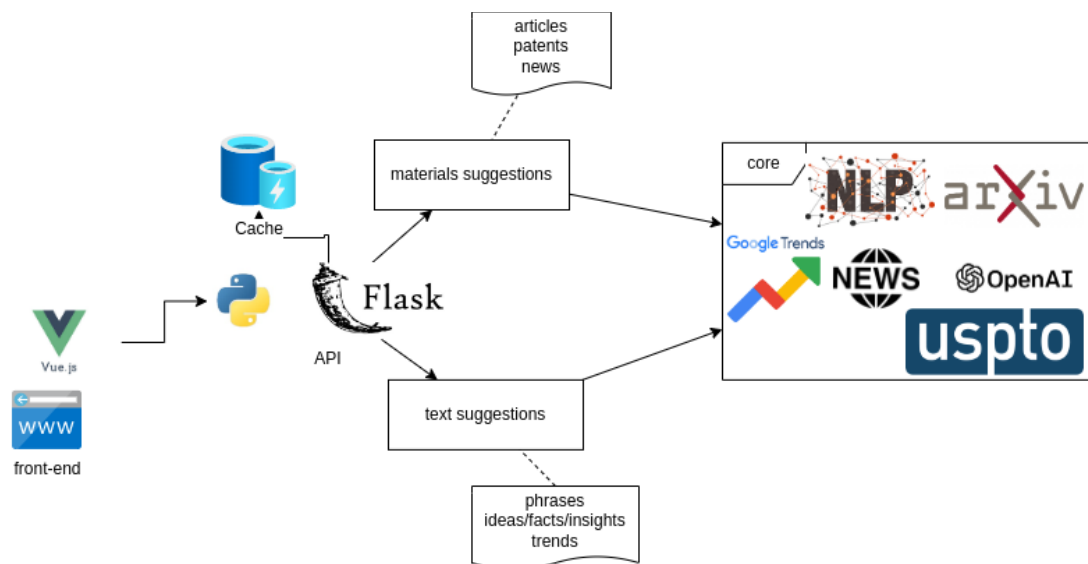


Рисунок 3.2. Архітектура технічно-орієнтованого застосунку для створення ідей з використанням штучного інтелекту

Як було зазначено у попередньому розділі, фронт-енд реалізовувався за допомогою Vue.JS. Python використовується для бек-енду, де, за допомогою фреймворку Flask, здійснюється доступ до різних API.

3.3. Імплементация основної частини додатку

Flask надає доступ по API для взаємодії з моделями ГН, а також arXiv, patents_view, arxivexplorer і Google News для отримання посилань відповідно до запиту користувача. Наприклад, даний фрагмент коду з функцією «patents_view_sample»:

```
def patents_view_sample(query):
    print('patents_view_sample')
    url = "https://api.patentsview.org/patents/query"

    # Set the parameters for the API request
    params = {
        "q": '{"_text_any":{"patent_abstract":"%s"}}' % query,
```

```

    "f": '["patent_title", "patent_date", "patent_abstract", "patent_number"]',
    "o": '{"per_page": %d}' % MAX_RESULTS,
    "s": '[{"patent_date": "desc"}]',
}

response = requests.get(url, params=params)

if response.status_code == 200:
    data = response.json()
    results = data.get('patents', [])

    ret = []
    for patent in results:
        ret.append({
            'title':patent.get('patent_title', 'Not Found'),
            'date':patent.get('patent_date', 'Not Found'),
            'number':patent.get('patent_number', 'Not Found'),
            'summary':patent.get('patent_abstract', 'Not Found')
        })
    return ret
else:
    raise f"Error fetching data from PatentsView API: {response.status_code}"

```

використовує API «PatentsView» для пошуку патентів за ключовим запитом. Вона виконує HTTP-запит з відповідними параметрами і повертає результат у вигляді списку словників з назвою, датою публікації, номером та коротким описом патенту.

Цей ж фрагмент коду з функцією «arxiv_explorer_sample»:

```

def arxiv_explorer_sample(query):
    print('arxiv_explorer_sample')
    url = 'https://us-west1-semanticexplorer.cloudfunctions.net/semantic-explorer-db'

    resp = requests.get(url, params={'query': query})

    elems = []
    for el in resp.json():
        url = 'https://arxiv.org/abs/' + el['id']
        el = el['metadata']
        elems.append({'title':el['title'], 'date': parse(el['date']), 'url': url,
'abstract': el['abstract']})

    elems.sort(key=lambda el: el['date'], reverse=True)

    elems = elems[:MAX_RESULTS]
    return elems

```

використовує функціонал «SemanticExplorer». Це семантичний пошук, який працює через OpenAI embeddings, завдяки чому з'являється можливість шукати статті, які схожі на вказаний запит. Вона виконує HTTP-запит з відповідними параметрами і

повертає результат у вигляді списку словників з назвою, датою публікації, посиланням та коротким описом статті.

Даний фрагмент коду з функцією «get_chatgpt_ideas»:

```
def get_chatgpt_ideas(query):
    print('chatgpt_ideas_sample')

    def generate_ideas(prompt):
        prompts = [prompt for _ in range(MAX_RESULTS)]
        response = openai.Completion.create(
            engine="text-davinci-003",
            prompt=prompts,
            max_tokens=500,
            stop=None,
            temperature=0.5,
        )

        ideas = [choice.text.strip() for choice in response.choices]
        return ideas

    prompt = "Generate idea or fact or insight related to '%s' that can be useful
for tech-startup" % (query)
    ret = generate_ideas(prompt)
    return ret
```

реалізує функціонал генерації ідей з використанням мовних моделей, зокрема моделі GPT-3 від OpenAI. Функція generate_ideas(prompt) створює певну кількість промптів для генерації ідей на основі введеного користувачем запиту prompt. Далі, функція openai.Completion.create() генерує текстову відповідь на кожен з промптів з використанням моделі GPT-3. Змінна engine вказує на тип використовуваної моделі. У даному випадку використовується модель «text-davinci-003». Параметр max_tokens визначає максимальну кількість символів в згенерованій відповіді. Параметр stop визначає знаки пунктуації, при зустрічі з якими генерація тексту буде зупинена. В даному випадку знаки пунктуації не використовуються, тому stop має значення None. Параметр temperature визначає ступінь «креативності» генерованого тексту. Значення параметру від 0 до 1, де 1 відповідає найбільш «креативному» тексту. Функція get_chatgpt_ideas(query) використовує функцію generate_ideas(prompt) для генерації ідей на задану тему, яка передається в параметрі query. Перед генерацією промпти формуються за допомогою задалегідь заданого шаблону, де %s замінюється на значення параметра query. Отримані згенеровані ідеї повертаються як результат роботи функції.

Фрагмент коду з функцією «get_hugging_face_inference»:

```
def get_hugging_face_inference(query):
    print('hugging_face_inference_sample')

    generator = pipeline('text-generation', model='gpt2')
    results = generator("New idea/fact/insight related to the %s is" % query,
max_length=200,
                        num_return_sequences=MAX_RESULTS)

    ideas = [r['generated_text'].replace('\n\n', '\n') for r in results]
    return ideas
```

використовує модель з пакету Hugging Face для генерації тексту. Функція приймає рядок query як вхідний параметр, що передається у якості промпта для генерації тексту. У функції створюється інстанс генератора тексту, використовуючи функцію pipeline() з пакету Hugging Face. Зокрема, для генерації тексту використовується модель gpt2. Після того, як генератор був створений, функція generator() викликається з наступними параметрами:

- text: це початковий текст, який використовується як prompt для генерації тексту. У цьому випадку використовується рядок "New idea/fact/insight related to the %s is" % query.
- max_length: це максимальна довжина згенерованого тексту, що повертається.
- num_return_sequences: це кількість згенерованих послідовностей тексту, які повертаються. У цьому випадку використовується константа MAX_RESULTS.

Після виклику функції generator(), функція створює список ideas, до якого додаються всі згенеровані послідовності тексту (текст розділяється за допомогою символу переносу рядка). Функція повертає цей список ideas.

3.4. Інфраструктура додатку (фронтенд та бекенд)

На бекенд частині використовується мікро веб-фреймворк Flask для веб-сервісу, що надає search ендпоінт.

```
app = Flask(__name__)
```

```
cache = Cache(app, config={'CACHE_TYPE': 'simple', 'CACHE_DEFAULT_TIMEOUT': 60*5})
```

Для кешування результатів запитів використовується flask_caching рішення як розширення до Flask, що базується на таймінгу та запиті користувача, що є ключем в кеші.

```
@app.route('/search')
@cache.cached(timeout=60*5, query_string=True)
def search():
```

Таким чином, для використання API додатку достатньо встановити необхідні залежності з requirements.txt, та запустити веб-сервіс через python main.py — локально чи на хмарних рішеннях таких як AWS EC2, heroku. Після цього результати згенерованих ідей можна отримувати через search ендпоінт.

В свою чергу для фронтенд частини було використано SPA підхід - web.html, що поєднує в собі html, css, Vue.js та jquery.

```
const app = new Vue({
  el: '#app',
  data: {},
  methods: {
    search() {
      ...
      // Make API request
      axios.get(.../search', { params })
        .then(response => {
          this.results = response.data.results;
          this.results.sort(() => Math.random() - 0.5);
        })
      ...
    },
    showDetails(index) {
      this.selectedResult = index;
    },
  },
});
```

Для взаємодії з фронтенд функціоналом потрібно перейти на головну сторінку додатку /, що пошле запит на відповідний ендпоінт веб-сервісу та поверне web.html, як результат.

РОЗДІЛ 4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

4.1. Робота системи для створення ідей

Система була реалізована відповідно до архітектури (рис. 3.2), що була приведена у попередньому розділі. На рисунку 4.1 показано яким чином система генерує нові ідеї шляхом роботи моделей ГН.

The screenshot displays the 'Ideation Tool' interface. At the top, the title 'Ideation Tool' is centered. Below it, a 'Details' section shows a JSON object:

```
{ "main_info": "New idea/fact/insight related to the grocery business is to provide something that takes a little bit more effort for a small company o", "type": "AI-free" }
```

. A search bar contains the query 'grocery business' with a 'Search' button. Below the search bar, there are filters: Arxiv, News, Patents, OpenAI, AI-free, Trends. The 'Number of results per type' is set to 15. The main content area is a table with 8 rows of results. Each row has a model type in the first column and a text-based idea in the second column. The results include ideas from OpenAI and AI-free models, such as leveraging mobile technologies, streamlining operations, and using technology for inventory management.

| Model Type | Idea |
|------------|---|
| OpenAI | Grocery businesses can benefit from leveraging mobile technologies to offer customers a more personalized shopping experience. By deploying technologies like mobile applications, digital coupons, and location-based services, grocery stores can increase customer engagement and loyalty, ultimately leading to greater sales. |
| OpenAI | Grocery businesses can benefit from leveraging technology to streamline their operations. By implementing automated inventory tracking, ordering, and delivery systems, businesses can reduce costs, increase efficiency, and improve customer satisfaction. Additionally, leveraging online grocery delivery services can help businesses reach a larger customer base and increase sales. |
| AI-free | New idea/fact/insight related to the grocery business is that groceries are expensive to produce so it has been a big step to get them all to this level of quality - including those that are high price-oriented or high-margin. I think that the goal would be to give the supermarket the space and to increase cost-savings that stores would experience as an increased sense of consumer choice. If we can go to one of the better grocery retailers in the country I'd like them to expand its footprint beyond just my supermarket and to create a sense of personal shopping experience. To me, the grocery store as a business model that has made this possible for more and more consumers is actually more compelling. |
| OpenAI | Grocery businesses can benefit greatly from investing in technology such as artificial intelligence and machine learning to improve customer experience and optimize inventory management. AI and ML can be used to identify customer buying patterns and predict future demand, allowing businesses to better manage their inventory and stock the right products at the right time. Additionally, AI and ML can be used to create personalized shopping experiences for customers, such as personalized product recommendations and customized discounts. |
| OpenAI | Grocery businesses can benefit from using technology to automate back-end processes such as inventory management, order fulfillment, and customer service. Automation can help streamline operations, improve accuracy and reduce costs. Additionally, leveraging data analytics can help grocery businesses gain valuable insights into customer preferences and buying behaviors, enabling them to better target and tailor their offerings. |
| OpenAI | Grocery businesses can benefit from using technology that can help them better manage their inventory. By using an automated inventory system, grocery businesses can track what items are selling quickly, what items are not selling, and what items need to be reordered. This can help them save time, money, and resources as they can more accurately anticipate customer demand. |
| AI-free | New idea/fact/insight related to the grocery business is to bring back a "new" retail store to "do business locally" that had been lost (and not in a very nice way). But wait! There is another way to bring in store-lifters and bring them locally... The local grocery store "lifts money" from customers via online store credit, with coupons or other offers to "use" their cash at local stores or grocery stores. (See www.pennagas.com) Some of their current customers use a business credit card or card account, and local stores often offer free access to their customers. So these customers can give an extra bit of money to someone for the sake of being able to "use" their cash when their shopping gets out of hand. This method of "lifting money" to local stores, using the customer's credit card, or using something like cash "as a gift," is called a "local |

Рисунок 4.1. Демонстрація генерації системою нових ідей

На рисунку вище можна побачити, що користувач вводить «grocery business», що є запитом для генерації нових ідей. Потім користувач вибирає зі списку типів інформації лише OpenAI (модель GPT-3) і AI-free (GPT-2 від Hugging Face). Також користувач задає кількість результатів, які він хоче побачити за кожним із типів. Далі, при натисканні кнопки пошуку, система через деякий час повертає результати у вигляді ідей, які були згенеровані моделями ГН.

Якщо ж аналізувати інформацію на рисунку 4.1, що видають моделі, то вона містить згенеровані ідеї, факти, інсайти стосовно запита користувача. Таким чином результати можуть надати нову незнайому користувачу інформацію, що сприятиме винайденню нових ідей або це можуть бути вже готові ідеї. Точність чи якість результатів може бути покращена, надаючи більш детальні та коректні вхідні prompts для моделей.

Не дивлячись на активний розвиток моделей штучного інтелекту, одним з найкращих генераторів ідей є інтелект людини. Саме тому в систему було вирішено реалізувати можливість отримувати не лише нові ідеї шляхом штучного інтелекту, але й побачити ті, що є реалізованими зараз. На рисунку 4.2 показано, яким чином система здійснює пошук існуючих ідей.

Ideation Tool

Details

```

{
  "details": {
    "date": "2022-09-20",
    "number": "11449522",
    "summary": "Sensor data (or IoT) analytics plays a critical role in taking business decisions for various entities (e.g., organizations, project ..."
  },
  "main_info": "Systems and methods for recommending execution environment for analysing sensor observational data",
  "type": "PATENTS"
}

```

Enter your query:

Arxiv
 News
 Patents
 OpenAI
 AI-free
 Trends
 Number of results per type:

| Type | Main info |
|---------|---|
| NEWS | Chedraui USA to open 5 new supermarkets, massive distribution ... - Winsight Grocery Business |
| ARXIV | Large Scale Business Discovery from Street Level Imagery |
| ARXIV | Fine-Grained Product Class Recognition for Assisted Shopping |
| ARXIV | The Elephant in the Room: Analyzing the Presence of Big Tech in Natural Language Processing Research |
| PATENTS | Systems and methods for recommending execution environment for analysing sensor observational data |
| PATENTS | Method and system for bundling automotive data |
| ARXIV | Learning Personalized Page Content Ranking Using Customer Representation |
| ARXIV | Learning to Personalize Recommendation based on Customers' Shopping Intents |
| NEWS | Lidl US to open new store in Manhattan - Winsight Grocery Business |
| GTrends | ["Market-Economics", "Customer-Topic", "Wholesaling-Topic", "Management-Topic", "Supermarket-Topic", "Marketing-Topic"] |
| ARXIV | The complete set of efficient vectors for a reciprocal matrix |
| NEWS | Ahold Delhaize delivers solid Q1 results, driven by its strong U.S. ... - Winsight Grocery Business |
| ARXIV | On the Time-Varying Structure of the Arbitrage Pricing Theory using the Japanese Sector Indices |
| ARXIV | Levy distribution and long correlation times in supermarket sales |
| PATENTS | Storing entries as ordered linked lists |

Рисунок 4.2. Демонстрація пошуку системою існуючих ідей

З рисунку вище можна побачити, що за запитом користувача «grocery business» програмний застосунок надає різноманітну інформацію по існуючим

ідеям. Для цього необхідно обрати усі типи інформації окрім OpenAI і AI-free. Як було зазначено у попередньому розділі, інформаційними ресурсами для пошуку існуючих ідей є:

- статті з arXiv, в яких можна знайти реалізовані ідеї іншими людьми, а також більш детальну інформацію про тему суб'єкта для нових ідей і можливостей;
- новини пов'язані з нещодавніми подіями на тему запитуваного суб'єкта, що дозволяє рекомендувати та підказувати людині нові можливості і перспективи в галузі;
- патенти, які дозволяють виявити нові напрямки розвитку в галузі, а також зрозуміти, які ідеї вже є закріпленими, проте які можна розвинути в іншому руслі;
- тренди по заданому запиту, які отримані через API до Google Trends, що дозволяє відстежувати останні тенденції та зміни в галузі.

Користувач може натиснути на кожне із посилань, і сайт поверне йому результат у JSON форматі, де міститься інформація про різні деталі, такі як дата публікації, номер і короткий опис, а також головна інформація і тип посилання.

На рис. 4.3 показаний приклад запита-відповіді при взаємодії фронтенда на Vue.js з Flask бекендом.

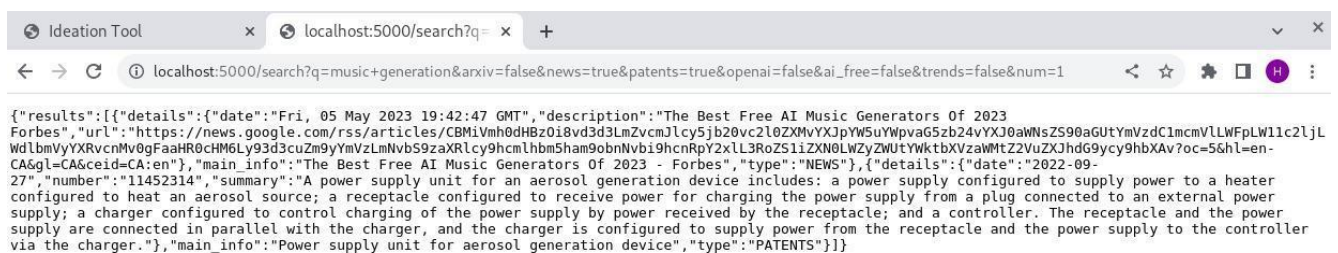


Рисунок 4.3. API запит-відповідь.

4.2. Можливі покращення системи

Наявність функціоналу повертання API дає можливість покращити систему шляхом її інтегрування з іншими системами або сервісами. API можна

використовувати для отримання доступу до даних, які генерує програма, і використовувати їх в інших програмах або на сайтах.

Наприклад, можна задеплоїти інфраструктуру у вигляді Telegram Bot API з subscription based моделлю, який дозволяє зв'язати Telegram бота з сервером та отримувати/відправляти повідомлення до/від користувачів. При реалізації бота користувачі зможуть підписатись на нього для отримання щоденної свіжої інформації по заданій темі. Якщо певна інформація сподобалась користувачам, вони можуть зберегти її у спеціальному списку, та відправляти її по електронній пошті колегам або друзям. Також можна додати категорії ідей, щоб бот збирав якомога більше інформації, що цікавить користувачів.

Для покращення системи можна запропонувати користувачам семантично протилежну інформацію, що допоможе уникнути традиційних думок та стереотипів, які можуть обмежувати творчість. Щоб реалізувати подібне можна використовувати векторну різницю та моделі ГН по типу glove, bert, word2vec, які допоможуть отримувати фрази чи речення, що відрізнялися б по вибраній метриці від вхідних.

Кластеризація та узагальнення інформації можуть допомогти в структуруванні та організації ідей. Це можна зробити за допомогою алгоритмів навчання без вчителя (кластеризації), які дозволяють групувати схожі ідеї разом, а також методів самеризації, які дозволяють відбирати найбільш релевантну та значиму інформацію.

Збереження результатів, цікавої інформації та нотатків користувача може допомогти зберегти та організувати зібрану інформацію, що дозволить легко повертатися до неї в майбутньому. Для цього можна використовувати бази даних або спеціальні сервіси для зберігання та організації інформації, такі як hyper index або dag graph.

Для вирішення проблеми блокування IP та деплойменту на більшу кількість користувачів потрібно використовувати авторизацію, проксі-сервери та хмарні рішення, які зможуть ефективно забезпечити доступ до необхідної інформації.

ВИСНОВКИ

Отриманий застосунок є результатом дослідження та розробки з метою надання користувачам цінного інструменту для пошуку натхнення, нових ідей та інформації, необхідних для технічних бізнес проєктів чи творчих завдань. Він поєднує в собі потужність штучного інтелекту з інтуїцією та креативністю людини для отримання ефективного рішення по створенню ідей.

В результаті виконання кваліфікаційної роботи на здобуття ступеня магістра були реалізовані усі поставлені задачі для досягнення зазначеної мети, а саме:

1. Було визначено можливість застосування методів штучного інтелекту для створення ідей. Подібна можливість досягається в основному за допомогою моделей-трансформерів глибинного навчання.

2. Були вивчені актуальні наукові праці та розробки в галузі створення ідей з використанням засобів штучного інтелекту. Схожими по призначенню до розробленого застосунку для створення ідей є такі ресурси як InfraNodus та IdeasAI.

3. В ході аналізу і усунення недоліків вищезазначених додатків було розроблено власну концепцію застосунку для створення ідей. Вона полягає у використанні засобів ШІ для генерації ідей/фактів/інсайтів, враховуючи запити користувача, але не покладається лише на ШІ, даючи можливість інтелекту людини прийти до потрібної ідеї. Це досягається через “розумний” пошук, що надає доступ до патентів, трендів, новин, наукових публікацій, семантично схожих фраз.

4. По вищезазначеній концепції було спроектовано веб-додаток, архітектура якого складається з фронтенд частини на Vue.JS та бекенду на Flask, що використовує моделі ГН, NLP підходи, API та обробку Google Trends/News, arXiv, USPTO, OpenAI, тощо.

5. Після цього було розроблено і продемонстровано роботу застосунку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Russel S. Artificial intelligence—a modern approach 3rd Edition / S. Russel, P. Norvig. // The Knowledge Engineering Review. – 2012. <https://doi.org/10.1017/S0269888900007724>
2. Creative idea generation method based on deep learning technology / [T. Zhao, J. Yang, H. Zhang та ін.]. // Int J Technol Des Educ. – 2021. – №31. – С. 421–440. <https://doi.org/10.1007/s10798-019-09556-y>
3. Deng L. Artificial Intelligence in the Rising Wave of Deep Learning: The Historical Path and Future Outlook [Perspectives] / Deng. // IEEE Signal Processing Magazine. – 2018. – №35. – С. 180–177. doi: 10.1109/MSP.2017.2762725.
4. Turing M. A. Computing Machinery and Intelligence / Alan Turing M. // Mind LIX. – 1950. – №236. – С. 433–460.
5. Babbage C. Of the Analytical Engine / Charles Babbage. // Passages from the Life of a Philosopher. – 1864. – №3. – С. 112–141.
6. Federico Menabrea L. Sketch of The Analytical Engine Invented by Charles Babbage / L. Federico Menabrea, A. Lovelace. // Scientific Memoirs. – 1843. – №3. – С. 666–731.
7. Cohen H. How to Draw Three People in a Botanical Garden / Harold Cohen. – 1988. – С. 846–855.
8. Cope D. Experiments in Musical Intelligence (EMI): Non-Linear Linguistic-Based Composition / David Cope. // Interface. – 1989. – №18. – С. 117–139.
9. Hofstadter R. D. The Copycat Project: A Model of Mental Fluidity and Analogy-Making / D. Hofstadter R., M. Mitchell. // Advances in connectionist and neural computation theory. – 1994. – №2. – С. 31–112.
10. Scientific Discovery: Computational Explorations of the Creative Process / P.Langley, H. Simon A., G. Bradshaw L., J. Zytkow M.. // The MIT Press. – 1987.
11. Meehan R. J. TALE-SPIN, an Interactive Program That Writes Stories / James Meehan R. – 1977. – №1. – С. 91–98.

12. Racter. *The Policeman's Beard Is Half Constructed* / Racter. – New York: Warner Books, Inc., 1984. – 129 c.
13. Amílcar C. *Converging on the Divergent: The History (and Future) of the International Joint Workshops in Computational Creativity* / C. Amílcar, T. Veale, G. Wiggins A. // *AI Magazine*. – 2009. – №30. – C. 15–22.
14. Colton S. *Computational Creativity: The Final Frontier?* / S. Colton, G. Wiggins A. – 2012. – №242. – C. 21–26.
15. Wiggins A. G. *Searching for Computational Creativity* / Geraint Wiggins A. // *New Generation Computing*. – 2006. – №24. – C. 209–222.
16. *Generative Deep Learning* – Sebastopol, CA: O'Reilly, 2019. – 330 c.
17. *Alias-Free Generative Adversarial Networks* / [T. Karras, M. Aittala, S. Laine та ін.]. // *Advances in Neural Information Processing Systems (Online)*. – 2021. – №34. – C. 852–863.
18. *Neural Speech Synthesis with Transformer Network* / [N. Li, S. Liu, Y. Liu та ін.]. – 2019.
19. *Highly Accurate Protein Structure Prediction with AlphaFold* / [J. Jumper M., R. Evans, A. Pritzel та ін.]. // *Nature*. – 2021. – №596. – C. 583–589.
20. *Beyond the Creative Species* / Oliver Bown. – Cambridge: The MIT Press, 2021. – 416 c.
21. *The Processes of Creative Thinking* / A. Newell, J. Shaw C., H. Simon A. // *Contemporary Approaches to Creative Thinking: A Symposium Held at the University of Colorado*. – 1962. – C. 63–119.
22. *Essential Barometer or Dated Concept?* / Dan Ventura. // *Proc. of the 7th International Conference on Computational Creativity (Paris, France)*. – 2016.
23. *Creating Creativity: 101 Definitions (what Webster Never Told You)* / A. Aleinikov G., S. Kackmeister, R. Koenig. – 2000.
24. *Creativity, Creative Thinking, and Critical Thinking: In Search of Definitions* / Donald Treffinger J. // *Center for Creative Learning*. – 1996.
25. *The Emotion Machine* / Marvin Minsky. – New York: Simon & Schuster, 2006. – 400 c.

26. Jordanous A. Four PPPerspectives on Computational Creativity in Theory and in Practice / Anna Jordanous. // *Connection Science*. – 2016. – №28. – C. 294–216.
27. Boden A. M. *The Creative Mind: Myths and Mechanisms* / Margaret Boden A. – London, UK: Routledge, 2003. – 360 c.
28. Macedo L. *Assessing Creativity: The Importance of Unexpected Novelty* / L. Macedo, A. Cardoso. // *Workshop on Creative Systems (Lyon, France)*. – 2002.
29. Gardner H. *Creating minds: An anatomy of creativity seen through the lives of freud* / H. Gardner. – New York: Basic Books, 1993. – 464 c.
30. Moran S. *The roles of creativity in society* / S. Moran. // *The Cambridge handbook of creativity*. – 2010. – C. 74–90.
31. Engell J. *The creative imagination: Enlightenment to romanticism* / J. Engell., 1982. – 440 c.
32. Simonton K. D. *Origins of genius: Darwinian perspectives on creativity* / D. Simonton K., 1999. – 320 c.
33. Gabora L. *Evolutionary approaches to creativity* / L. Gabora, S. Kaufman B. // *Cambridge handbook of creativity*. – 2010. – C. 279–300.
34. Aunger R. *Darwinizing Culture: The Status of Memetics as a Science* / R. Aunger., 2000. – 256 c.
35. D. Dennett. *Conditions of personhood*. In *What Is a Person?*, pages 145–167. Springer, 1988.
36. Dietrich A. *Brain mechanisms of creativity: What we know, what we don't* / A. Dietrich. // *Mobile Brain-Body Imaging and the Neuroscience of Art, Innovation and Creativity*. – 2019. – C. 23–28.
37. Kingma D. *Auto-Encoding Variational Bayes* / D. Kingma, M. Welling. // *International Conference on Learning Representations (Banff, Canada)*. – 2014. – №2.
38. Jimenez Rezende D. *Stochastic Backpropagation and Approximate Inference in Deep Generative Models* / D. Jimenez Rezende, S. Mohamed, D. Wierstra. – 2014. – №32. – C. 1278–1286.
39. Kingma D. *An Introduction to Variational Autoencoders* / D. Kingma, M. Welling. // *Foundations and Trends in Machine Learning*. – 2019. – №12. – C. 307–392.

40. An Introduction to Variational Methods for Graphical Models / M. Jordan, Z. Ghahramani, T. Jaakkola, L. Saul. // Machine Learning. – 1999. – №37. – С. 183–233.
41. Ganguly A. An Introduction to Variational Inference / A. Ganguly, S. Ermon. // arXiv. – 2021. – №2108. – С. 1–13.
42. Zhao S. Towards Deeper Understanding of Variational Autoencoding Models / S. Zhao, J. Song, S. Ermon. // arXiv. – 2017. – №1702. – С. 1–11.
43. A Contrastive Learning Approach for Training Variational Autoencoder Priors / J. Aneja, A. Schwing, J. Kautz, A. Vahdat. // Advances in Neural Information Processing Systems (Online). – 2021. – №34. – С. 480–493.
44. Burda Y. Importance Weighted Autoencoders / Y. Burda, R. Grosse, R. Salakhutdinov. – 2016. – №4.
45. Attention is All you Need / [A. Vaswani, N. Shazeer, N. Parmar та ін.]. // Advances in Neural Information Processing Systems (Long Beach, CA). – 2017. – №30.
46. On the Opportunities and Risks of Foundation Models / [R. Bommasani, D. Hudson, E. Adeli та ін.]. // arXiv. – 2021. – №2108. – С. 1–214.
47. Bahdanau D. Neural Machine Translation by Jointly Learning to Align and Translate / D. Bahdanau, K. Cho, Y. Bengio. – 2015. – №3.
48. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin, M. Chang, K. Lee, K. Toutanova. – 2019. – №1. – С. 4171–4186.
49. Language Models are Few-Shot Learners / [T. Brown, B. Mann, N. Ryder та ін.]. // Advances in Neural Information Processing Systems (Online). – 2020. – №33. – С. 1877–1901.
50. Paranyushkin D. InfraNodus [Електронний ресурс] / Dmitry Paranyushkin – Режим доступу до ресурсу: <https://infranodus.com/>.
51. OpenAI. IdeasAI [Електронний ресурс] / OpenAI – Режим доступу до ресурсу: <https://ideasai.com/>.
52. A thematic cluster analysis of parents' online discussions about fussy eating / [B. Markides, R. Laws, K. Hesketh та ін.]. // Matern Child Nutr. – 2022. – С. 1–12.

53. LSTM: A Search Space Odyssey / [K. Greff, K. Srivastava, J. Koutnik та ін.]. // IEEE Transactions on Neural Networks and Learning Systems. – 2017. – №28. – С. 2222–2232.

54. Long short-term memory [Електронний ресурс] // Wikipedia – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Long_short-term_memory.

55. Generative pre-trained transformer [Електронний ресурс] // Wikipedia – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Generative_pre-trained_transformer.

56. Aye G. Learning Autocompletion from Real-World Datasets / G. Aye, S. Kim, H. Li. // ResearchGate. – 2020. – С. 1–9. Режим доступу до ресурсу: https://www.researchgate.net/publication/345654307_Learning_Autocompletion_from_Real-World_Datasets

57. Sciforce. What is GPT-3, How Does It Work, and What Does It Actually Do? [Електронний ресурс] / Sciforce // Medium. – 2021. – Режим доступу до ресурсу: <https://medium.com/sciforce/what-is-gpt-3-how-does-it-work-and-what-does-it-actually-do-9f721d69e5c1>.