УДК 681.3
DOI: 10.18523/2617-3808.2024.7.102-111

Глибовець М. М., Задохін Д. В., Дехтяр Б.-Я., Пєчкурова О. М.

ОБРОБЛЕННЯ ПРИРОДНОЇ МОВИ ЗА ДОПОМОГИ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ І МЕТОДІВ МАШИННОГО НАВЧАННЯ

У статті представлено аналіз можливостей великих мовних моделей для вирішення задач NLP. Описано особливості архітектури Transformer, що є основою для сучасних моделей з оброблення природної мови. Розглянуто окремі компоненти архітектури, їхню роль і важливість для роботи з людською мовою. Проведено порівняльний аналіз Transformer та інших наявних моделей для завдання машинного перекладу.

Проаналізовано фактори, що дали змогу створювати моделі з мільярдами параметрів — великі мовні моделі. Розглянуто сім'ю моделей Llama від Meta як приклад такої моделі. Особливу увагу було приділено моделям порівняно невеликого розміру, що можуть бути потужним і водночас доступним інструментом для оброблення природної мови.

Наразі глибинне машинне навчання і згорткові нейронні мережі (CNN) посідають важливе місце у сфері оброблення природної мови (NLP). Тому в статті оцінено ефективність використання його алгоритмів, моделей і методів для вирішення основних задач на прикладі задачі розпізнавання іменованих сутностей (NER).

Наведено методи глибинного навчання, які зробили революцію в NER, надавши можливість набагато краще розуміти контекст, фіксувати залежності на великих відстанях і ефективно використовувати великі обсяги даних. Проведено класифікацію моделей на основі трансформерів, що дають найкращі результати на цей момент. Зараз існує багато моделей, розроблених на основі трансформера.

Описано результати порівняння двох із найпоширеніших моделей — BERT (гарні результати у широкому спектрі завдань NLP, зокрема відповіді на запитання, класифікація тексту, висновок природною мовою, передбачення лівого і правого контексту слова) і GPT-3 (великі успіхи, як-от мовне моделювання, генерування тексту й відповіді на запитання). Ці моделі проходять попереднє навчання на великих текстових наборах даних, щоб вивчити фундаментальні мовні уявлення. Обидві моделі активно використовують потенціал тонкого налаштування.

Ключові слова: NLP, NER, CNN, машинне навчання, архітектура нейронних мереж, архітектура Transformer, машинний переклад, великі мовні моделі (Llama, BERT, GPT).

Вступ

Оброблення природної мови (NLP) — це один із напрямів досліджень у сфері штучного інтелекту, що полягає у створенні інструментів для оброблення, аналізу і генерування людської мови. Вирішення проблем NLP можуть потенційно стерти бар'єр у взаємодії між людиною і комп'ютером, адже для людей основний засіб комунікації — це мова, натомість як комп'ютерні системи в основному сприймають інформацію за строгими протоколами.

Можна виділити два основні підходи до розв'язання проблем NLP — статистичний і лінгвістичний. Лінгвістичний використовує лінгвістичні знання про граматику, морфологію, синтаксис і семантику [17]. Цей підхід намагається знайти структурні й логічні правила, які описують, як формується і використовується мова. За статистичним підходом моделі можуть вивчати закономірності в текстах, не маючи знань про граматику чи правила мови, просто аналізуючи велику кількість текстових даних [14].

Алгоритми машинного навчання належать до другого — статистичного підходу. Розвиток цих алгоритмів, зокрема поява глибоких нейронних мереж, дав сильний поштовх для розвитку NLP. До 2017 року рекурентні нейронні мережі [10], довготривала короткочасна пам'ять (LSTM) [4] і реку-

© *Глибовець М. М., Задохін Д. В., Дехтяр Б.-Я., Пєчкурова О. М.*, 2024

рентні мережі з воротами (GRU) [7] міцно закріпилися як передові підходи до задач моделювання послідовностей і трансдукції, таких як моделювання мови та машинний переклад.

Основним недоліком рекурентних нейронних мереж є послідовна природа їхньої роботи, що не дає можливості повною мірою використати паралельні обчислення для тренування та інференції [12]. Вирішити цю проблему намагались за допомогою згорткових нейронних мереж [6; 9], однак зі збільшенням вікна уваги сильно збільшувалась кількість параметрів та обчислювальна вартість роботи таких мереж.

Архітектуру Transformer було розроблено для вирішення наведених проблем в 2017 р. [22]. Відтоді нейронні мережі, побудовані на цій архітектурі, стали state-of-the-art інструментом для вирішення задач з оброблення природної мови.

Архітектура Transformer і механізм уваги

В основі архітектури Transformer лежить механізм уваги. Він встановлює зв'язки між різними словами однієї послідовності і використовує ці зв'язки для обчислення їх представлень. Самоувагу (self-attention) успішно застосовували в різних задачах, як-от розуміння прочитаного, абстрактне узагальнення, текстові висновки та навчання незалежних від задач представлень речень, і до появи Transformer [15; 16].

Особливість архітектури полягає у тому, що вона **цілком** покладається на цей механізм, не використовуючи рекурентну структуру чи згорткові фільтри.

Для кращого розуміння важливості уваги для NLP розглянемо такий приклад у задачі машинного перекладу з англійської на французьку. Є два речення, що відрізняються лише останнім словом: "The animal didn't cross the street because it was too tired.", "The animal didn't cross the street because it was too wide.". У цих реченнях слово "it" стосується різних понять: тварина у першому випадку і вулиця в другому. Ці слова мають різні роди в французькій мові, і тому для правильного перекладу другої половини речення моделі необхідно проаналізувати зв'язки між словом "it" та всіма іншими словами у реченні.

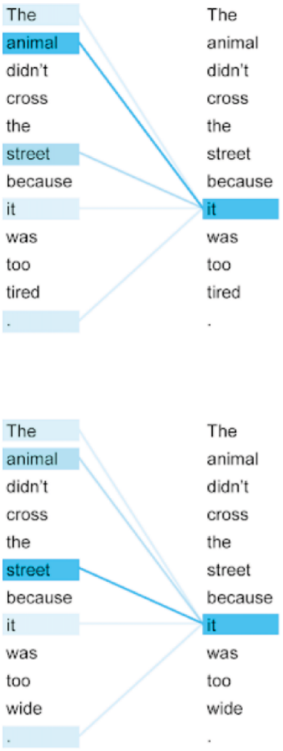

Рис. 1. Коефіцієнти уваги між словом it та іншими словами в реченнях

На рис. 1 зображено значення коефіцієнтів уваги між словами в обох реченнях, що були обраховані на одному з шарів мережі Transformer. Завдяки цій інформації модель змогла правильно перекласти обидва речення.

Оригінальна модель Transfomer має структуру encoder-decoder [5]: складається із двох складових енкодера і декодера. Енкодер (зліва на рис. 2) перетворює вхідну символьну послідовність на послідовність векторних представлень такої самої розмірності.

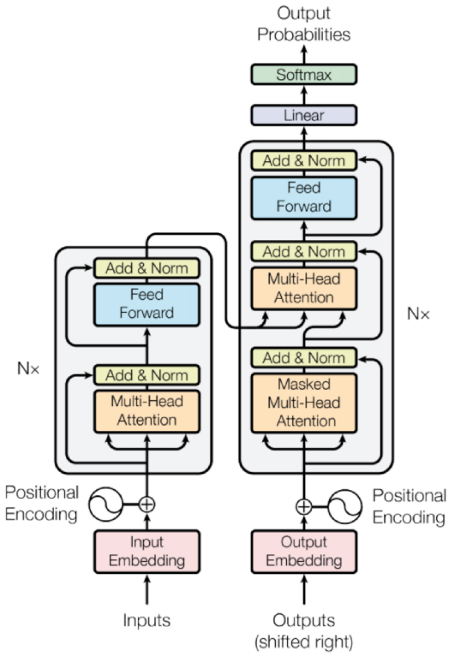

Рис. 2. Структура мережі Transformer

Роль декодера (справа на рис. 2) — із послідовності Z, згенерованої енкодером для вхідної послідовності токенів, послідовно генерувати послідовність токенів $Y = \{y_1, y_2, ..., y_m\}$. Цей процес відбувається, поки не згенерований спеціальний токен EOS (end of sequence) або не досягнуто граничної довжини відповіді.

Оброблення символьних вхідних даних починається з процесу word embedding — перетворення токенів на багатовимірні (d_{model}) числові вектори. Це відбувається завдяки окремо навченим embedding моделям, що натреновані переводити семантичні зв'язки між словами у просторові між багатовимірними векторами.

Також застосовують позиційне кодування. Позиція слова у реченні може сильно впливати на його значення, особливо в деяких мовах. Тому інформація про позицію кожного токена у послідовності необхідна для правильного розуміння тексту.

У кінці декодера міститься додаткове лінійне перетворення, а також застосовується функція *soft* max:

$$soft\max_i(X) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}.$$

На рис. 2 видно, що основною частиною енкодера і декодера є повторювані блоки. Кількість цих блоків N — гіперпараметр моделі, що напряму впливає на глибину мережі і кількість параметрів.

Блок енкодера складається з двох частин: Multi-Head Attention і Feed Forward. Перший буде розглянуто детальніше в наступних абзацах, а другий являє собою два лінійні перетворення із функцією активації Relu між ними. Для обох наявні залишкові з'єднання [11] і нормалізація [13].

Блок декодера має подібну структуру. Відмінністю є наявність двох Multi-Head Attention компонентів: перший відповідає за увагу над уже згенерованими токенами відповіді, другий на вхід також отримує представлення, створені енкодером.

Multi-Head Attention (досл. «багатоголова увага») — підхід до уваги, вперше використаний саме в архітектурі Transformer. Ідея полягає в тому, щоб порівнювати між собою не повнорозмірні вектори, а їхні проєкції на простори менших розмірностей (d_k). При цьому, розмірність цих просторів має бути кратно меншою від розмірності векторних представлень токенів (ембедингів): $d_{model} = d_k \cdot h$. Параметр h — це кількість центрів уваги.

У кожному «центрі уваги» незалежно тренуються матриці параметрів: W_Q, W_K, $W_V \in \mathbb{R}^{d_{model} \times d_k}$. Три матриці використовуються для трьох лінійних проекцій векторів із простору $\mathbb{R}^{d_{model}}$ в простір $\mathbb{R}^{d_k}$. Таким чином, для кожного вхідного представлення токена розмірності d_{model} буде отримано три вектори розмірності d_k: Q, K, V.

Для обчислення коефіцієнтів уваги для кожного слова рахується скалярний добуток між вектором Q і векторами K всіх елементів послідовності. Скалярні добутки діляться на коефіцієнт $\sqrt{d_k}$ а також застосовується функція *soft* max, — таким чином отримуються коефіцієнти між 0 та 1. Далі ці коефіцієнти перемножуються з векторами V для отримання наступних представлень нейронної мережі.

Результати кожної голови уваги мають розмірність d_k, склеюються для отримання повнорозмірного представлення, а також зазнають ще одного лінійного перетворення:

$$MultiHead(X) = concat(head_1, \ \dots \ , head_h)W_0$$

$$head_i = soft\max\left(\frac{XW_i^Q(XW_i^K)^T}{\sqrt{d_k}}\right)XW_i^V.$$

Усі матриці W — це треновані параметри.

У таблиці 1 наведено порівняльну характеристику різних типів шарів у нейронних мережах, що використовуються в NLP.

Таблиця 1

Порівняльна характеристика різних типів шарів у глибоких нейронних мережах для NLP

Тип шару	Обчислювальна складність	Послідовні операції
Self-Attention	$O(n^2 \cdot d)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(\log_k(n))$

Примітка: n — довжина послідовності, d — розмірність представлень токенів, k — розмір згорткового фільтру.

Із таблиці 1 видно, що обчислення представлень в шарі уваги може бути виконане за константну кількість послідовних операцій. Також обчислювальна складність лінійно залежить від розмірності моделі, на відміну від квадратичної залежності у інших моделях.

Це означає, що для обчислення уваги можна дуже ефективно застосовувати паралельні обчислення, а також що шари уваги менше чутливі до збільшення розмірності моделі.

Розробники архітектури Transformer показали, що, використовуючи тільки механізм уваги, можна досягти кращих результатів у обробленні природної мови за меншої (на декілька порядків) складності навчання та інференції. Також, використовуючи кластери графічних процесорів, можна порівняно швидко тренувати моделі на великих обсягах текстових даних.

Великі мовні моделі. Огляд моделей Llama

Оптимізованість архітектури Transformer для паралельних обчислень, а також висока ефективність архітектури для задач оброблення природної мови дали змогу тренувати моделі з величезною кількістю параметрів на терабайтах текстових даних.

Оригінальна модель Transformer мала приблизно 50 мільйонів параметрів і показала найкращі на той час результати з машинного перекладу. Розмір деяких сучасних мовних моделей уже перевищує трильйон параметрів.

Враховуючи величезні ресурси, що необхідні для тренування таких мереж, розробляти такі моделі мають можливість лише великі технологічні корпорації з майже необмеженими ресурсами. Розробляють додаткові оптимізації та нові підходи: моделі, створені декілька років тому, вже вважають застарілими і неактуальними.

Водночас з’являються відкриті моделі, що доступні для безкоштовного скачування, тренування на власних даних та використання. Прикладом можуть слугувати моделі Llama від компанії Meta [20; 21].

Llama — це група моделей різного розміру і версій. Першу версію випустила Meta AI у лютому 2023 р., вона тренована на 1,4 трильйона токенів. Модель було навчено на широкому спектрі джерел даних, із підтримкою 20 мов із латинським і кириличним алфавітами. Навчальні дані також брали із відкритих репозиторіїв на платформі Github, із книжок і наукових публікацій.

У найновішій версії Llama 3.2 моделі навчались вже на 15 трильйонах токенів. Серед моделей останньої версії доступні маленькі моделі розміром 1 і 3 мільярди, які можна запускати навіть на мобільних пристроях і які можуть виконувати нескладні задачі.

Більші моделі — 7B, 13B, 33B і 70B (B — від англ. billion). На кінець 2024 р. кожна із цих моделей набирає найвищі бенчмарки серед моделей своєї розмірної категорії. Найбільша модель має 405 мільярдів параметрів і становить конкуренцію пропрієтарним моделям більшого розміру. Моделі від 33B також треновані обробляти графічні зображення.

Оскільки моделі відкриті, будь-хто може завантажити, дотреновувати та публікувати свої версії моделей. У мережі вже доступні сотні кастомізацій мережі, кожна з яких спеціалізується для вирішення конкретної задачі.

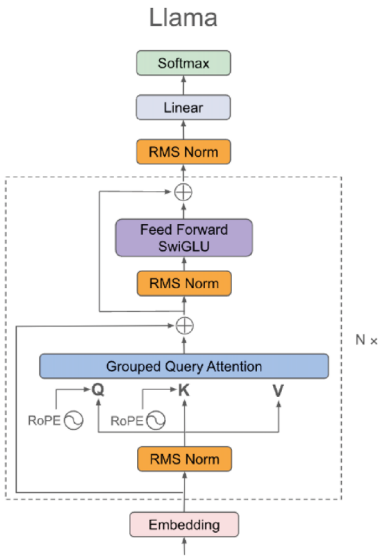

Рис. 3. Структура моделі Llama

На рис. 3 зображено структурну діаграму Llama. Найпомітніша відмінність від попередньо розглянутої архітектури — немає окремого енкодера. Моделі, що спеціалізуються саме на генерації тексту, часто використовують таку структуру — decoder only Transformer. Інша принципова відмінність полягає у механізмі уваги.

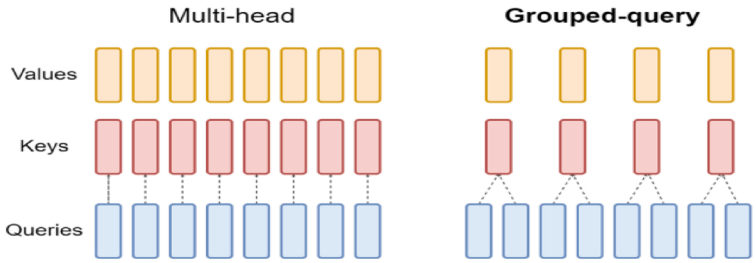

Рис. 4. Grouped-query Attention vs Multi-head Attention

Із другої версії в Llama застосовується механізм уваги Grouped-query Attention (рис. 4) [3]. Multi-head Attention, що застосовувався в оригінальній мережі Transformer, передбачав унікальні лінійні перетворення для всіх трьох векторів Q, K і V для кожної голови уваги. Grouped-query Attention пропонує таку оптимізацію: голови уваги згруповані, перетворення Q залишаються унікальними для кожної із них, а K і V — спільні у кожні групі. Така оптимізація дещо негативно впливає на точність моделі, але дуже сильно зменшує кількість тренованих параметрів.

Окрім механізму уваги, змінились також методи позиційного кодування (rotary position embedding [19]), нормування (RMS normalisation [24]) та функції активації в лінійних шарах мережі (SwiGLU [18]).

Ефективність використання моделей глибинного навчання в NLP на прикладі NER

Нині глибинне машинне навчання посідає важливе місце у сфері оброблення природної мови (NLP). Важливо оцінити ефективність використання його алгоритмів, моделей і методів для вирішення основних задач, наприклад задачі розпізнавання іменованих сутностей (NER) [1].

Розпізнавання іменованих сутностей (NER) — це підгалузь оброблення природної мови (NLP), яка зосереджена на ідентифікації та класифікації конкретних сутностей із текстового вмісту. NER працює з важливими деталями тексту, відомими як іменовані сутності — окремі слова, фрази або послідовності слів — шляхом ідентифікації та категоризації їх у заздалегідь визначені групи. Категорії охоплюють різноманітне коло тем, згаданих у тексті, зокрема імена людей, географічне розташування, назви організацій, дати, події та навіть конкретні кількісні значення, такі як гроші та відсотки.

Задача розпізнавання іменованих сутностей (ЗРІС) має довгу історію спроб розв'язання (ведення великої колекції імен, застосування регулярних виразів, оснований на правилах і списках шаблонів на основі токенів слів та POS-тегів), що робить її більш придатною для ілюстрації використання машинного навчання. Ця задача залишається актуальною і зараз. Підходи до її розв'язку з використанням глибинного машинного навчання активно розвиваються [2]. Серед сфер застосування традиційно виділяють пошук інформації, рекомендаційні системи. У сфері машинного навчання ЗРІС зазвичай називають «маркуванням (розміткою) послідовностей».

Пояснимо цю концепцію. Розглянемо речення: «Я ... косу». Якщо класифікатор аналізує слово «косу», тоді для вирішення, чи воно стосується зачіски людини, берега річки або знаряддя праці, потрібно розглядати контекст речення (враховувати слова навколо неї). Починають працювати класифікатори послідовностей. Вони враховують слова, які передують поточному слову і йдуть за ним. Наприклад, якщо попереднє слово вказує на річку, існує більша ймовірність того, що це слово є іменником, що належить до типу берега річки. Такий підхід доповнюється інженерією функцій, де конкретні атрибути та інформація про дані витягуються вручну для покращення продуктивності моделі. До поширених функцій належать: характеристика слова, контекст, синтаксична інформація, шаблони слів, морфологічні подробиці.

Серед базових методів реалізації тут традиційно виділяють машини опорних векторів (SVM), дерева рішень та умовні випадкові поля (CRF).

Машинне навчання використовують, коли сутності бувають різних типів і не завжди дотримуються послідовної схеми. Якщо існує великий набір анотованих даних, тоді можна навчити модель, яка зможе добре узагальнювати різні шаблони сутностей. Цей підхід проблемний на етапі розроблення функцій, значно залежить від якості та розміру набору даних, потребує значних обчислювальних ресурсів і додаткового оброблення зашумлених даних. Звідси випливає неідеальна продуктивність. Безсумнівною перевагою цього підходу є значна адаптивність, добре усвідомлення контексту та покращене запам'ятовування.

У цьому розділі ми зосередимось на уточненні ефективності використання моделей глибинного навчання у розв'язанні ЗРІС.

Рекурентні нейронні мережі (РНМ) мають спеціальний механізм пам'яті, який дає їм змогу розглядати слова, що з'являються до і після аналізованого слова. Однак у них виникає проблема зникаючих і вибухових градієнтів, коли аналізуються довгі послідовності. Це впливає на ефективність фіксації контекстної інформації. Усувають ці проблеми мережі довгої короткочасної пам'яті, або LSTM (long short-term memory).

Архітектура LSTM — це тип РНМ зі складнішою коміркою пам'яті. Така комірка вирішує проблему зникаючого градієнта, явно керуючи потоком інформації. Комірка має спеціальний вентиль забуття для прийняття рішення про те, яку частину інформації з попередніх комірок слід зберігати, вхідний вентиль для контролю кількості нової інформації, що має зберігатися, та вихідний вентиль для визначення ступеня впливу поточного вмісту на вихід.

Використання такої пам'яті дає їм можливість ефективно фіксувати далекосяжні залежності в тексті. Така архітектура добре застосовується до вхідних даних, які охоплюють одне або два речення. Проблеми з залежностями на великій відстані тут виникають, коли справа доходить до оброблення великої контекстної семантики. Тоді ці мережі можуть неефективно вловлювати нюанси дуже

довгих текстових послідовностей. Недоліком архітектури LSTM є обмежена можливість розпаралелювання і виявлення довготривалих залежностей.

Однією з ключових переваг трансформерної архітектури є те, що її можна навчати на великих обсягах даних паралельним способом, що робить її дуже масштабованою. Багато випадків використання NER включають вхідний текст, який охоплює кілька абзаців і сторінок. Трансформер ідеально підходить для цього.

Зараз існує багато моделей, розроблених на основі трансформера. Розглянемо одні з найпоширеніших —BERT і GPT-3.

BERT — це модель, розроблена Google, яка досягла гарних результатів у широкому спектрі завдань NLP, як-от відповіді на запитання, класифікація тексту і висновок природною мовою. Вона складається з двонаправленого кодера-трансформера, що означає, що модель навчена передбачати лівий і правий контекст цього слова. BERT попередньо навчається на великих обсягах текстових даних за допомогою двох завдань: моделювання замаскованої мови та прогнозування наступного речення.

GPT-3 — це мовна модель, розроблена OpenAI, яка досягла вражаючих результатів у широкому спектрі завдань NLP, як-от мовне моделювання, генерація тексту та відповіді на запитання. Вона була (до недавнього часу перемагала Мегатрон-Тюрінг NLG з параметрами 530B [23]) найбільшою мовною моделлю з 175 мільярдами параметрів, навченою на різноманітному діапазоні текстових даних. GPT-3 — це генеративна модель, яка генерує новий текст на основі заданого запиту. На відміну від BERT, вона не така багатоцільова, навчена генерувати текст, і, отже, не такий хороша, як спеціалізовані моделі NER. Спроби використовувати GPT для NER [8] існують, але результати не є значними.

Спробуємо порівняти BERT і GPT.

Ні GPT, ні BERT безпосередньо не виконують конкретні завдання у своїй базовій формі. Вони проходять попереднє навчання на масивних текстових наборах даних, щоб вивчити фундаментальні мовні уявлення. Обидві моделі активно використовують потенціал тонкого налаштування: їх можна точно налаштувати для різних подальших завдань, таких як NER, аналіз настроїв, відповіді на запитання та узагальнення тексту.

Зрозуміло, що є і відмінності. Насамперед це архітектура: GPT оснований на архітектурі авторегресійного трансформера, де він передбачає наступне слово на основі попередніх; BERT — на двонаправленій архітектурі трансформера, що дає можливість аналізувати все речення одночасно. Окрім цього, у них різна мета попередньої підготовки: GPT — передусім навчений для генерації мови; BERT — для розуміння мовного контексту. У таблиця 2 узагальнено відмінності між GPT і BERT.

Таблиця 2

Відмінності між GPT і BERT

	GPT	**BERT**
Архітектура	Авторегресійний трансформер	Двонаправлений трансформер
Мета попередньої підготовки	Передусім підготовлений для генерації мови	Навчений розуміти мовний контекст
Для чого добре підходить	Творче письмо, генерація коду та діалог	Аналіз настроїв, відповіді на запитання та NER

Згорткові нейронні мережі (CNN) спочатку було розроблено для оброблення зображень. Тим не менш, їх також успішно застосовують для завдань оброблення природної мови, таких як класифікація тексту (насправді, включно з NER), аналіз настроїв і мовне моделювання.

У NLP CNN зазвичай використовують для оброблення послідовностей вкладень слів змінної довжини, де кожне слово представлено у вигляді щільного вектора у високовимірному просторі. Вхідна послідовність спочатку пропускається через один або кілька згорткових шарів, які застосовують фільтри фіксованої ширини до вхідної послідовності для вилучення локальних об'єктів. Вихідні дані згорткових шарів потім пропускаються через один або кілька шарів об'єднання, які зменшують дискретизацію карт об'єктів, щоб зменшити їхню розмірність.

У NLP використовується кілька варіацій архітектури CNN, залежно від конкретного завдання і характеру вхідних даних. Одним із поширених підходів є використання кількох фільтрів різного розміру для фіксації різних ознак n-грам, таких як уніграми, біграми і триграми. Вихідні дані кожного фільтра потім об'єднуються в один вектор ознак і пропускаються через один або кілька повністю пов'язаних шарів для класифікації або регресії.

Іншим підходом є ієрархічна архітектура CNN, де вхідна послідовність спочатку розбивається на менші підпослідовності, такі як речення або абзаци. Кожна підпослідовність обробляється незалеж-

но CNN нижчого рівня. Вихідні дані CNN нижчого рівня потім об'єднуються і обробляються CNN вищого рівня, щоб зафіксувати глобальні особливості та залежності між підпослідовностями.

CNN мають низку переваг для завдань NLP. Вони є обчислювально ефективними та можуть паралельно обробляти вхідні послідовності, що робить їх придатними для великомасштабних наборів даних. Вони також можуть фіксувати локальні особливості та закономірності в послідовності введення, що може бути корисним для таких завдань, як аналіз настроїв і класифікація тексту. Однак CNN менш ефективні в моделюванні довгострокових залежностей і послідовних зв'язків у текстовій послідовності, ніж рекурентні нейронні мережі (РНМ) і трансформерні моделі, які краще підходять для мовного моделювання і завдань машинного перекладу.

Загалом, CNN у NER насамперед цікаві з історичної точки зору; за таблицею лідерів paperswithcode.com, найефективніша архітектура на основі CNN розташована на 58-й позиції; це гібридна модель, що використовує комбіновану потужність CNN і LSTM, яка була представлена ще в 2015 р. Отже, за останні 9 років не було зроблено жодних покращень у NER з архітектурами на основі CNN.

Для прикладу розглянемо хронологію використання NER в Electronic Health Records (див. рис. 5, запозичений із [21]). Як бачимо, глибоке навчання з'являється у 2017 р. До 2019 р. панівною архітектурою стала двонаправлена довготривала короткочасна пам'ять (BiLSTM). Однак з 2021 р. архітектура BERT та її варіанти стали основною моделлю NER, що застосовується до ЕМК, і ця тенденція зберігається донині.

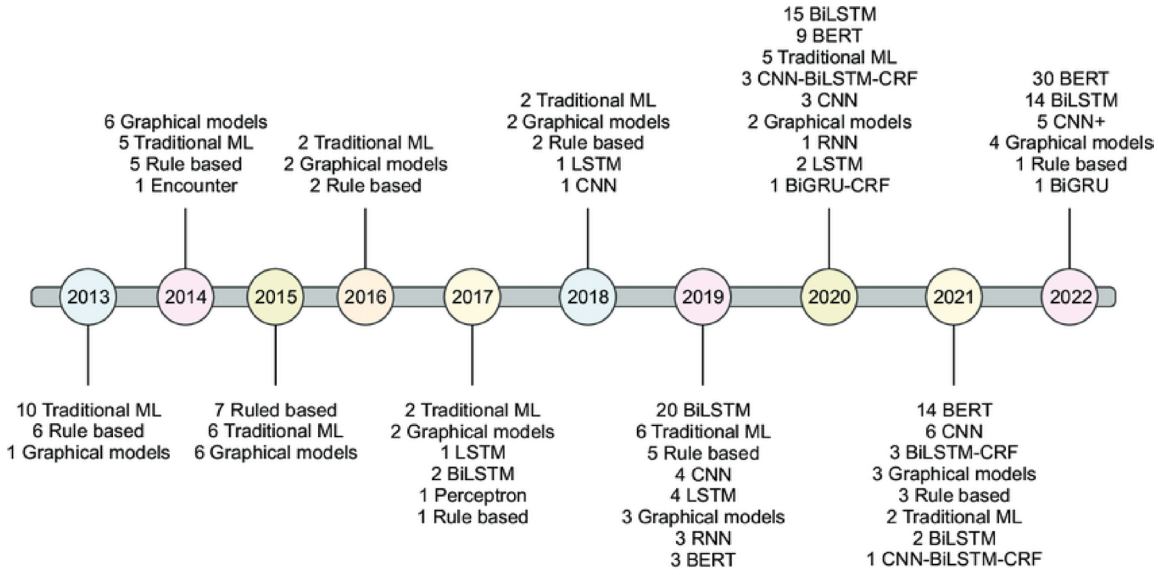

Рис. 5. Еволюція підходів

Висновки

Поява архітектури Transformer відкрила можливості у вирішенні задач оброблення природної мови, що ще зовсім недавно були недоступними.

Моделі з багатьма мільярдами параметрів можуть виконувати різноманітні задачі над текстом. Додаткові оптимізації, а також розвиток технічних засобів для паралельних обчислень призвели до появи ефективніших, більших і розумніших моделей. Навчання та використання гігантських мереж є складним і дорогим процесом, однак існують альтернативні, доступніші моделі. Моделі Llama від компанії Meta є хорошим прикладом такого доступного рішення з високим відношенням розмір / ефективність.

Поява такого доступного NLP засобу відкриває нові горизонти в розробленні комп'ютерних систем, що здатні взаємодіяти з людиною за допомогою природної мови.

З іншого боку, не варто цілком покладатись на великі мовні моделі при вирішенні задач. У моделей є ціла низка недоліків, такі як можливість галюцинацій, неможливість знайти джерело інформації та причин тієї чи тієї відповіді.

Оптимальним сценарієм використання є гібридні системи, в яких велика мовна модель (або декілька) є лише одним з компонентів, що виконує вузькоспеціалізовану задачу з оброблення природної мови.

Методи глибокого навчання зробили революцію в NER, надавши методи з набагато кращим розумінням контексту, здатністю фіксувати залежності на великих відстанях і ефективно використовувати великі обсяги даних. Моделі на основі трансформерів дають найкращі результати на цей момент, переважною архітектурою є BERT. Однак дослідження в цій галузі все ще тривають, і нові методи (наприклад, підказки) незабаром можуть витіснити наявні. Незважаючи на те, що трансформери є найбільш поширеними, все ще використовуються інші, більш базові моделі, наприклад LSTM, CNN, і навіть методи, основані на правилах.

Список літератури

1. Глибовець А. М. Автоматизований пошук іменованих сутностей у нерозмічених текстах українською мовою / А. М. Глибовець // Штучний інтелект. — 2017. — № 2. — С. 45–51.
2. Глибовець А. М. Архітектура системи моделей глибоких нейронних мереж для знаходження подібності об'єктів в гетерогенному середовищі / А. М. Глибовець, Т. І. Легіневич // V Міжнародно науково-практична конференція «Обчислювальний інтелект». — Ужгород, 2019. — С. 249–250.
3. Ainslie Joshua. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints [Electronic resource] / Joshua Ainslie et al. — 2023. — Mode of access: arXiv:2305.13245.
4. Cheng Jianpeng. Long short-term memory-networks for machine reading [Electronic resource] / Jianpeng Cheng, Li Dong, and Mirella Lapata. — 2016. — Mode of access: arXiv:1601.06733.
5. Cho Kyunghyun. Learning phrase representations using rnn encoder-decoder for statistical machine translation. CoRR, abs [Electronic resource] / Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 1406.1078, 2014. — Mode of access: https://aclanthology.org/D14-1179/.
6. Chollet Francois. Xception: Deep learning with depthwise separable convolutions [Electronic resource] / Francois Chollet. — 2016. — Mode of access: arXiv:1610.02357.
7. Chung Junyoung. Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR, abs [Electronic resource] / Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, Yoshua Bengio. — Mode of access: https://arxiv.org/abs/1412.3555.
8. Durango María. Named Entity Recognition in Electronic Health Records: A Methodological Review / María Durango, Silva Ever Torres, Andres Orozco-Duque // Healthcare Informatics Research. — 2023. — Vol. 29. — Pp. 286–300. — https://doi.org/10.4258/hir.2023.29.4.286.
9. Gehring Jonas. Convolutional sequence to sequence learning [Electronic resource] / Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, Yann N. Dauphin. — 2017. — Mode of access: arXiv:1705.03122v2.
10. Graves Alex. Generating sequences with recurrent neural networks [Electronic resource] / Alex Graves. — 2013. — Mode of access: arXiv:1308.0850.
11. He Kaiming. Deep residual learning for image recognition / Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2016. — Pp. 770–778.
12. Hochreiter Sepp. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies [Electronic resource] / Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber. — 2001. — Mode of access: https://www.researchgate.net/publication/2839938_Gradient_Flow_in_Recurrent_Nets_the_Difficulty_of_Learning_Long-Term_Dependencies.
13. Lei Ba Jimmy. Layer normalization [Electronic resource] / Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton. — 2016. — Mode of access: arXiv:1607.06450.
14. Manning Christopher. Foundations of statistical natural language processing / Christopher Manning, Schutze Hinrich. — MIT press, 1999.
15. Parikh Ankur. A decomposable attention model. In Empirical Methods in Natural Language Processing [Electronic resource] / Ankur Parikh, Oscar Täckström, Dipanjan Das, Jakob Uszkoreit // Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. — 2016. — Pp. 2249–2255. — Mode of access: https://aclanthology.org/D16-1244/.
16. Paulus Romain. A deep reinforced model for abstractive summarization [Electronic resource] / Romain Paulus, Caiming Xiong, Richard Socher. — 2017. — Mode of access: arXiv:1705.04304.
17. Sag Ivan A. Linguistic theory and natural language processing/ Ivan A. Sag // Natural Language and Speech: Symposium Proceedings Brussels. — November 26/27, 1991. — Berlin, Heidelberg : Springer Berlin Heidelberg, 1991.
18. Shazeer Noam. Glu variants improve transformer [Electronic resource] / Noam Shazeer. — 2020. — Mode of access: arXiv:2002.05202.
19. Su Jianlin. Roformer: Enhanced transformer with rotary position embedding / Su Jianlin et al. // Neurocomputing. — 2024. — 568: 127063.
20. Touvron Hugo. Llama 2: Open foundation and fine-tuned chat models [Electronic resource] / Hugo Touvron et al. — 2023. — Mode of access: arXiv:2307.09288.
21. Touvron Hugo. Llama: Open and efficient foundation language models [Electronic resource] / Hugo Touvron et al. — 2023. — Mode of access: arXiv:2302.13971.
22. Vaswani A. Attention is all you need / A. Vaswani // Advances in Neural Information Processing Systems. — 2017.
23. Wang S. Named entity recognition via large language models [Electronic resource] / S. Wang, X. Sun, X. Li, R. Ouyang, F. Wu, T. Zhang, ... & G. Wang. — 2023. — Mode of access: arXiv:2304.10428.
24. Zhang Biao. Root mean square layer normalization / Biao Zhang, Rico Sennrich // Advances in Neural Information Processing Systems. — 2019. — Vol. 32.

References

Ainslie, Joshua, et al. (2023). GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. arXiv:2305.13245.

Cho, Kyunghyun, Merrienboer, Bart van, Gulcehre, Caglar, Bougares, Fethi, Schwenk, Holger, & Bengio, Yoshua. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. CoRR, abs, 1406.1078. https://aclanthology.org/D14-1179/.

Chollet, Chung, Junyoung, Gülçehre, Çaglar, Cho, Kyunghyun, & Bengio, Yoshua. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR, abs. 1412.3555. https://arxiv.org/abs/1412.3555.

Durango, María, Torres, Silva Ever, & Orozco-Duque, Andres. (2023) Named Entity Recognition in Electronic Health Records: A Methodological Review. *Healthcare Informatics Research*, *29*, 286–300. https://doi.org/10.4258/hir.2023.29.4.286.

Francois. (2016). Xception: Deep learning with depthwise separable convolutions. arXiv:1610.02357.

Gehring, Jonas, Auli, Michael, Grangier, David, Yarats, Denis, & Dauphin, Yann N. (2017). Convolutional sequence to sequence learning. arXiv:1705.03122v2.

Graves. Alex. (2013). Generating sequences with recurrent neural networks. arXiv:1308.0850.

Hlybovets, A. M. (2017). Avtomatyzovanyi poshuk imenovanykh sutnostei u nerozmichenykh tekstakh ukrainskoiu movoiu. *Shtuchnyi intelekt*, *2,* 45–51 [in Ukrainian].

Hlybovets, A. M., Lehinevych, T. I. (2019). Arkhitektura systemy modelei hlybokykh neironnykh merezh dlia znakhodzhennia podibnosti obiektiv v heterohennomu seredovyshchi. In *Mizhnarodno naukovo-praktychna konferentsiia "Obchysliuvalnyi intelekt"* (pp. 249–250) [in Ukrainian].

Hochreiter, Sepp, Yoshua Bengio, Paolo Frasconi, & Jürgen Schmidhuber. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. https://www.researchgate.net/publication/2839938_Gradient_Flow_in_Recurrent_Nets_the_Difficulty_of_Learning_Long-Term_Dependencies.

Jianpeng, Cheng, Li Dong, & Mirella, Lapata. (2016). Long short-term memory-networks for machine reading. arXiv:1601.06733.

Kaiming, He, Zhang, Xiangyu, Ren, Shaoqing, & Sun, Jian. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).

Lei Ba, Jimmy, Kiros, Jamie Ryan, & Hinton, Geoffrey E. (2016). Layer normalization. arXiv:1607.06450.

Manning, Christopher, & Hinrich, Schutze. (1999). *Foundations of statistical natural language processing*. MIT press.

Parikh Ankur, Täckström, Oscar, Das, Dipanjan, & Uszkoreit, Jakob. (2016). A decomposable attention model. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 2249–2255). https://aclanthology.org/D16-1244/.

Paulus, Romain, Xiong, Caiming, & Socher, Richard. (2017). A deep reinforced model for abstractive summarization. arXiv:1705.04304.

Sag, Ivan A. (1991). Linguistic theory and natural language processing. *Natural Language and Speech: Symposium Proceedings Brussels*. Springer Berlin Heidelberg.

Shazeer, Noam. (2020). Glu variants improve transformer. arXiv:2002.05202.

Su, Jianlin, et al. (2024). Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063.

Touvron, Hugo, et al. (2023a). Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

Touvron, Hugo, et al. (2023b). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systemsm*.

Wang, S., Sun, X., Li, X., Ouyang, R., Wu, F., Zhang, T., ... & Wang, G. (2023). Gpt-ner: Named entity recognition via large language models. arXiv preprint arXiv:2304.10428.

Zhang, Biao, & Rico Sennrich. (2019) Root mean square layer normalization. *Advances in Neural Information Processing Systems*, *32*.

M. Glybovets, D. Zadokhin, B. Dekhtiar, O. Pyechkurova

NATURAL LANGUAGE PROCESSING USING LARGE LANGUAGE MODELS AND MACHINE LEARNING METHODS

The article analyzes the capabilities of large language models in solving NLP tasks. It describes the features of the Transformer architecture, which serves as the foundation for modern natural language processing models. The individual components of the architecture, their roles, and their significance for working with human language are discussed. A comparative analysis of the Transformer and other existing models in the context of machine translation task is provided.

Factors that have enabled the development of models with billions of parameters—known as large language models—are analyzed. The Llama model family from Meta is reviewed as an example of such models. Special attention is given to smaller-scale models, which can be powerful yet accessible tools for natural language processing.

Currently, deep machine learning and convolutional neural networks (CNN) hold an important place in the field of natural language processing (NLP). Therefore, the article evaluates the effectiveness of these algorithms, models, and methods for solving key tasks, using the named entity recognition (NER) task as an example.

Deep learning methods have revolutionized NER, providing a significantly better understanding of context, capturing dependencies over long distances, and enabling the effective use of large datasets. A classification of Transformer-based models that currently yield the best results is provided. Currently, many models have been developed based on the Transformer architecture.

We describe the results of comparing two of the largest BERT models (which have achieved strong results across a wide range of NLP tasks, including question answering, text classification, natural language interference, and context prediction) with GPT-3 (which has demonstrated impressive successes in language modeling, text generation, and question answering). These models are pre-trained on large-scale textual datasets to learn fundamental linguistic representations. Both models leverage fine-tuning to enhance their performance.

Keywords: NLP, NER, CNN, machine learning, neural network architecture, Transformer architecture, machine translation, large language models (Llama, BERT, GPT).

Матеріал надійшов 23.09.2024

Creative Commons Attribution 4.0 International License (CC BY 4.0)