КІБЕРНЕТИКА

УДК 004.89

О.Г. СКУРЖАНСЬКИЙ
Київський національний університет імені Тараса Шевченка, Київ, Україна,
e-mail: *oleksandr.skurzhanskyi@gmail.com.*

О.О. МАРЧЕНКО
Київський національний університет імені Тараса Шевченка, Київ, Україна,
e-mail: *rozenkrans17@gmail.com.*

А.В. АНІСІМОВ
Київський національний університет імені Тараса Шевченка, Київ, Україна,
e-mail: *avatatan@gmail.com.*

СПЕЦІАЛІЗОВАНЕ ПОПЕРЕДНЄ НАВЧАННЯ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ НА СИНТЕТИЧНИХ ДАНИХ ДЛЯ ПОКРАЩЕННЯ ГЕНЕРАЦІЇ ПЕРЕФРАЗУВАННЯ

Анотація. Генерація перефразувань є фундаментальною проблемою в галузі обробки природних мов. Завдяки значному успіху технології перенесення навчання підхід «попереднє навчання → точне налаштування» став стандартним. Однак популярні універсальні методики попереднього навчання зазвичай потребують величезних наборів даних та значних обчислювальних потужностей, а доступні навчені моделі обмежені фіксованою архітектурою та розміром. Запропоновано простий та ефективний підхід до попереднього навчання спеціально для генерації перефразувань, який помітно підвищує якість генерації перефразувань та забезпечує суттєве покращення моделей загального призначення. Використано як наявні публічні дані, так і нові, згенеровані великими мовними моделями. Досліджено, як ця процедура попереднього навчання впливає на нейронні мережі різної архітектури, та доведено, що вона працює ефективно для всіх архітектур.

Ключові слова: штучний інтелект, машинне навчання, нейронні мережі, генерація перефразування, попереднє навчання, точне налаштування.

ВСТУП

Задача генерації перефразувань є однією з найпопулярніших та найскладніших у галузі обробки природних мов. По-перше, ця задача є спеціальним випадком генерації тексту. Є багато моделей генерації тексту, які можна застосувати до задачі генерації перефразувань. По-друге, остання по суті є аналогом машинного перекладу. Єдина відмінність полягає у тому, що речення має бути «перекладене» на ту саму мову, але з використанням інших слів. Тому до цієї задачі можна безпосередньо застосовувати не лише моделі машинного перекладу, а й метрики якості машинного перекладу, які часто підходять для оцінювання систем перефразування.

Особливістю задачі генерації перефразувань, на відміну від інших задач обробки природних мов, є велика кількість публікацій, які не використовують анотовані дані, а оперують лише звичайними текстовими корпусами. Зауважимо, що вхід та вихід для цієї задачі є взаємозамінними: якщо з речення $x_1, x_2, \ldots, x_n$ можна з високою ймовірністю отримати речення $y_1, y_2, \ldots, y_k$, то

© О.Г. Скуржанський, О.О. Марченко, А.В. Анісімов, 2024

логічно, що для вхідної послідовності $y_1, y_2, \ldots, y_k$ вивід $x_1, x_2, \ldots, x_n$ повинен мати високу ймовірність. Понад те, кожне речення не повинно мати строго одного перефразування та може бути переписане у різні способи, що підкреслює ймовірнісну природу проблеми. Для задачі генерації перефразувань є відносно велика кількість джерел даних різної якості та рівнів.

Починаючи з кінця 2022 р., у галузі обробки природної мови набули популярності великі мовні моделі. Ці системи дають змогу генерувати та редагувати текст з надзвичайною якістю завдяки своїм великим обсягам даних та властивостям глибокого навчання. Вони здатні генерувати перефразування з разючою природністю та різноманіттям, що робить їх корисними інструментами для покращення роботи з текстом та розуміння семантики.

У цій статті описано підхід до покращення якості нейронних мереж, які слугують для генерації перефразувань. Запропоновано просту та ефективну процедуру попереднього навчання, яка є специфічною для цієї задачі. Вона послідовно підвищує якість на різних наборах оцінювання та архітектурах моделей. Завдяки послідовному навчанню на синтетичних даних (менш якісних наявних даних, а також нових, згенерованих великими мовними моделями) вдається досягти визначної якості у розв'язанні задачі перефразування, перевершуючи найкращі підходи.

ЗАПРОПОНОВАНИЙ ПІДХІД

На сьогодні переважає парадигма «попереднє навчання $\rightarrow$ точне налаштування». Зокрема в обробці природної мови попереднє навчання моделей забезпечило значне підвищення якості. Доведено [1], що ця техніка підвищує надійність моделі, збагачує її кращим контекстним представленням і додатковими знаннями. Зазвичай для попереднього навчання використовують великі корпуси нерозмічених текстів. Цільові задачі навчання можуть бути як загальними, наприклад, моделювання мови з маскуванням, так і специфічними для конкретної задачі. Широко відомо, що синтетичне генерування даних (задача очищення тексту від «шуму») підвищує точність нейронних мереж для виправлення граматичних та орфографічних помилок [2, 3].

Водночас використання загальнодоступних моделей значно обмежує вибір архітектури моделі. Це часто становить серйозну проблему, особливо з погляду часу генерації тексту. Нерідко дослідник може прагнути до гнучкості в архітектурі моделі. Наприклад, універсальні моделі seq2seq зазвичай мають однакову кількість шарів для кодування та декодування, хоча було показано [4], що неглибоке декодування дає значне прискорення, часто без втрати точності.

Спеціалізоване попереднє навчання, орієнтоване на конкретну задачу, сприяє розв'язанню цієї проблеми. Зауважимо, що для реалізації цього підходу потрібно мати набір даних, який би містив надзвичайно велику кількість прикладів та водночас був тісно пов'язаним із задачею. Для генерації перефразувань добре підходить синтетично згенерована база даних ParaNMT-50M [5]. Вона містить понад 50 мільйонів пар перефразувань речень типу «англійська–англійська». Її створено автоматично за допомогою нейронного машинного перекладу для перекладу неанглійської частини великого паралельного корпусу. Отже, можна спочатку навчити модель на цих даних, а потім точно налаштувати її на конкретних наборах даних для генерації перефразувань.

Станом на 2023 р. передовим підходом до генерації синтетичних даних є застосування великих мовних моделей на кшталт ChatGPT [6]. Ці системи вже продемонстрували надзвичайну якість генерації завдяки навчанню на

безпрецедентно великій кількості даних та їхньому величезному розміру (десятки мільярдів параметрів). Текстова модальність робить їх універсальним інтерфейсом для розв'язання будь-яких задач, які обмежуються природною мовою — потрібно лише коректно підібрати опис завдання (prompt). Генерація перефразувань є саме такою задачею. Далі фактично відбувається дистиляція на рівні даних — менша модель навчається на цих даних.

У такий спосіб вибудовують навчальний «ланцюг»: від великого обсягу синтетичних даних найгіршої якості до анотованих людьми найбільш якісних даних відносно невеликого обсягу. Спочатку виконують навчання на синтетично згенерованій базі даних ParaNMT-50M, далі здійснюють налаштування на згенерованих парах ChatGPT і остаточне точне налаштування на фінальних навчальних даних та оцінювання якості на їхніх відповідних тестувальних вибірках.

Використані дані. Як і в більшості досліджень з навчання моделі з учителем для генерації перефразувань, в експериментах використано набори даних MSCOCO [7] та Quora Question Pairs (QQP). Набір даних MSCOCO від початку створено для задачі опису зображень. Кожне зображення відповідає п'яти різним анотаціям, які описують найбільш помітний об'єкт або дію. Ці описи можна розглядати як перефразування один одного, оскільки вони зазвичай близькі за змістом. Є дві версії набору даних: 2014 р. та 2017 р. У цьому дослідженні використано версію 2017 р. Для кожного набору перефразувань під час навчання застосовано всі можливі пари. Це дає змогу значно збільшити кількість прикладів для навчання. Для оцінювання якості використано перший опис як джерело, а решту — як посилання.

Набір даних QQP є корпусом для ідентифікації перефразувань. Запитання з вебсайту Quora були позначені модераторами як «дублікат» або «не дублікат». В експериментах використано лише пари, позначені як дублікати. Оскільки в оригінальному наборі даних немає поділу на навчання/розробку/тестування, дотримано поділ, описаний у роботі [8]. Аналогічно для кожної пари використано обидва запитання як перефразування одне одного, тому остаточний набір даних удвічі більший. Під час оцінювання маємо лише одне посилання.

Згенеровано новий набір даних під назвою ParaChatGPT за допомогою великої мовної моделі ChatGPT. Для цього використано OpenAI API, модель gpt-3.5-turbo. Після численних ітерацій вибрано такий остаточний опис задачі: "You are a helpful AI assistant with extensive linguistic knowledge. Rewrite the following text to make it simpler and more straightforward." Як вхідні речення взято випадкові приклади з ParaNMT або речення, самостійно згенеровані моделлю за відповідним запитом. Для кожного вхідного речення мільйон разів виконано два парафрази. У такий спосіб отримано 1 000 000 триплетів.

Огляд набору даних, використаних у цій роботі, наведено в табл. 1.

Таблиця 1. Набори даних для генерації перефразувань

Назва	Анотація	Кількість прикладів	Розмір набору	Кількість пар
ParaNMT	Синтетична	50 000 000	2	100 000 000
QQP	Зроблена людиною	400 000	2	800 000
MSCOCO	Зроблена людиною	123 000	5	2 460 000
ParaChatGPT	Синтетична	1 000 000	3	6 000 000

Метрики оцінювання якості. Оскільки оцінювання генерації тексту зазвичай є складною задачею, в експериментах використано комбінацію метрик. У результатах експериментів наведено стандартні метрики BLEU [9] і TER [10] та семантичну метрику METEOR [11]. У дослідженні [12] показано, що оцінки згенерованих перефразувань, зроблені людиною, добре корелюють з цими метриками.

Щоб забезпечити надійність оцінювання, використано бібліотеку SacreBLEU [13] для обчислення BLEU і TER. Для обчислення METEOR застосовано оригінальний програмний код на Java. Обидві бібліотеки приймають детокенізовані («сирі») дані і у такий спосіб усувають вплив токенізації.

Навчальне налаштування. В експериментах досліджено три типи різних нейронних мереж: повнозгорткову мережу [14], довгу короткочасну пам'ять (LSTM) [15] і трансформер [16]. Всі вони мають майже однакову кількість параметрів. Застосовано спільні ваги для представлення токенів для кодування, декодування та вихідного шару (softmax), оскільки генерація перефразувань є одномовною задачею.

Для моделі типу «трансформер» взято базові налаштування. LSTM-модель складається з 3-х шарів (і для кодування, і для декодування) з прихованим розміром (hidden size) 512 і механізму уваги Луонга (Luong). Повнозгорткова модель має таку структуру: 4 шари згорток з розміром ядра 512 і шириною 3; 2 шари згорток з розміром ядра 1024 і шириною 3; 1 шар згорток з розміром ядра 2048 і шириною 1. Під час навчання використано графік оберненого квадратного кореня з розминкою (warm up). Спочатку моделі навчено на наборах даних ParaNMT-50M та ParaChatGPT, а потім здійснено точне налаштування найкращих контрольних точок на QQP та MSCOCO окремо.

РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТІВ

Результати експериментів наведено в табл. 2. Є багато проблем, пов'язаних з методологією оцінювання задачі генерації перефразувань, наприклад, відмінності у версіях або у розбитті наборів даних, скороченні довжини речень, токенізації. Тому, аби мати можливість порівнювати навчені моделі, використано стратегію оцінювання, подібну до описаної у [17], і проаналізовано отримані результати.

У роботі [17] автори навчають нейронні мережі з попередніх праць щодо генерації парафразу з фіксованими стратегіями навчання та оцінювання. Серед них моделі кодування-декодування, як-от: залишкова LSTM [18] та базова

Таблиця 2. Порівняння моделей типу «трансформер», навчених на ParaNMT та ParaChatGTP (1+2+3), з ParaNMT (1+3), BART та іншими системами з [17]

Система	Набір даних QQP test			Набір даних MSCOCO dev		
	BLEU↑	TER↓	METEOR↑	BLEU↑	TER↓	METEOR↑
Скорочена LSTM	28.4	59.1	30.2	26.9	63.3	24.2
Трансформер	29.1	59.5	30.5	26.9	63.3	24.2
CGMH	22.5	65.0	27.0	17.3	72.6	21.9
MCPG	24.1	64.5	31.8	16.5	73.5	23.2
PTS	25.6	58.7	31.4	17.0	69.9	22.8
BART base	30.5	57.3	32.8	28.2	**57.6**	26.0
Трансформер base (1+2+3)	**32.6**	**56.9**	**35.6**	**28.5**	**57.8**	**26.8**
Трансформер base (1+3)	30.6	57.4	33.2	27.6	**57.6**	26.2

модель-трансформер, а також метод зі слабким наглядом CGMH [19]. Крім того, вони представляють методи пошуку дерев Монте-Карло для генерації перефразувань (MCPG) і пошуку дерев Парето (PTS), де задачу генерації перефразувань розглядають як задачу багатокритерійного пошуку з використанням великої бази даних PPDB 2.0 [20].

Моделі, попередньо навчені послідовно на наборах даних ParaNMT та ParaChatGTP, демонструють суттєво кращу якість перефразування, ніж усі інші моделі. Зокрема це помітно на тестовій частині набору QQP. Водночас різниця у разі навчання на MSCOCO не настільки помітна, можливо через розмір навчальної частини набору даних.

Водночас моделі, попередньо навчені лише на ParaNMT, показують кращі результати (для обох наборів метрик і за всіма метриками), ніж усі інші сучасні моделі. Хоча різниця між результатами, продемонстрованими різними моделями, у разі оцінювання за метрикою BLEU не така вже й велика, базова модель типу «трансформер» показує значне покращення результатів для метрик METEOR і TER (кращі значення метрик у табл. 2 виділено жирним шрифтом). Зауважимо, що попередньо навчена модель типу LSTM не поступається найкращим моделям кодерів-декодерів або навіть перевершує їх.

Щоб порівняти якість перефразування з якістю, яку забезпечують попередньо навчені моделі загального призначення, навчено модель BART [21], яка має архітектуру, подібну до архітектури базової моделі. Обидві складаються з 6 блоків трансформера (як кодування, так і декодування), мають спільні векторні представлення і відрізняються лише стратегією попереднього навчання. Трансформер, попередньо навчений на ParaNMT-50M, демонструє якість, близьку до якості, яку забезпечує точно налаштована модель BART.

Дослідження впливу компонент. У разі переходу до використання попередньо навчених моделей вибір архітектури «трансформер» є типовим (by default) для задач обробки природної мови. Водночас деякі останні дослідження [22] свідчать про те, що не лише трансформери можуть застосовувати знання, отримані на етапі попереднього навчання. У цьому дослідженні розглянуто вплив попереднього навчання для генерації перефразувань на якість роботи моделі залежно від архітектури.

У табл. 3 наведено метрики якості роботи моделей з різними архітектурами для різної кількості кроків попереднього навчання. Найкращі значення у кожному блоці виділено жирним шрифтом. З табл. 3 видно, що попереднє навчання підвищує якість роботи моделі незалежно від архітектури. У деяких випадках повністю згорткові моделі та LSTM перевершують трансформери з погляду приросту якості від попереднього навчання. Цікаво, що середній приріст є більшим на наборі даних MSCOCO (у разі оцінювання за метриками BLEU та TER), незважаючи на те, що його навчальна вибірка більша, ніж у наборі даних QQP.

Також досліджено якість нейронних мереж, які навчалися лише на наборі даних ParaNMT-50M без подальшого точного налаштування (табл. 3, блок з останніх трьох рядків). Для цих моделей оцінка (score) за метрикою METEOR є вищою для набору QQP test і близькою для набору MSCOCO dev. Моделі, навчені на реальних наборах даних, очікувано демонструють вищу якість роботи у разі оцінювання за метриками BLEU та TER.

Зауважимо, що трансформер, який використовує лише ParaNMT, показує стабільно гірші результати на обох наборах даних порівняно з LSTM і повнозгортковою мережею. Однією з можливих причин є те, що завдяки кращому індуктивному зміщенню трансформер краще підлаштовується під набір даних ParaNMT а отже гірше узагальнює інші набори даних.

Таблиця 3. Метрики якості роботи моделей з різними архітектурами для різної кількості кроків попереднього навчання

Модель	Кількість кроків	Набір даних QQP test			Набір даних MSCOCO dev		
		BLEU↑	TER↓	METEOR↑	BLEU↑	TER↓	METEOR↑
Трансформер base	3	**28.7**	**58.6**	31.5	25.2	**60.6**	24.5
LSTM + Луонг	3	27.5	59.7	30.1	25.0	61.1	24.6
Повністю згорткова	3	27.9	59.9	30.7	**25.3**	61.5	**24.9**
Трансформер base	1+3	**30.6**	**57.4**	**33.2**	27.6	**57.6**	**26.2**
LSTM + Луонг	1+3	29.2	58.1	32.6	26.7	59.0	25.5
Повністю згорткова	1+3	29.5	57.8	32.6	**27.7**	**57.8**	25.7
Трансформер base	1+2+3	**32.6**	**56.9**	**35.6**	**28.5**	**57.8**	**26.8**
LSTM + Луонг	1+2+3	30.3	58.0	33.6	27.7	58.4	25.7
Повністю згорткова	1+2+3	30.7	57.3	34.7	27.6	57.9	26.0
Трансформер base	1	23.7	66.7	31.4	17.4	69.9	23.6
LSTM + Луонг	1	24.1	64.1	31.5	18.1	68.7	24.1
Повністю згорткова	1	24.2	65.8	31.4	17.9	69.1	24.2

СУЧАСНИЙ СТАН ДОСЛІДЖЕНЬ

На основі ідеї варіаційного автокодування з дискретними латентними структурами у [23] запропоновано модель латентного мішка слів (BOW) для генерації перефразувань. Семантику дискретної латентної змінної моделюють за допомогою BOW з цільових речень. Цю латентну змінну використовують для побудови повністю диференційованої моделі планування змісту та поверхневої реалізації. Вихідні слова слугують для прогнозування їхніх сусідів і моделювання цільової BOW за допомогою суміші softmax. Репараметризацію Гумбеля top-k використовують для виконання диференційованої вибірки підмножин з прогнозованого розподілу BOW. Отримані вибіркові представлення слів застосовують для доповнення моделі декодування та спрямування його простору пошуку генерації. Прихована модель BOW не тільки покращує процес декодування, але й демонструє чітку інтерпретовність (interpretability) щодо неконтрольованого навчання сусідніх слів та покрокової процедури генерації. На основі вихідного речення програма спочатку генерує сусідні слова з вибірки зі згенерованого BOW (планування), а потім генерує речення (реалізація). Ще однією перевагою моделей латентних змінних є те, що вони дають змогу контролювати кінцевий вивід з латентного коду.

У [24] переформульовано неконтрольоване перенесення стилю у задачу генерації перефразувань. Перший етап описаного підходу передбачає нормалізацію вхідних речень шляхом пропускання їх через різнопланову (diverse) модель перефразування. Результатом цього процесу є набір даних нормалізованих речень, який дає змогу сформувати псевдопаралельний корпус між кожним оригінальним реченням та його перефразованою версією. Цей псевдопаралельний корпус використано для навчання стильової моделі, яка намагається реконструювати оригінальне речення за його перефразованою версією. Оскільки попередньо навчена модель перефразування видаляє ідентифікатори стилю зі своїх вхідних даних, інтуїція цієї моделі зворотного перефразування полягає в тому, що вона вчиться вставляти стилістичні

особливості у процесі реконструкції. Під час виведення довільне речення (в будь-якому конкретному стилі) перетворюється на речення в цільовому стилі з використанням двоетапного процесу нормалізації стилю з подальшою стилізацією за допомогою інверсного перефразувальника. Далі автори допрацьовують великомасштабну попередньо навчену GPT2-модель мови, щоб реалізувати перефразувальник та інверсний перефразувальник для кожного стилю. Останнім етапом застосування підходу є вибір навчальних даних для моделі перефразування. Виявлено, що максимізація лексичної та синтаксичної різноманітності вихідних перефразувань має вирішальне значення для ефективної нормалізації стилю. Вони сприяють різноманітності вихідних даних, навчаючи модель перефразування на агресивно відфільтрованій підмножині ParaNMT-50M.

У дослідженні [25] запропоновано використовувати синтаксичні перетворення для м'якого «перевпорядкування» вихідного речення і керувати нейронною моделлю перефразування. Спочатку, на основі вхідного речення метод виводить набір можливих синтаксичних перестановок, використовуючи модель кодування-декодування. Ця модель оперує частково лексичним, частково синтаксичним представленням речення і може переставляти великі фрагменти. Далі метод використовує кожну запропоновану перестановку для створення послідовності позиційних вставок, що спонукає остаточну модель типу «кодер-декодер», яка здійснює перефразування, звертати увагу на вихідні слова в певному порядку. Спочатку вибирають кортежі фраз, які формують вхідні дані для моделі seq2seq. Кортеж фрази складається з піддерева з абстрагованими складовими (заміненими на їхні синтаксичні категорії). Далі отримують перестановки для кожного кортежу фраз. Автори використовують модель «від послідовності до послідовності» (SOW-модель), яка приймає рядок на вході і видає відповідну вихідну послідовність. Вирівнювання на рівні слів між вхідною та згенерованою вихідною послідовностями (з використанням косинусної подібності між вбудовуваннями GloVe) виконують для того, щоб отримати перегрупування, яке має бути застосоване до вхідної послідовності. На попередньому кроці було отримано перестановку для піддерева. Щоб отримати перестановку на рівні речень, автори спочатку рекурсивно застосовують алгоритм REORDER до піддерев, який повертає верхні k перестановок кожного піддерева. Отримані в такий спосіб представлення на рівні речень оцінюють як середнє арифметичне всіх перестановок на рівні фраз.

У [26] запропоновано метод генерування перефразувань англійських запитань, які зберігають початковий зміст, але використовують іншу поверхневу форму (surface form). Модель кодування-декодування навчено реконструювати запитання з перефразувань із тим самим значенням і зразка з тією самою поверхневою формою, що зумовлює створення відокремлених закодованих просторів. Векторно-квантовану варіаційну модель автокодування використано для представлення поверхневої форми як набору дискретних латентних змінних, що дає змогу застосувати класифікатор для вибору іншої поверхневої форми під час тестування. Запропонований підхід, який називають SEPARATOR, використовує модель кодування-декодування для перетворення вхідного запитання в латентний простір кодування, а потім назад у вихідне перефразування. Принципове вузьке місце з точки зору інформації (a principled information bottleneck) і ретельний вибір схеми навчання зумовлюють створення простору кодування, який окремо представляє намір (intent) і поверхневу форму. Таке розділення дає змогу перефразувати вхідне запитання, змінюючи поверхневу форму вихідної відповіді, безпосередньо маніпулюю-

чи синтаксичним кодуванням вхідних даних і зберігаючи семантичне кодування незмінним. Припущено, що під час навчання є доступ до еталонних кластерів перефразувань, наборів запитань з різними поверхневими формами, які були зіставлені як такі, що мають однакове значення або намір.

У статті [27] досліджено використання структурованих варіаційних моделей автокодування для виведення латентних шаблонів для генерації речень. Запропоновано застосовувати м'яку безперервну «релаксацію» та репараметризацію для навчання. Зокрема, автори пропонують використовувати Gumbel-CRF, тобто безперервну «релаксацію» алгоритму дискретизації CRF на основі розслабленого підходу «фільтрація вперед — дискретизація назад» (FFBS). Як репараметризований градієнтний оцінювач, Gumbel-CRF дає більш стабільні градієнти, ніж оцінювачі на основі функції оцінювання. Щодо мережі структурованого виведення показано, що вона вивчає інтерпретовні шаблоні під час навчання, що дає змогу керувати декодером під час тестування. Ефективність методів продемонстровано в експериментах з неконтрольованої генерації перефразувань.

ВИСНОВКИ

Досліджено вплив попереднього навчання на наборах даних ParaNMT та ParaChatGPT для генерації перефразувань. Набір даних ParaChatGPT згенеровано спеціально для цієї роботи. Запропоновано простий та ефективний підхід до покращення якості нейронних моделей для виконання цієї задачі.

Показано, що послідовне попереднє навчання суттєво підвищує якість нейронних мереж незалежно від їхньої архітектури і забезпечує значно кращий ефект, ніж універсальні попередні навчання, як показано на прикладі моделі BART. Моделі, навчені виключно на ParaNMT, вже демонструють задовільну якість на обох наборах даних. Моделі, попередньо навчені тільки на ParaNMT, працюють з таким самим рівнем якості, що і модель BART.

Використання доступних попередньо навчених моделей часто обмежує вибір архітектури. Це може суттєво вплинути на такі важливі параметри, як час генерації тексту та пам’ять графічного процесора.

Релевантне попереднє навчання підвищує якість роботи нейронних мереж без жодних додаткових втрат з погляду розміру моделі або часу виведення. Методи попереднього навчання без урахування специфіки задачі потребують значних обчислювальних ресурсів, а доступні моделі обмежені архітектурою. Водночас попереднє навчання, орієнтоване на конкретну задачу, значно покращує якість роботи моделі і є простішим у підготовці.

СПИСОК ЛІТЕРАТУРИ.

1. Han X., Zhang Z., Ding N., Gu Y., Liu X., Huo Y., Qiu J., Yao Y., Zhang A., Zhang L., et al. Pre-trained models: past, present and future. *AI Open.* 2021. Vol. 2. P. 225–250. https://doi.org/10.1016/j.aiopen.2021.08.002.
2. Zhao W., Wang L., Shen K., Jia R., Liu J. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. https://doi.org/10.48550/arXiv.1903.00138.
3. Omelianchuk K., Atrasevych V., Chernodub A., Skurzhanskyi O. GECToR–Grammatical error correction: tag, not rewrite. arXiv:2005.12592v2 [cs.CL] 29 May 2020. https://doi.org/10.48550/arXiv.2005.12592.
4. Kasai J., Pappas N., Peng H., Cross J., Smith N.A. Deep encoder, shallow decoder: reevaluating non-autoregressive machine translation. 2020. arXiv:2006.10369v4 [cs.CL]. 24 Jun 2021. https://doi.org/10.48550/arXiv.2006.10369.

5. Wieting J., Gimpel K. ParaNMT-50M: pushing the limits of paraphrastic sentence embeddings with millions of machine translations. arXiv:1711.05732v2 [cs.CL] 20 Apr 2018. https://doi.org/10.48550/arXiv.1711.05732.

6. Ouyang L., Wu J., Jiang X., Almeida D. et al. Training language models to follow instructions with human feedback. arXiv:2203.02155v1 [cs.CL] 4 Mar 2022. https://doi.org/10.48550/arXiv.2203.02155.

7. Lin T.-Y., Maire M., Belongie S., Bourdev L., Girshick R., Hays J., Perona P., Ramanan D., Zitnick C.L., Dollár P. Microsoft COCO: common objects in context. arXiv:1405.0312v3 [cs.CV] 21 Feb 2015. https://doi.org/ 10.48550/arXiv.1405.0312.

8. Wang Z., Hamza W., Florian R. Bilateral multi-perspective matching for natural language sentences. 2017. arXiv:1702.03814v3 [cs.AI] 14 Jul 2017. https://doi.org/10.48550/arXiv.1702.03814.

9. Papineni K., Roukos S., Ward T., Zhu W.-J. Bleu: a method for automatic evaluation of machine translation. *Proc. 40th annual meeting on Association for Computational Linguistics* (7–12 July 2002, Philadelphia, Pennsylvania, USA). Philadelphia, 2002. P. 311–318. https://doi.org/10.3115/1073083.1073135.

10. Snover M., Dorr B., Schwartz R., Micciulla L., Makhoul J. A study of translation edit rate with targeted human annotation. *Proc. 7th Conference of the Association for Machine Translation in the Americas: Technical Papers* (8–12 August 2006, Cambridge, Massachusetts, USA). Cambridge, 2006. P. 223–231. URL: https://aclanthology.org/2006.amta-papers.25.

11. Lavie A., Agarwal A. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. *Proc. Second Workshop on Statistical Machine Translation* (June 2007, Prague, Czech Republic). Prague, 2007. P. 228–231. URL: https://aclanthology.org/W07-0734.pdf.

12. Wubben S., Van Den Bosch A., Krahmer E. Paraphrase generation as monolingual translation: Data and evaluation. *Proc. 6th International Natural Language Generation Conference* (7–9 July, 2010, Trim, Co. Meath, Ireland). Trim, 2010. URL: https://aclanthology.org/W10-4223.pdf.

13. Post M. A call for clarity in reporting BLEU scores. *Proc. the Third Conference on Machine Translation: Research Papers.* (31 October – 1 November 2018, Brussels, Belgium). Brussels, 2018. https://doi.org/10.48550/arXiv.1804.08771.

14. Gehring J., Auli M., Grangier D., Yarats D., Dauphin Y.N. Convolutional sequence to sequence learning. *Proc. 34th International Conference on Machine Learning* (6–11 August 2017, Sydney NSW Australia). Sydney, 2017. PMLR. 2017. Vol. 70. P. 1243–1252. https://doi.org/10.48550/arXiv.1705.03122.

15. Hochreiter S., Schmidhuber J. Long short-term memory. *Neural Computation.* 1997. Vol. 9, Iss. 8. P. 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

16. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser Ł., Polosukhin I. Attention is all you need. *Proc. 31st Conference on Neural Information Processing Systems (NIPS 2017)* (4–9 December 2017, Long Beach, CA, USA). Long Beach, 2017. *Advances in Neural Information Processing Systems.* 2017. Vol. 30. P. 5998–6008. https://doi.org/10.48550/arXiv.1706.03762.

17. Fabre B., Urvoy T., Chevelu J., Lolive D. Neural-driven search-based paraphrase generation. *Proc. 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume.* (19–23 April 2021, virtual event). P. 2100–2111. https://doi.org/10.18653/v1/2021.eacl-main.180.

18. Prakash A., Hasan S.A, Lee K., Datla V., Qadir A., Liu J., Farri O. Neural paraphrase generation with stacked residual LSTM networks. 2016. https://doi.org/10.48550/arXiv.1610.03098.

19. Miao N., Zhou H., Mou L., Yan R., Li L. CGMH: Constrained sentence generation by Metropolis–Hastings sampling. *Proc. 33rd AAAI Conference on Artificial Intelligence (AAAI-19)* (27 January – 1 February 2019, Honolulu, Hawaii, USA). Honolulu, 2019. Vol. 33, N 1. P. 6834–6842. https://doi.org/10.48550/arXiv.1811.10996.

20. Pavlick E., Rastogi P., Ganitkevitch J., Van Durme B., Callison-Burch C. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. *Proc. 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing* (26-31 July 2015, Beijing, China). Beijing, 2015. Vol. 2, Short Papers. P. 425–430. https://doi.org/10.3115/v1/P15-2070.

21. Lewis M., Liu Y., Goyal N., Ghazvininejad M., Mohamed A., Levy O., Stoyanov V., Zettlemoyer L. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Proc. 58th Annual Meeting of the Association for Computational Linguistics* (July 2020, online event). Online event, 2020. P. 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703.

22. Tay Y., Dehghani M., Gupta J., Bahri D., Aribandi V., Qin Z., Metzler D. Are pre-trained convolutions better than pre-trained transformers? *Proc. 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (August 2021, online event). Online event, 2021. Vol. 1: Long Papers. P. 4349–4359. URL: https://aclanthology.org/2021.acl-long.335.pdf.

23. Fu Y., Feng Y., Cunningham J.P. Paraphrase generation with latent bag of words. 2020. https://doi.org/10.48550/arXiv.2001.01941.

24. Krishna K., Wieting J., Iyyer M. Reformulating unsupervised style transfer as paraphrase generation. *Proc. 2020 Conference on Empirical Methods in Natural Language Processing* (EMNLP) (16–20 November 2020, online event). Online event, 2020. https://doi.org/10.18653/v1/2020.emnlp-main.55.

25. Goyal T., Durrett G. Neural syntactic preordering for controlled paraphrase generation. *Proc. 58th Annual Meeting of the Association for Computational Linguistics* (5-10 July 2020, online event). Online event, 2020. https://doi.org/10.18653/v1/2020.acl-main.22.

26. Hosking T., Lapata M.. Factorising meaning and form for intent-preserving paraphrasing. *Proc. 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (1-6 August 2021, virtual event). Virtual event, 2021. Vol. 1: Long Papers. P. 1405–1418. https://doi.org/10.18653/v1/2021.acl-long.112.

27. Fu Y., Tan C., Bi B., Chen M., Feng Y., Rush A. Latent template induction with Gumbel-CRFs. *Proc. Thirty-fourth Conference on Neural Information Processing Systems (NeurIPS2020)* (6-12 December 2020, virtual event). Virtual event, 2020. https://doi.org/10.48550/arXiv.2011.14244.

O.H. Skurzhanskyi, O.O. Marchenko, A.V. Anisimov

SPECIALIZED PRE-TRAINING OF NEURAL NETWORKS ON SYNTHETIC DATA FOR IMPROVING PARAPHRASE GENERATION

Abstract. Generating paraphrases is a fundamental problem in natural language processing. In light of the significant success of transfer learning technology, the “pre-training fine-tuning” approach has become the standard. However, popular general-purpose pre-training methods typically require large datasets and computational resources, and available pre-trained models are limited by fixed architecture and size. We propose a simple and effective approach for pre-training specifically for paraphrase generation, which significantly improves model quality and matches the quality level of general-purpose models. Both existing public data and new data generated by large language models were used. The impact of this procedure on neural networks of different architectures was investigated, and it was shown to work for all of them.

Keywords: artificial intelligence, machine learning, neural networks, paraphrase generation, pre-training, fine-tuning.

Надійшла до редакції 12.10.2023