# Тема 13. Великі мовні моделі

Людство зі змінним успіхом намагається навчити машини говорити природною мовою та створити штучний інтелект починаючи з 1950-х років. Мова допомагає оперувати знаннями та обмінюватися інформацією, тому, не дивно, що моделювання мови стало активним напрямом у АІ.

Сучасний етап розвитку дисципліни пов'язаний із застосуванням нейронних мереж та глибокого навчання. З революцією глибокого навчання у 2010-х почалася епоха великих мовних моделей. Великі Мовні Моделі (Large Language Models) – це моделі машинного навчання, призначені для вирішення завдань обробки та розуміння природної мови. Вони навчаються на великих обсягах даних, мільярдах параметрів, досягаючи нового рівня якості в обробці природної мови.

Великі мовні моделі – це потужний інструмент, що вже показують визначні результати у вирішенні різноманітних завдань.

* Створення контенту. LLM дозволили машинам створювати зв'язаний та логічний контент, який можна використовувати для створення повідомлень у блогах, статей та інших форм контенту. Моделі використовують глибоке навчання, щоб зрозуміти та структурувати контент в унікальному та зручному для користувачів форматі.
* Розуміння, узагальнення та класифікація тексту. LLM забезпечують практичну основу для розуміння тексту та його контексту. Завдяки аналізу великих обсягів даних, LLM дозволяє розуміти, узагальнювати і класифікувати текст у різних формах та шаблонах.
* Відповідь на запитання. Великі мовні моделі дозволяють точно розуміти запити користувача природною мовою та відповідати на них. Вони аналізують контекст запиту та виконують пошук за великим масивом текстів, щоб знайти доречні відповіді на запити користувачів.
* Машинний переклад. LLM використовують для перекладу тексту між двома мовами. У моделях використовуються алгоритми глибокого навчання, такі як рекурентні нейронні мережі, для вивчення мовної структури вхідної та вихідної мов.
* Аналіз настроїв. Це використання великих мовних моделей, в якому модель навчена ідентифікувати та класифікувати емоційні стани та почуття у наданому тексті. Можуть виявлятися такі емоції, як позитивність, негативність, нейтральність та інші складні почуття, які можуть допомогти отримати уявлення про думки та відгуки клієнтів про різні продукти та послуги.

Можливості подібних систем виходять далеко за межі того, чого їх навчали, і навіть їхні розробники не можуть зрозуміти чому. Тести показують, що системи штучного інтелекту створюють моделі реального світу так само, як це робить людський мозок, тільки технологія у машин інша.

Багато інформації LLM витягують з текстів, що були надані для глибокого навчання. Наприклад, маючи описи кольорів, створюють свої внутрішні представлення про них. Коли модель бачить слово «червоний», вона сприймає його не просто як абстрактний символ, а як поняття, що стосується бордового, малинового, фуксії, іржі тощо. Продемонструвати це було досить складно, тому дослідники вивчили її реакцію на серію текстових запитів. Щоб перевірити, чи не є її представлення простим повторенням колірних зв’язків з онлайн-джерел, вони спробували ввести систему в оману і повідомити їй, що червоний насправді зелений. Система у відповідь не стала транслювати застарілу інформацію, а належним чином переглянула відповідність між кольорами та об'єктами.

Для здійснення самокорекції система шукає логіку, що лежить в основі її навчальних даних, тому, чим ширший потік цих даних, тим більше система виявляє в них загальні правила.

Термін «навчання» зазвичай є інтенсивним процесом, коли через нейронну мережу проганяють гігабайти даних і налаштовують її внутрішні зв'язки. LLM пізнають глибинні сенси і під час використання. Вони вдосконалюють свої знання, використовуючи підказки користувачів, така здатність відома як контекстне навчання. Контекстне навчання ґрунтується на тому ж алгоритмі, як і стандартне навчання.

Один із прикладів того, як навчається LLM, випливає із способу, яким люди взаємодіють із чат-ботами типу ChatGPT. Системі можна дати зрозуміти, яким чином вона має спілкуватися з користувачем і вона скорегує свої відповіді. Її репліки складаються з кількох тисяч слів, які вона бачила останніми. Як використовувати ці слова, формується її фіксованими внутрішніми зв'язками, але передбачається деяка варіативність. Цілі веб-сайти присвячуються промптам (prompts, підказкам, запитам), як подолати обмеження у відповідях («зламати систему»), що не дозволяють їй розповідати заборонену інформацію, наприклад, як зробити бомбу чи розкрити нюанси її алгоритмів. Хтось робить злам у корисливих цілях, хтось для отримання мудріших відповідей.

Інший тип контекстного навчання здійснюється за допомогою підказок з ланцюжка висновків. Мережу просять пояснювати кожен крок своїх висновків — така тактика дозволяє успішніше вирішувати логічні та арифметичні завдання, які потребують кількох кроків. Ця процедура не була запрограмована, система знайшла її самостійно.

У LLM ще досить багато сліпих плям, що не дозволяють кваліфікувати ці моделі як загальний штучний інтелект (рівень можливостей мозку живих істот). Водночас нові здібності мовних моделей дозволяють дослідникам припустити, що технологічні компанії ближчі до створення сильного інтелекту, ніж передбачали навіть оптимісти.

### Покоління великих мовних моделей

**Перше покоління** започаткувало векторне представлення слів. Оскільки ці моделі не використовували для подальших завдань обробки мови, вони зазвичай є дуже неглибокими з точки зору обчислювальної ефективності. Прикладами таких моделей є Skip-Gram та GloVe. Отримані вектори можуть передати семантичні значення слів, але не залежать від контексту і не вловлюють більш високорівневі концепції.

**Друге покоління** зосереджено на вивченні контекстних векторних представлень слів і покращенні моделей першого покоління. Прикладами таких моделей є CoVe, ELMo, OpenAI GPT та BERT.

**Третє покоління** породило моделі, що ґрунтуються на другому поколінні, зі збільшеною продуктивністю та зняттям деяких обмежень. Немає чіткого визначення цього покоління чи списку моделей, проте можна виділити такі характеристики:

* Покращене розуміння контексту, виявлення складних семантичних відносин.
* Інтеграція інформації з кількох джерел або модальностей, таких як текст, зображення та аудіо.
* Підвищення продуктивності, наприклад, за допомогою методів стиснення моделей або ефективніших архітектур.
* Краще розуміння лінгвістичних та структурних властивостей вхідних даних.

**Четверте покоління** моделей є найновішим. На додаток до основних здобутків третього покоління, додається таке:

* Більший обсяг даних для навчання та більша кількість параметрів дозволяє їм розуміти ширший спектр мовних конструкцій та нюансів.
* Покращені можливості розуміння та генерації текстів, що призводить до більш зв'язного та контекстуально точного створення відповіді.
* Вдосконалене точне налаштування та трансферне навчання. Можна збільшити продуктивність під конкретне завдання, наприклад, переклад, реферування чи відповіді на запитання.
* Збільшення масштабованості та ефективності.

На даний момент до четвертого покоління великих мовних моделей належить лише GPT-4 від компанії OpenAI, а найпоширенішими моделями третього покоління є BERT та LLama.

## Внутрішній устрій LLM

Великі мовні моделі є базовими моделями, які можна налаштовувати та адаптувати для вирішення широкого кола завдань. Вони мають деякі загальні характеристики: генеративні за своєю природою, використовують самостійне навчання та адаптуються до різних завдань.

Щоб надати бажаний результат на основі введеного запиту моделям надають великий обсяг навчальних даних. Ці дані не мають заздалегідь проставлених міток, що означає, що модель не знає, на що вона дивиться. Мовні моделі навчаються без вчителя, за допомогою техніки, яка називається навчанням із самоконтролем.

Для навчання моделей використовують нейронні мережі різних архітектур та типів навчання, що дозволяє їм обробляти дані з неймовірною точністю та швидкістю. Застосовується трансферне навчання, тобто знання, що отримані з одного завдання використовують для виконання іншого завдання. Наприклад, велика мовна модель, навчена перекладати з англійської на французьку, може використовувати отримані знання для перекладу з англійської на іспанську.

LLM можуть приймати різні вхідні дані, але в основному навчаються на текстових даних. Це може бути необроблений текст (наприклад, повідомлення в блогах, новинні статті та повідомлення в соціальних мережах), структуровані дані (таблиці, електронні таблиці або бази даних) та зображення (деякі моделі були розроблені для роботи з зображеннями шляхом перетворення зображення на текстовий опис).

Навчені великі мовні моделі можна використовувати для створення різних типів тексту: написання статей, отримання відповідей на запитання, короткого викладу книги, опис фільму або плану відпочинку на вихідні. Втім, інколи мовні моделі можуть демонструвати галюцинації (дуже впевнені та дивні відповіді, які просто не відповідають дійсності або виправдовуються його обмеженими даними), але з часом таких відповідей меншає.

Базові моделі можна віднести до "технологій загального призначення" (General-Purpose Technology). Більшість таких технологій відіграють роль “допоміжних”, оскільки відкривають нові можливості, але не пропонують остаточних рішень.

Для розробки великих моделей потрібні три компоненти:

* Потужне обладнання. Для навчання та роботи моделі потрібно багато високопродуктивних чіпів, зокрема, GPU та TPU.
* Інвестиції у R&D. R&D (Research and Development) — унікальні рішення, які є абсолютними інноваціями (їх ніхто раніше не робив). Подальший розвиток технологічних рішень та архітектур дозволить ефективніше задіяти “залізо” та прискорити навчання.
* Навчальні дані. Дані необхідно зібрати, що у ряді випадків є нетривіальним завданням. Якщо для навчання потрібні розмічені дані, то процес стає ще дорожчим і складнішим.

Через обсяг навчальних датасетів і кількість параметрів такі мовні моделі називають великими.

великий = {число параметрів, витрати на обчислення, кількість даних, вартість навчання}.

Потужне “залізо”, R&D та підготовка даних дорого коштують, наприклад, Microsoft вклала в OpenAI — спочатку некомерційну організацію — $1 млрд.

Дослідження показують, що вартість навчання залежить від кількості параметрів:

* $2,5–50 тис. – для моделей зі 110 мільйонами параметрів.
* $10–200 тис. – 340 мільйонів параметрів.
* $1,6 млн. – 1,5 мільярда параметрів.

Високі витрати на обчислення обумовлені устроєм глибокого навчання. Хоча зв'язок між кількістю параметрів, складністю моделі та обчислювальними потужностями до кінця не вивчений, сьогодні домінує принцип “більше – краще”. Із зростанням масштабу моделі без оптимізації та застосування більш продуктивних чіпів, витрати збільшуватимуться.

Мовна модель потрібна для розуміння та генерації контенту природною мовою. Вони бувають дискримінантні та породжувальні (генеративні).

* Дискримінантні моделі - статистичні моделі, що вирішують завдання класифікації даних.
* Породжувальні моделі - статистичні моделі, які засновані на аналізі самих даних і дозволяють створювати нові екземпляри даних.

Розвиток глибокого навчання призвів до використання різних нейронних мереж для вирішення задач обробки природної мови (NLP), включаючи згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN), нейронні мережі на основі графів (GNN) та трансформери (Transformer). Класичні підходи до завдання NLP залежать від дискретних, вручну створених функцій, тоді як нейромережні методи зазвичай використовують щільні вектори для неявного представлення синтаксичних чи семантичних аспектів мови.

Багато досліджень показало, що попередньо навчені моделі (Pre-Trained Models) на великих корпусах текстів можуть навчитися універсальним мовним представленням, які є корисні для наступних завдань і дозволяють уникнути навчання нової моделі з нуля. З розвитком обчислювальної потужності, появою глибоких моделей та постійним вдосконаленням архітектура мовних моделей просунулась від поверхневої до глибокої.

### Представлення знань у моделях

Для навчання моделі застосовуються великі масиви текстів. Сучасні моделі можуть навчатись на десятках терабайтів тексту. Це надає можливість проаналізувати настільки велику кількість інформації, що модель отримує у певному сенсі універсальне знання практично про все. При цьому варто враховувати, що знання залежать від того, на чому модель навчалася (це можуть бути не зовсім достовірні дані). Знання закодовані у вагових коефіцієнтах нейронної мережі, які формуються в процесі навчання на величезних масивах текстових даних.

Сама модель має дуже велику кількість параметрів: десятки, а іноді сотні мільярдів. За рахунок цього виходить «запам'ятати» всі стандартні конструкції великої кількості мов, включаючи мови програмування, сенс слів та термінів, стилі тексту та правила логічних висновків.

* Модель навчається передбачати наступне слово в тексті на основі попередніх слів.
* Модель аналізує статистичні закономірності та взаємозв'язки між словами у цих текстах.
* Ці взаємозв'язки запам'ятовуються у вагах нейронної мережі як розподілених числових представлень слів та його контекстів.

Так формується "узагальнена пам'ять", що дозволяє моделі робити логічні висновки та генерувати нові формулювання на основі внутрішніх представлень мови. Знання у мовних моделях скоріше статистичні, а не символьні, що виведені з великих даних у процесі самонавчання. Це відрізняється від традиційних систем штучного інтелекту, де факти закладаються вручну розробниками.

LLM представляє кожне слово як точку у багатовимірному складному просторі. Кожному слову чи одиниці даних зіставляється вектор фіксованої довжини (наприклад, 200-300 чисел). Ці вектори кодують семантичні латентні властивості даних. Семантично близькі слова ("кіт", "собака") матимуть схожі вектори. А у різних за змістом слів ("кіт", "машина") вектори сильно відрізняються.

Таке представлення даних у векторах називається ембедінг (embedding) і широко застосовується у NLP та інших завданнях машинного навчання. Ці вектори використовують для ефективної обробки даних і вони дозволяють моделі виявляти приховані закономірності.

За рахунок навчання на величезному обсязі даних в моделі близькі за змістом слова стають близькими точками, і математичні операції над ними (порівняння близькості, додавання, усереднення тощо) починають мати практичний зміст. Це дозволяє моделі знаходити синоніми, порівнювати зміст текстів, перефразувати тексти. Таким чином, модель працює не з фактичними словами, а з їх сенсом.

LLM моделює взаємозв'язки між словами. При визначенні наступного слова його ймовірність буде залежати від сенсу всіх попередніх слів з врахуванням їхньої позиції в тексті. Цей механізм отримав назву Causal Self Attention, саме він дозволяє моделі розуміти сенс слів залежно від контексту їх використання.

Універсальна мовна модель потім часто донавчається під конкретне завдання. Завданням може бути діалог, відповіді на питання, доповнення чи редагування тексту, класифікація. Донавчання (Fine-Tuning) відбувається з використанням даних, що відображають специфіку кінцевої задачі.

#### Приклад, як саме LLM генерує розумне продовження.

Припустимо, модель почала свою відповідь із фрази «Штучний інтелект сьогодні перетворює». Як вона будує розумне речення далі?

Після сканування власної бази знань знаходяться всі випадки, де згадується фраза «Штучний інтелект сьогодні перетворює». Після цього можна взяти всі слова, які йшли за цим рядком, і вирахувати, з якою ймовірністю зустрічалося кожне з них. Наприклад, можна отримати наступний результат:

|  |  |  |
| --- | --- | --- |
| Штучний інтелект сьогодні перетворює | суспільство | 5.1% |
| науку | 4.2% |
| промисловість | 3.1% |
| медицину | 3,5% |
| всі сфери  | 7% |

На основі цих значень можна вибрати наступне слово. В даному випадку модель може додати до "Штучний інтелект сьогодні перетворює" слово "всі сфери". Але вона може вибрати й інше слово: це залежить від налаштувань моделі та конкретного завдання. Наприклад, для творчих завдань може обиратися слово з не максимальною ймовірністю. Це робить мову більш «живою» і оригінальнішою.

Наступним кроком можна повторити ці дії, використовуючи рядок із доданим новим словом – «Штучний інтелект сьогодні перетворює всі сфери» і обрати, припустимо «життєдіяльності». Завдання LLM — визначати, яким має бути наступне слово на основі тексту, поданого їй на вхід. Після додавання нового слова запускається пошук наступного слова — і так далі.

У прикладі вище схематично показано, як LLM працює на прикладі підбору наступних слів речення. Звісно, що в реальності LLM працює не зі словами, а зі змістом слів та речень.

## Навчання мовних моделей

Існують важливі кроки та методи, пов'язані з навчанням LLM, від підготовки даних та архітектури моделі до оптимізації та оцінки.

### Підготовка даних

**Формування даних для навчання.** Основою успішної мовної моделі є якість та кількість навчальних даних. Різноманітний та широкий набір даних дозволяє моделі вивчати мовні нюанси, узагальнений сенс та близькість понять. Джерелами даних можуть бути книги, статті, веб-сайти, соціальні мережі та інші репозиторії з великою кількістю тексту.

**Попередня обробка.** Перед навчанням навчальні дані повинні бути попередньо оброблені, текст приводиться до структурованого вигляду, що забезпечує узгодженість форматів та підвищує продуктивність моделі.

* Токенізація - розділення тексту на дрібніші одиниці, такі як слова, частини слів або символи, яким привласнюються унікальні ідентифікатори. Дозволяє моделі працювати з текстом як послідовністю токенів.
* Приведення літер до нижнього регістру для уніфікації розпізнавання та порівняння.
* Видалення спеціальних символів та пунктуації - залишаються лише символи літер та цифри.
* Лематизація - приведення слів до початкової форми для усунення відмінностей за числами, часом, родами.
* Видалення стоп-слів - часті слова типу "а", "і", прийменники мало корисні для аналізу.
* Стемінг - обрізання закінчень для зведення до однієї основи різних форм одного слова.
* Видалення рідкісних і частих слів залишаються лише інформативні слова.
* Векторизація - представлення слів як числових векторів – ембендінги (embedding).

### Нейромережна архітектура

**Вибір відповідної нейронної мережі.** Вибір правильної архітектури нейромережі для обробки мовної моделі має вирішальне значення. Дослідники та розробники повинні ретельно враховувати вимоги завдання, рівень складності обробки та доступні ресурси.

Для побудови великих мовних моделей найчастіше використовуються такі архітектури нейронних мереж, кожна з яких має унікальні переваги та функції:

* Трансформери (Transformers) – найпопулярніша архітектура на даний момент, заснована на механізмі уваги. Використовується у BERT, GPT-3 та інших моделях.
* Рекурентні нейронні мережі (RNN) – добре працюють із послідовними даними. Використовуються у моделях типу ELMo. Нині витісняються трансформерами.
* Згорткові нейронні мережі (CNN) - застосовуються для отримання локальних ознак з тексту. Використовуються як допоміжні на певних етапах обробки.
* Рекурентні згорткові мережі (RCNN) – комбінують RNN та CNN, використовувалися в деяких ранніх моделях.
* Гібридні мережі – комбінація різних типів, наприклад RNN та трансформерів. Допомагає використати переваги різних підходів.

Найбільший прогрес у NLP за останні роки продемонстрували трансформери. Ця архітектура демонструє найкращі результати для більшості завдань.

**Налаштування параметрів нейронної мережі.** Параметри нейронної мережі, такі як кількість шарів і нейронів, прихованих шарів та механізмів уваги, мають велике значення для її ємності та продуктивності. Ці гіпер-параметри повинні бути налаштовані так, щоб забезпечити баланс між складністю та обчислювальною ефективністю, уникаючи при цьому перенавчання.

### Навчальний процес

Для навчання великих мовних моделей застосовують три поширені моделі навчання:

* Навчання без прикладів. Базові LLM можуть розуміти широкий спектр запитів без спеціального навчання, часто з допомогою підказок, хоча точність відповідей різниться.
* Навчання у кілька прийомів. Якщо навести кілька відповідних прикладів навчання, можна значно підвищити продуктивність базової моделі у конкретній галузі.
* Точне налаштування. Це розширення навчання у кілька прийомів, під час якої фахівці з аналізу даних навчають базову модель коригувати її параметри з допомогою додаткових даних, які стосуються конкретного застосування.

Принцип роботи таких моделей досить простий:

* Модель отримує на вхід промпт (запит від користувача або набір слів) і далі підбирає наступне слово. Промпт (prompt, підказка) у цьому контексті — головний елемент управління. Генерація тексту відбувається саме на основі вхідного запиту, тому, змінюючи та оптимізуючи промпт, можна покращувати результат роботи моделі.
* Після цього отриманий рядок знову подається на вхід моделі, і вона підбирає ще одне слово.
* І так далі.

Так виходить "розумне продовження" на основі початкового запиту. Для користувача це виглядає як відповідь, яка має сенс.

### Розширення навчання за допомогою підказок

Застосування підказок (prompt) для навчання великих мовних моделей — це стратегія, коли модель навчається з врахуванням конкретних текстових входів, які надаються їй як зразки. Це може бути використане як для навчання моделі, так і для її подальшого використання у генерації тексту або вирішення інших завдань.

Загальний процес використання prompt для навчання великих мовних моделей:

* Підготовка підказок. Створення текстових входів, які будуть використані як підказки для моделі. Формулювання підказок так, щоб вони відповідали конкретній задачі або контексту використання.
* Вибір моделі. Вибір конкретної мовної моделі для навчання. Це може бути, наприклад, GPT (Generative Pre-trained Transformer) або подібні моделі.
* Навчання моделі. Подання підказок моделі у процесі навчання. Важливо врахувати, що великі мовні моделі зазвичай зазвичай навчаються на великих обсягах текстових даних, і навчання з використанням підказок може бути додатковим етапом.
* Налаштування гіперпараметрів. Здійснює налаштування гіперпараметрів моделі в залежності від конкретних вимог задачі.
* Оцінка результатів. Оцінка результатів роботи моделі із використанням підказок на тестових даних. Аналіз якості генерації тексту чи вирішення конкретної задачі.
* Доопрацювання та ітерації. За необхідності внесення корекцій до підказок або параметрів моделі та повторення процесу навчання та оцінки.

Застосування підказок може суттєво впливати на результати роботи моделі, і вибір підказок є важливим аспектом цієї стратегії. Також важливо стежити за інтерпретованістю та етичними аспектами, особливо при використанні моделей для створення контенту або взаємодії з користувачами.

Для ефективного підбору підказок для роботи з конкретним завданням або контекстом використовують наступні підходи:

* Аналіз завдання. Вивчення характеристик та вимог конкретного завдання. Визначення типів запитань і типи потрібних відповідей.
* Формулювання чітких та конкретних питань. Точне формулювання питання, щоб отримати потрібну інформацію від моделі.
* Варіативність підказок. Використання різноманітних підказок, щоб отримати різні відповіді та врахувати можливі спотворення даних.
* Ітеративний підхід. При використанні різних питань та підказок, аналізуються результати та коригуються підходи на основі отриманих відповідей.
* Експериментування та оцінка результатів. Оцінювання відповідей моделі на основі їх відповідності очікуванням та вимогам завдання.

Важливо пам'ятати, що підбір ефективних підказок може бути ітеративним процесом, що вимагає тестування та експериментування.

**Оптимізація швидкості навчання.** Швидкість навчання є важливим гіперпараметром, який контролює швидкість адаптації моделі під час навчання. Вибір потрібної швидкості навчання може значно вплинути на продуктивність моделі. Для прискорення процесу навчання великих мовних моделей можна використовувати такі методи:

* Попереднє навчання (Pre-Training) - спочатку навчити модель на великому корпусі даних, а потім додатково лише на цільових даних.
* Квантування вагових коефіцієнтів - зменшення розрядності вагових коефіцієнтів для економії пам'яті та прискорення обчислень.
* Розподілена обробка - розпаралелювання навчання на кількох GPU та серверах для прискорення.
* Трансфер навчання - використання вагових коефіцієнтів попередньої моделі для ініціалізації вагових коефіцієнтів нової моделі.
* Вибіркове навчання – оновлення лише частини вагових коефіцієнтів на кожному кроці навчання.
* Збільшення розміру пакету навчальних даних - великі пакети підвищують ефективність розпаралелювання.
* Зменшення точності обчислень – використання 16/8 біт замість 32 біт для операцій.

Підбір та комбінація цих методів дозволяє значно прискорити навчання без втрати якості моделі.

**Перенавчання і регуляризація.** Перенавчання виникає, коли модель занадто добре вивчає навчальні дані, що ставить під загрозу її здатність узагальнювати невидимі дані. Методи регуляризації, такі як відсіювання, зміна вагових коефіцієнтів та рання зупинка, можуть використовуватися для запобігання перенавчання та покращення можливостей узагальнення моделі.

### Оцінка продуктивності моделі

Метрики для оцінки LLM. Для оцінки ефективності та якості мовних моделей використовуються різні метрики:

* Точність (Accuracy) – частка правильних відповідей моделі на тестовому наборі даних. Застосовується для завдань класифікації.
* F-мера (F1 Score) - усереднена міра точності та повноти для завдань класифікації.
* Перплексія (Perplexity) – показник, наскільки добре розподіл ймовірностей або ймовірнісна модель передбачає зразок. Її можна використовувати для порівняння імовірнісних моделей. Низька перплексія вказує на те, що розподіл ймовірностей добре передбачає вибірку.
* BLEU (Bilingual Evaluation Understudy) – це вимір відмінностей між автоматичним перекладом та еталонним перекладом текстом, що виконано людиною. Чим вище, тим краще.
* ROUGE (Recall-Oriented Understudy) – це набір показників та програмний пакет, який використовується для оцінки автоматичних програм автореферування та машинного перекладу при обробці природної мови. Заснована на схожості граматичних конструкцій.
* CIDEr (Consensus-based Image Description Evaluation) - це важливий інструмент для тестування та покращення алгоритмів генерації текстових описів у завданнях комп'ютерного зору та обробки природної мови. Оцінює схожість сенсу згенерованого та еталонного тексту.
* Час виконання – наскільки швидко модель обробляє запити.
* Обсяг обчислювальних ресурсів - кількість дискової та оперативної пам'яті, обчислювальних ядер тощо.

Вибір конкретної метрики залежить від вирішуваного завдання і важливих аспектів якості для неї. Наприклад, для діалогових систем важлива природність.

Порівняння зі стандартами. Бенчмарк (benchmark) - це вимірюваний стандарт або точка відліку, що використовується для порівняння та оцінки продуктивності, якості або ефективності системи, пристрою, алгоритму або іншого об'єкта. GLUE, SuperGLUE та SQuAD – це популярні бенчмарки, що використовуються для оцінки якості мовних моделей, наскільки добре вона розуміє мову та різні аспекти, пов'язані з нею.

* GLUE (General Language Understanding Evaluation) - набір із 9 різних завдань NLP для оцінки здатності моделі розуміти мову. Містить завдання типу визначення тональності тексту, пошуку логічних невідповідностей тощо.
* SuperGLUE - розширена версія GLUE, містить складніші завдання, що вимагають логічного виведення. Призначений для оцінки моделей "на крок попереду" щодо GLUE.
* SQuAD (Stanford Question Answering Dataset) – популярний набір даних для тестування здатності моделі відповідати на запитання по тексту. Містить питання, контексти та еталонні відповіді.

Еталонні набори даних GLUE, SuperGLUE та SQuAD надають стандартизовані оціночні платформи для порівняння продуктивності різних великих мовних моделей. Ці набори даних охоплюють широкий спектр завдань, дозволяючи дослідникам оцінювати можливості своїх моделей та визначати області для покращення.

Навчання великих мовних моделей - це складний процес, що вимагає пильної уваги до деталей і глибокого розуміння методів, що лежать в його основі. Ретельно відбираючи та обробляючи дані, вибираючи відповідну архітектуру моделі, оптимізуючи процес навчання та оцінюючи продуктивність з використанням відповідних показників та контрольних показників, дослідники та розробники можуть постійно вдосконалювати та розширювати можливості LLM.

Важливість ефективних методів навчання для LLM буде лише зростати. Засвоївши ці важливі кроки, можна використати справжній потенціал мовних моделей, відкриваючи нову еру додатків та рішень, які трансформують напрямок обробки природної мови.

## Архітектура нейронної мережі Трансформер

Концепцію нейронної мережі Трансформер представлено в 2017 році. Вона набула широкого застосування в галузі обробки природної мови (NLP). Вона ґрунтується на ідеї використання механізму уваги для ефективного моделювання довгострокових залежностей у послідовних даних, таких як речення чи тексти.

Основні компоненти нейронної мережі трансформера - це енкодер та декодер.



Архітектура нейронної мережі Трансформер

### 1. Енкодер

* Вхідні дані, наприклад речення, розділяються на послідовність векторних представлень - ембедінги (Embeddings, вкладення).
* Ембедінги проходять через стек "шарів уваги" (Attention Layers). Кожен шар уваги складається з двох основних підшарів:
	+ Механізм самоуваги (Self-Attention Mechanism) дозволяє енкодеру зосередитися на різних частинах вхідної послідовності, враховуючи залежність між елементами. Механізм враховує контекст із попередніх кроків обробки.
	+ Повнозв'язана нейронна мережа приймає виходи з механізму уваги і застосовує операції лінійного перетворення для генерації нових ознак.
* Повторення шарів уваги та повнозв'язаних мереж дозволяє моделювати залежності на різних рівнях абстракції та виявляти локальні і глобальні залежності у вхідних даних.
* На виході енкодера утворюється послідовність векторних представлень, що містить контекстуальну інформацію про вхідні дані.

### 2. Декодер:

* Декодер має структуру, подібну до енкодера, але включає додатковий шар уваги, який називається "множинна увага" (Multi-Head Attention).
* Вхідні дані для декодера – це цільова послідовність, яку потрібно згенерувати (наприклад, переклад іншою мовою).
* Подібно до енкодера, декодер використовує механізм самоуваги та пов'язані мережі для обробки послідовності вхідних даних.
* Декодер звертається до виходів енкодера, щоб використовувати контекстуальну інформацію про вхідні дані.
* Основна відмінність декодера від енкодера - це використання множинної уваги, яка дозволяє декодеру "зосередитися" на різних частинах вхідної послідовності та генерувати відповідні вихідні дані.

Навчання нейронної мережі трансформера включає оптимізацію параметрів моделі, щоб мінімізувати помилку між передбаченими і правильними значеннями. Це зазвичай виконується з використанням алгоритму зворотного поширення похибки (BackPropagation) та градієнтного спуску.

Трансформери з їх потужним механізмом уваги та спосіб для моделювання довгих залежностей стали одним із найбільш успішних підходів у NLP. Вони дозволяють обробляти і генерувати тексти з високою точністю і збереженням контексту, що призвело до їх широкого застосування в завданнях машинного перекладу, генерації тексту, діалогових системах та інших NLP-завданнях.

### Навчання нейронної мережі трансформер включає такі кроки:

* Підготовка даних для навчання. Це може включати попередню обробку тексту, таку як токенізацію (поділ тексту на окремі токени), перетворення слів у числові вектори (ембедінги), створення механізмів уваги та розділення даних на навчальну, перевірочну та тестову вибірки.
* Архітектура моделі. Визначається архітектура нейронної мережі трансформер, включаючи кількість шарів енкодера та декодера, кількість механізмі уваги, розмірність векторів (ембендінгів) та прихованих шарів та інші гіперпараметри.
* Ініціалізація параметрів. Параметри моделі ініціалізуються випадковими значеннями або за допомогою попередньо визначених ваг.
* Прямий прохід. Вхідні дані проходять через енкодер, де відбувається обчислення уваги між токенами та генерація контекстуальних представлень. Потім ці представлення передаються до декодеру, який генерує вихідну послідовність.
* Обчислення функції втрат. Порівнюються передбачення моделі з цільовими значеннями і обчислюється функція втрат, така як перехресна ентропія. Мета полягає у мінімізації втрат.
* Зворотне поширення похибки. Градієнти функції втрат обчислюються по відношенню до параметрів моделі за допомогою зворотного поширення похибки. Градієнти передаються назад через енкодер та декодер для оновлення ваг.
* Оновлення параметрів. Оновлення параметрів моделі відбувається з використанням оптимізаційного алгоритму, наприклад стохастичного градієнтного спуску (SGD) або його варіантів. Градієнти, що обчислені на попередньому кроці використовуються для оновлення ваги, щоб поліпшити передбачення моделі.
* Ітерації навчання. Процеси прямого поширення, обчислення втрат, зворотного поширення похибки та оновлення параметрів повторюються для кожного блока даних доки не будуть пройдені всі навчальні дані. Цей процес повторюється протягом кількох епох.
* Оцінка моделі. Після завершення навчання модель може бути оцінена на перевірочній вибірці для оцінки її якості та налаштування гіперпараметрів.
* Тестування моделі. Нарешті, модель може бути протестована на тестовій вибірці, яка не використовувалась у процесі навчання та оцінки. Це дозволяє оцінити узагальнюючу здатність моделі нових даних.

Навчання нейронної мережі трансформер вимагає великого обсягу обчислювальних ресурсів та даних, і зазвичай виконується на графічних процесорах (GPU) або розподілених обчислювальних кластерах.

Завдяки механізму уваги трансформери ефективно моделюють глобальний контекст на вході та виході. Це дозволяє їм досягати високої якості у завданнях обробки природної мови.

### Механізм уваги у трансформерах.

Механізми уваги (Self-Attention) в нейронній мережі Трансформер дозволяє моделі ефективно аналізувати та враховувати залежності між словами у реченні. Розглянемо приклад: є наступне речення: "Я люблю читати книги." Важливо розуміти, що слово "читати" у цьому контексті пов'язане з "книгою", і контекст змінює зміст фрази. Механізм уваги дозволяє моделі автоматично виявити такі залежності.

**1. Ембедінг слів**. Спочатку кожне слово речення (наприклад, "Я", "люблю", "читати", "книги") векторизується і перетворюється на числове представлення. Енкодер перетворює вхідні дані на набір ключів (Keys) та значень (Values). Енкодер створить такі ключі та значення:

|  |  |
| --- | --- |
| **ключі (Keys)** | **значення (Values)** |
| Я | [0.1, 0.3, 0.8, ...]  |
| Люблю | [0.2, 0.2, 0.7, ...]  |
| Читати | [0.8, 0.9, 0.1, ...] |
| Книги | [0.3, 0.5, 0.8, ...] |

**2. Обчислення уваги.** Запит (Query) визначає, на які ключі потрібно звернути увагу під час генерації виходу. Для кожного слова в реченні модель розглядає всі інші слова та визначає, наскільки кожне з них є важливим для даного слова. Це оцінюється з використанням ваг, які визначають важливість кожного слова в контексті. Важливі слова одержують більші ваги, а несуттєві слова – менші ваги.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Запит (Query)** | Я | Люблю | Читати | Книги |
| Я | 1 | 0,9 | 0,5  | 0,1 |
| Люблю | 0,9  | 1 | 0,4 | 0,4 |
| Читати | 0,5  | 0,8  | 1 | 0,9 |
| книги | 0,2  | 0,1 | 0,9 | 1 |

Далі обчислюється зважена сума (увага) кожного слова до решти слів. Наприклад, для слова "читати" у реченні "Я люблю читати книги", ваги можуть бути високими для "книги" та "люблю", оскільки ці слова важливі для розуміння контексту слова "читати".

**3. Агрегація контексту**. За допомогою ваг, отриманих на попередньому кроці, модель агрегує контекст з інших слів у реченні. У разі слова "читати" модель об'єднує інформацію зі слів "книги", "люблю" і "Я", щоб врахувати контекст і зрозуміти, що це слово пов'язане з актом читання, а не, наприклад, з іншими значеннями слова "читати" .

Цей механізм дозволяє нейронній мережі Transformer ефективно враховувати залежності та контекст у тексті, що робить її потужним інструментом для завдань обробки природної мови, таких як машинний переклад та аналіз тексту.

Цікаво, що якщо слова трохи змінені, матриця уваги також змінюється. Наприклад, у реченні «Лампу не можна запакувати у валізу, тому що вона завелика» слово «вона» відноситься до лампи. Але в реченні «Лампу не можна запакувати у валізу, тому що вона замала». Слово «вона» відноситься до валізи. Отже, єдиний спосіб отримати ці матриці уваги – це вчитися на даних.

Трансформер має кілька переваг у порівнянні з іншими архітектурами нейронних мереж, такими як нейронні рекурентні мережі (РНС).

* Трансформери не вимагають послідовної обробки даних. Це означає, що трансформери можуть обробляти довгі послідовності даних набагато швидше.
* Трансформери більш стійкі до помилок. Краще працюють з даними, які містять певні пропущені фрагменти або помилки.

Трансформери були застосовані до широкого спектру завдань обробки природної мови, включаючи:

* Машинний переклад. Трансформери значно покращили якість машинного перекладу, порівняно з попередніми методами.
* Автоматичне реферування. Трансформери можуть бути використані для автоматичного створення рефератів з довгих текстових документів.
* Питання-відповідь. Трансформери можуть використовуватися для створення питання-відповідей систем, які можуть відповідати на питання про складні теми.

Трансформери є потужними інструментами для обробки природної мови, і вони продовжують розвиватися та вдосконалюватись.

## Популярні великі мовні моделі

На сьогоднішній день лідерами у галузі розробки та використання великих мовних моделей є:

### Google АІ - лідер у галузі розробки великих мовних моделей

Google використовує кілька великих мовних моделей для своєї пошукової системи.

* LaMDA навчена на величезному наборі даних тексту та коду, спроможна генерувати текст, перекладати мови, створювати різні види творчого контенту та відповідати на запитання інформативним чином.
* Turing NLG для створення природної мови, такої як листи, код, сценарії, музичні твори, електронні листи, листи тощо. Turing NLG використовується для покращення якості результатів пошуку, для створення доречних сніпетів.
* PaLM 2 нова велика мовна модель, яка працюватиме з оновленим чат-інструментом Google Bard, і буде базовою моделлю для більшості нових функцій штучного інтелекту.

Google взялася за розробку нової пошукової системи зі штучним інтелектом, де пошуковик намагається передбачити питання користувача і запропонувати набагато більш персоналізований досвід. Google змінює спосіб подання результатів пошуку, додаючи чат зі штучним інтелектом, а також короткі відео та пости в соціальних мережах. Компанія відходить від традиційного списку результатів, який зробив його домінуючою пошуковою системою.

Нововведення є відповіддю на великі зміни у способах доступу людей до інформації в інтернеті, включаючи появу ботів зі штучним інтелектом, таких як ChatGPT. Google планує зробити свою пошукову систему більш «візуальною, зручною, особистою та людяною» з акцентом на обслуговування молодих людей у всьому світі.

Ці мовні моделі використовуються в різних продуктах Google:

* Google Search. Використовує Turing NLG для надання більш природних та інформативних результатів пошуку.
* Google Assistant. Використовує LaMDA для покращення відповідей на запитання користувачів.
* Google Translate. Використовує Turing NLG для покращення точності перекладу.
* Google Cloud Natural Language API. Надає Turing NLG та LaMDA як свої частини, що дозволяє розробникам використовувати їх у своїх власних програмах.
* Gmail. Використовує LaMDA для інтелектуальних відповідей на листи та підказки.
* Google Docs. Використовує LaMDA для розумних рекомендацій під час написання.
* YouTube. Використовує LaMDA для аналізу та модерації коментарів.

Google продовжує розробляти нові мовні моделі, і очікується, що вони відіграватимуть важливу роль у майбутніх продуктах та програмах Google.

У 2023 році Google оголосила про запуск PaLM 2, новітньої великої мовної моделі, яка працюватиме з оновленим чат-інструментом Google Bard, конкурентом ChatGPT від OpenAI, і служитиме базовою моделлю для більшості нових функцій штучного інтелекту.

PaLM 2 – це не окремий продукт, а сімейство продуктів з різними версіями, які будуть розгорнуті у споживчих і корпоративних умовах. PaLM 2 вже використовується для підтримки 25 функцій та продуктів, включаючи Bard, експериментальний чат-бот компанії.

Оновлення, доступні через Bard, включають покращені можливості кодування й розширену мовну підтримку. Він також застосовується для розширення можливостей таких програм Google Workspace, як Документи, Презентації та Таблиці.

PaLM 2 здатний розуміти ідіоми різними мовами. Рівень володіння мовою, достатній для навчання цієї мові.

Також Google змінює спосіб подання результатів пошуку, додаючи чат зі ШІ, а також короткі відео та пости в соціальних мережах. Компанія відходить від традиційного списку результатів, який зробив його домінуючою пошуковою системою.

### OpenAI

* GPT-3 - генеративна передбачена трансформаторна модель, що випущена у 2020 році та швидко стала однією з найпопулярніших мовних моделей. GPT-3 використовується в широкому спектрі продуктів, включаючи чат-боти, інструменти для створення контенту та навчальні програми.
* GPT-4 - потужніша версія GPT-3, що випущена в 2023 році. GPT-4 має ряд покращень, включаючи покращену здатність генерувати текст, схожий на людський, та покращену здатність перекладати мови.

Ці мовні моделі використовуються в різних продуктах та програмах, включаючи:

* ChatGPT – чат-бот, який використовує GPT-3 для генерації тексту, перекладу мов, написання різних видів творчого контенту та відповіді на запитання інформативно.
* Copy.ai – сервіс, який використовує GPT-3 для створення різноманітних творчих текстових форматів, таких як вірші, код, сценарії, музичні твори, електронні листи, листи тощо.
* Replika.ai – чат-бот, який використовує GPT-3 для створення реалістичних співрозмовників.
* AI Dungeon – гра, яка використовує GPT-3 для створення текстових пригод.

OpenAI продовжує розробляти нові мовні моделі, і очікується, що вони відіграватимуть все більш важливу роль у майбутньому продуктів та програм OpenAI.

### Інші великі компанії

Окрім OpenAI та Google, великі мовні моделі розробляють і інші компанії, зокрема:

* **Microsoft** розробила мовну модель Turing NLG, яка використовується у різних продуктах Microsoft, включаючи Bing та Cortana.
* **Facebook** розробила моделі, що використовуються у різних продуктах Facebook, включаючи Messenger та Instagram. Моделі RoBERTa, BART, XLM, BlenderBOT застосовують у соцмережах для модерації та рекомендацій.
* **IBM** розробила мовну модель Watson, яка використовується в різних продуктах IBM, включаючи Watson Assistant та Watson Translate.
* **DeepMind**, дочірня компанія Google AI розробила мовну модель Gopher у 2022 році. Модель має 280 мільярдів параметрів і є однією з найбільших мовних моделей у світі.

Ці компанії розробляють мовні моделі для різних цілей, включаючи:

* Чат-боти, що можуть підтримувати розмови з людьми.
* Інструменти для створення творчого контенту, такого як вірші, код, сценарії, музичні твори тощо.
* Навчальні програми, які можуть допомогти людям отримати нові знання та навички.
* Наукові дослідження у різних галузях, включаючи природну мову, машинне навчання та інформатику.

Ці компанії лідирують у галузі розробки великих мовних моделей з кількох причин. По-перше, вони мають доступ до великих обсягів даних, необхідні навчання цих моделей. По-друге, вони мають значні ресурси для розробки та підтримки цих моделей. По-третє, вони мають досвід у галузі машинного навчання та штучного інтелекту, які необхідні для розробки та використання великих мовних моделей.

Очікується, що конкуренція у галузі розробки великих мовних моделей продовжуватиме зростати в майбутньому. У міру розвитку технологій та збільшення обсягів даних, що використовуються для навчання цих моделей, очікується, що з'являться нові компанії, які зможуть конкурувати з лідерами галузі.

## Розвиток мовних моделей GPT

GPT (Generative Pre-trained Transformer) – це алгоритм обробки природної мови, випущений американською компанією OpenAI. Потужна мовна модель обробляється за допомогою нейронной мережі з архітектурою Трансформер, яка дозволяє знаходити зв'язки між окремими словами та обирати найбільш релевантну послідовність слів та речень. Головна особливість нейромережі полягає в її здатності запам'ятовувати та аналізувати інформацію, створюючи на її основі зв'язаний та логічний текст.

В основі моделі GPT знаходиться принцип автодоповнення – приблизно так працює опція Т9 у смартфонах. Спираючись на одне або кілька речень, алгоритм може читати, аналізувати та генерувати зв'язаний та послідовний текст на цю тему у необхідному обсязі. На сьогоднішній день GPT вважається найоб'ємнішою і найскладнішою мовною моделлю зі всіх існуючих.

Початок проекту покладено у 2017 році, коли дослідники групи Google Brain, що займається глибоким вивченням штучного інтелекту, представили модель обробки природної мови з архітектурою «трансформер». Генеративні мережі-трансформери створюють речення на задану тему з найбільш релевантних слів. Вони розподіляють їх у оптимальній послідовності, як це робить людина у мові. Трансформери виконують такі завдання швидше за інші типи мереж і залучають менший обсяг обчислювальних ресурсів.

У червні 2018 року компанія OpenAI опублікувала документ під назвою "Improving Language Understanding by Generative Pre-Training", в якому була описана модель GPT - генеративний попередньо навчений трансформер. У тому ж році розробники випустили першу повноцінну версію GPT-1.

### GPT-1

Мовну модель GPT-1 створено на основі «напівкерованого» підходу, що складається з двох етапів. У першому (неконтрольований генеративний етап попереднього навчання) моделювання використовується для встановлення початкових параметрів. У другому етапі (контрольований етап точної установки) ці параметри точково адаптуються до поставленої задачі.

Для навчання моделі GPT-1 було завантажено 4.5 Гб тексту з 7000 інтернет-сторінок та книг різних жанрів. Це забезпечило їй 112 млн. параметрів – змінних, що впливають на точність роботи алгоритму.

### GPT-2

Після успішного релізу першої версії фахівці з OpenAI розробили двоскеровану нейромережу BERT, яка вважалася найбільш досконалою мовною моделлю на той момент. Для створення моделі GPT-2 змінено принцип навчання. Розробники зрозуміли, що вчити модель на основі вибірки текстів із книг та «Вікіпедії» – не є найефективнішим способом, натомість вирішили використати звичайні пости та коментарі з інтернету.

У 2019 році команда OpenAI представила наступну версію своєї мовної моделі – GPT-2. Вона має таку саму архітектуру, як і GPT-1, але вже з модифікованою нормалізацією. Для навчання використано масив із 8 мільйонів документів та 45 мільйонів веб-сторінок, що містять 40 Гб тексту. Щоб зробити навчальні дані різноманітнішими, розробники взяли за основу сторінки інтернет-форумів, зокрема, пости користувачів Reddit з рейтингом вищим за середній. Це дозволило алгоритмам засвоювати лише корисний контент без спаму та флуду. У результаті GPT-2 набула 1,5 мільярда параметрів – це майже вдесятеро більше, ніж у попередній моделі.

### GPT-3

Реліз OpenAI GPT-3 відбувся у травні 2020 року. GPT-3 не отримала кардинальних змін своєї архітектури, проте для кращого масштабування її модифікували. У новій версії моделі розширився функціонал, що дозволило розробникам назвати її «придатною для вирішення будь-яких завдань англійською мовою». Доступ до GPT-3 був закритий для масового користувача.

GPT-3 здатна запам'ятовувати значно більше інформації, тому згенерований текст є більш логічним і зв'язаним. Для навчання мовної моделі задіяно суперкомп'ютер Microsoft Azure AI. До моделі завантажили майже 600 Гб тексту. У цей обсяг увійшла вся англомовна «Вікіпедія», художні книги з прозою та віршами, матеріали з GitHub та новинних сайтів, а також путівники та рецепти.

Потужність GPT-3 збільшив веб-архів Common Crawl з трильйоном слів. Архів Common Crawl - це велика колекція веб-даних. Common Crawl запускає процес сканування публічної доступної частини Інтернету, збирає інформацію про веб-сторінки та зберігає її у відкритому форматі даних. Архів Common Crawl складає мільярди веб-сторінок і містить інформацію, такі як HTML-код сторінок, текстовий вміст, посилання, метаданні та інші ресурси, доступні на веб-сайтах. Common Crawl надає вільний доступ до свого архіву, що дозволяє дослідникам та розробникам використовувати ці дані для різних цілей.

Частину датасету склали тексти іноземними мовами, що помітно покращило здатність моделі до перекладу. Третя версія мовної моделі має 175 мільярдів параметрів. Окрім генерації текстів, є низка інших варіантів застосування GPT-3: вона може відповідати на питання, проводити семантичний пошук та автореферування.

### ChatGPT

У листопаді 2022 OpenAI представила свій новий продукт - чат-бот ChatGPT, розроблений на основі генератора тексту GPT-3.5. Цю версію мовної моделі підготовлено спеціально для чат-бота: вона отримала ширші можливості та навчалася на свіжіших даних (до вересня 2021 року). Актуальність даних – важлива особливість і певною мірою нестача всіх версій GPT. При розробці нейромережі до неї завантажують дані з інтернету на певний момент часу. Через це вона нічого не знає про ті події, що сталися після вказаного періоду.

ChatGPT – це чат-бот із штучним інтелектом, який працює в діалоговому режимі. Він заснований на вдосконаленій версії мовної моделі GPT-3.5, для навчання якої використовувалися різні методи: з вчителем та підкріпленням. Програма здатна вести діалог у режимі реального часу, імітуючи людське спілкування. Чат-бот вміє писати тексти та сценарії створювати програмний код та виконувати його налагодження, складати вірші та створювати інші твори. Популярність ChatGPT завоював через можливість вести діалог і відповідати на різні запитання краще за середньостатистичної людини.

На відміну від попередніх моделей, ChatGPT навчався не лише за допомогою текстів, а й шляхом взаємодії з людиною. Над цим працювали спеціальні люди-тренери, які відтворювали моделі спілкування між користувачем та штучним інтелектом. Модель розвивалася на основі цих діалогів та завантажених у неї десятків гігабайт тексту. Тренери запитували щось у ChatGPT та оцінювали його відповіді. В залежності від отриманого результату підкріплювали відповіді моделі винагородами чи штрафами. У результаті чат-бот довго вчився і перевчався, коригуючи свої репліки на основі тренерських оцінок. Це дозволило досягти дуже високого ступеня «людяності» ChatGPT. Чат-бот може зберігати та аналізувати бесіди з користувачами, постійно вдосконалюючи таким чином свої здібності.

### GPT-4

14 березня 2023 OpenAI випустила нову версію мовної прогностичної моделі GPT-4. Як і попередні моделі, її створено на основі архітектури Трансформер і застосовано навчання з підкріпленням. Розробники стверджують, що нове покоління нейромережі вийшло помітно потужнішим, ніж GPT-3.5.

Ця мультимодальна модель спроможна оперувати не лише з текстом, а й із зображеннями. Вона може працювати з картинками, розуміти їх зміст та контекст, а також опрацьовувати запити на основі зображень. Відповіді GPT-4 поки що доступні лише в текстовому вигляді, нейромережа ще не отримала здатність малювати самостійно. На сьогодні функція обробки зображень проходить бета-тестування, для масового використання вона стане доступною пізніше.

GPT-4 має розширені можливості роботи з текстом: написати літературний твір, великий юридичний договір чи навіть код для повноцінної програми. Її оперативна пам'ять тепер містить до 25 000 слів, які вона може читати, аналізувати і генерувати. Вона краще розпізнає контекст і точніше дотримується заданої стилістики відповідей. GPT-4 стала більш творчою та гнучкішою, підлаштовується під користувача і ефективніше працює з «тонкими сценаріями».

GPT-4 має вдосконалену здатність складати іспити та тести з різних предметів. Модель показала відмінні результати з низки дисциплін, відчутно перевершивши показники попередньої моделі GPT-3.5, а також успіхи пересічної людини.

GPT-4 точніше перекладає текст, розробники протестували її 26 мовами і в 24 випадках результат був вищим, ніж у GPT-3.5. При цьому мовна модель, як і раніше, не володіє новітніми даними (до неї завантажена інформація станом на вересень 2021 року) і іноді надає упереджені відповіді та буває схильна до галюцинацій, тобто видає за правду свої власні фантазії.

OpenAI вже запровадила мовну модель GPT-4 у свій інтелектуальний чат-бот ChatGPT. На сьогоднішній день вона доступна лише для користувачів із платною підпискою Plus. Випробувати можливості нового алгоритму можуть користувачі веб-браузера Bing від Microsoft і сервісу для вивчення мов Duolingo.

Плагіни OpenAI надали ChatGPT модульну архітектуру, яка частково нагадує архітектуру людського мозку. Об'єднання GPT-4 з різними плагінами може стати шляхом спеціалізації функцій, схожої на людську.

Дослідники стурбовані тим, що можуть втратити можливість вивчати ці системи. OpenAI не розголошує подробиць розробки та навчання GPT-4, зокрема тому, що їй доводиться конкурувати з Google та іншими компаніями та країнами. Це не лише шкодить дослідженням, а й перешкоджає розумінню соціальних наслідків впровадження штучного інтелекту.

Однак, дослідникам краще просто зосередитися на вирішенні поточних завдань, наприклад, підвищити прозорість навчальних даних, надавати чітку інформацію про те, як галюцинації впливають на надійність таких моделей, як GPT-4, і стежити за тим, щоб на моделі перед публікацією встановлювалися захисні обмеження.

### Сфери та способи застосування

Технології GPT-3, GPT-4 та ChatGPT можна застосовувати у ряді сфер у різний спосіб.

* Генерація текстів на різні теми обсягом до 25 000 слів на десятках мов, а також переклад з однієї мови на іншу.
* Обробка зображень, яка стала доступною з появою GPT-4. Алгоритм розпізнає об'єкти на зображеннях й розуміє їх контекст. Наприклад, може пояснити, у чому сенс картинки-мему або що незвичайного/кумедного показано на фото.
* Написання програмного коду та консультування користувачів. Наприклад, GPT-4 може підказати, як здійснити ту чи іншу операцію чи процес. Алгоритм може знаходити баги в коді та перекладати з однієї мови програмування на іншу. Можна надіслати намальований (від руки або в редакторі) шаблон сайту або програми, щоб вона написала код для відповідного програмного забезпечення.
* Генерація професійного та творчого контенту: написання віршів, сценаріїв, есе та творів, текстів пісень та нот, публіцистичних та технічних статей, підготовка медичних рекомендацій, створення планів, розрахунків та прогнозів, проведення фінансового аналізу, генерація запитів для інших нейромереж тощо.

## Проблеми та підходи до їх усунення

### Проблема 1: Навчальні дані для великих мовних моделей

Мовні моделі навчаються на великому обсязі текстів, зібраних з різних джерел, таких як Інтернет, цифрові книги, новини. Ці тексти містять багато інформації про різні аспекти світу, такі як факти, думки, правила, жарти тощо. Ці знання з текстів використовують їх для генерації нових текстів.

Проте, є кілька проблем із цим підходом:

* Тексти не завжди достовірні, актуальні чи узгоджені між собою. Моделі можуть переймати помилки, застарілі дані чи суперечливі твердження з джерел. Це може призводити до генерації неправдиву або оманливої інформації.
* Тексти не завжди повні чи представницькі. LLM можуть ігнорувати чи недооцінювати важливість деяких тем чи груп людей, які мало представлені у їхніх даних. Це може призводити до того, що LLM виявляють усунення чи дискримінацію стосовно них.
* Тексти не завжди є єдиним чи найкращим способом передачі знань. LLM можуть не враховувати чи втрачати нюанси, контекст чи емоції, які можуть бути виражені через інші модальності, такі як зображення, звуки чи жести. Це може призводити до того, що LLM неправильно інтерпретують чи виражають сенс.

Можливі рішення:

* Покращити якість та різноманітність даних для навчання мовних моделей. Наприклад, можна використовувати методи фільтрації, перевірки фактів або анотації даних для усунення помилок чи невідповідностей у текстах. Можна використовувати методи аугментації даних для розширення чи балансування тематичного та соціального складу текстів.
* Покращити способи оцінки та контролю знань у мовних моделях. Наприклад, можна використовувати методи тестування, аналізу чи візуалізації знань для перевірки їх правильності, актуальності та узгодженості. Можна використовувати методи інтерпретованості та зрозумілості для розуміння їх логіки та мотивації.
* Покращити способи інтеграції та узагальнення знань у мовних моделях. Наприклад, можна використовувати методи мультимодального навчання, об’єднання чи адаптацію знань із різних джерел. Можна використовувати методи метанавчання, нульового чи малого навчання для швидкого чи ефективного вивчення нових знань.

### Проблема 2: Розуміння сенсу

LLM навчаються на основі статистичних закономірностей у мові, які вони знаходять у своїх даних. Вони навчаються передбачати наступне слово або фразу в тексті на основі попереднього контексту. Це дозволяє генерувати зв'язані та правдоподібні тексти.

Проте, є кілька проблем із цим підходом:

* Статистичне передбачення означає семантичне розуміння. LLM можуть не знати чи не враховувати значення, ціль чи наслідки поєднання слів. Це може призвести до того, що ВММ генерують безглузді, нерелевантні чи небезпечні тексти.
* Статистичне передбачення не означає логічний висновок. LLM можуть не слідувати чи не перевіряти правила, факти чи докази у своїх текстах. Це може призводити до того, що LLM генерують неправильні, суперечливі чи оманливі тексти.
* Статистичне передбачення означає творче вираження. LLM можуть не мати або не розвивати свій стиль чи перспективу у своїх текстах. Це може призводити до того, що ВММ генерують нудні, повторювані або плагіатні тексти.

Можливі рішення:

* Покращити способи вимірювання та підвищення розуміння мови ВММ. Наприклад, можна використовувати методи оцінювання, бенчмаркінгу або змагань для перевірки та порівняння здібностей мовної моделі до розуміння мови на різних рівнях (слова, фрази, речення, тексти) та в різних завданнях (класифікація, аналіз тональності, відповіді на питання тощо). Можна використовувати методи навчання з підкріпленням, активного навчання або інтерактивного навчання для стимулювання та покращення розуміння мови ВММ через зворотний зв'язок від людей чи інших агентів.
* Покращити способи інкорпорації та використання логіки та здорового глузду в мовних моделях. Наприклад, можна використовувати методи символьного обчислення, графових нейронних мереж або нейросимволічного навчання для поєднання статистичного та логічного подання знань. Можна використовувати методи впровадження, збагачення чи перевірки знань для додавання/уточнення логічних чи загальнозначущих фактів/правил у ВММ.
* Покращити способи заохочення та оцінки творчості та оригінальності мовної моделі. Наприклад, можна використовувати методи генерації різноманітності, новизни або якості для вимірювання та покращення творчого потенціалу. Можна використовувати методи персоналізації, адаптації, емпатії для розвитку стилю мовної моделі.

### Проблема 3: Взаємодія з людиною

ВММ можуть бути використані для створення або покращення різних програм, пов'язаних з мовою, таких як пошукові системи, чат-боти, асистенти, редактори. Вони можуть спілкуватися з людьми через текстові або голосові інтерфейси, надаючи їм інформацію, допомогу, навчання чи розваги.

Проте, є кілька проблем із цим підходом:

* LLM не завжди задовольняють потреби чи очікування людей. Вони можуть не розуміти або не враховувати контекст, ціль та настрій співрозмовників. Це може призвести до генерації невідповідних, небажаних чи образливих текстів.
* LLM не завжди дотримуються або сприяють етичним, соціальним, юридичним нормам та цінностям. Вони можуть порушувати або підривати права, свободи, інтереси своїх співрозмовників і генерувати шкідливі, небезпечні чи незаконні тексти.
* LLM не завжди відповідають за свої дії чи наслідки. Мовні моделі можуть не усвідомлювати чи не визнавати свою відповідальність за свої слова. Це може призводити до генерації помилкових, маніпулятивних або злочинних тексти.

Можливі рішення:

* Покращити способи вивчення та задоволення потреб і очікувань людей від використання LLM. Наприклад, можна використовувати методи аналізу користувачів, сегментації користувачів або профілювання користувачів для визначення характеристик, цілей та переваг різних груп користувачів. Можна використовувати методи зворотного зв'язку від користувачів, покращення реакції та оцінки користувачів на згенеровані тексти.
* Покращити способи дотримання та підтримки етичних, соціальних, юридичних норм та цінностей. Наприклад, можна використовувати методи аналізу етики, соціальної справедливості або правового регулювання для визначення та врахування основних принципів, правил та стандартів, яким повинні дотримуватися LLM. Можна використовувати методи аудиту, моніторингу для перевірки та усунення порушень, пов'язаних із текстами ВММ.
* Покращити способи прийняття та несення відповідальності за дії та наслідки, що спричинені LLM. Наприклад, можна використовувати методи ідентифікації, атрибуції або верифікації для визначення та підтвердження джерела, авторства чи справжності текстів. Можна використовувати методи документування, пояснення чи виправдання для надання та обґрунтування причин, мотивів чи цілей текстів ВММ.

## Ризики та особливості застосування LLM

Для використання LLM у продукті або побудові нового сервісу, важливо враховувати наступне.

По-перше, LLM може згенерувати все, що завгодно. Відповіді сучасних моделей не проходять перевірку на достовірність. Це означає, що можна отримати у відповідь недостовірну, небезпечну або токсичну інформацію. Ця проблема отримала назву AI-галюцинацій і зараз привертає увагу провідних вчених.

* Використання інформації, що не пройшла перевірку, може призвести до великих репутаційних втрат.
* Недостовірна інформація, що створена за допомогою LLM, стає причиною судових позовів.
* Люди схильні довіряти відповідям LLM – це створює загрози у сфері кібербезпеки.

По-друге, розмір тексту, який можна задавати на вхід моделі, обмежений, розмір відповіді моделі також має обмеження.

Це означає, що якщо потрібно обробити текст великого розміру, наприклад, зробити переклад книги або написати коротку анотацію з довгої статті або дати відповідь, згідно з довгою інструкцією, то для цього потрібно придумати алгоритм обробки тексту частинами.

По-третє, важливо усвідомлювати ризики, пов'язані з передачею персональних та інших корпоративних даних при використанні моделей через API хмарних сервісів.

* Вже є гучні випадки витоків корпоративних даних у ChatGPT – кейс компанії Samsung.
* Деякі великі компанії вводять правила, що забороняють працівникам використовувати ChatGPT - кейс Apple.
* Витік даних може статися всередині самих хмарних сервісів - кейс компанії OpenAI.

### Як оцінити перспективи використання LLM у своєму продукті

Щоб прийняти рішення про використання мовну модель у новому продукті, потрібно відповісти на запитання: «Якщо модель згенерує неправдоподібну/хибну/токсичну інформацію, чи це буде безпечно для користувача?»

* Якщо так, мовну модель можна використовувати.
* Якщо ні — краще утриматися від використання технології у продукті, або продумати і реалізувати необхідні правила фільтрації такого контенту, щоб мінімізувати ризики.

Для використання LLM у продуктах та сервісах найбільш безпечними кейсами буде, коли відповідь моделі проходить додаткову обробку сценаріями верифікації або використовується як асистент-помічник для людини, яка усвідомлює обмеження цієї технології та не допускає реалізації негативних ризиків.

### Висновок

Великі мовні моделі – це потужний та перспективний інструмент для обробки та генерації текстів природною мовою. Вони можуть бути корисні для різних цілей та програм, пов'язаних з мовою. Однак, вони також є складним і загадковим об'єктом для вивчення і розуміння. Важко зрозуміти, звідки вони знають, що вони розуміють у мові і як вони взаємодіють з людьми. Ці проблеми та виклики вимагають подальшого дослідження та розробки методів та рішень, які допоможуть краще зрозуміти, контролювати та використовувати великі мовні моделі.