DOI: https://doi.org/10.36910/6775-2524-0560-2021-45-11
УДК 004.53
Коровій Олександр Сергійович., студент VI-го курсу
Національний Технічний Університет України «Київський Політехнічний Інститут Ігоря Сікорського»

АДАПТАЦІЯ МЕТОДУ DISTILLING KNOWLEDGE В ОБРОБЦІ ПРИРОДНОЇ МОВИ ДЛЯ ТОНАЛЬНОГО ЗАБАРВЛЕННЯ ТЕКСТІВ

Коровій О. С. Адаптація методу Distilling Knowledge в обробці природної мови для аналізу тонального забарвлення текстів. У цій статті описано, як адаптувати прикладний метод «дистиляції знань» для аналізу настроїв для української та російської мов. Показано, як мінімізувати ресурси без втрати значної точності, але прискорити розпізнавання тексту, а також як зменшити витрати на хмарі за допомогою методу «перегонки знань». Для дослідження ми використовували два типи архітектури різних нейронних мереж для обробки природної мови: BERT замість моделей ансамблю та FastText як невелику модель. Поєднання цих двох нейронних мереж (BERT як викладач і FastText як учень) дозволило нам досягти прискорення до 5 разів і без великої точності в задачі аналізу настроїв.
Ключові слова: BERT, FastText, дистиляція знань, нейронна мережа, природня обробка мови, аналіз настроїв

Коровий А. С. Адаптация метода дистилляции знаний в обработке естественного языка для анализа тональности. В этой статье описывается, как адаптировать прикладной метод «дистилляции знаний» для анализа тональности для украинского и русского языков. Показано, как минимизировать ресурсы без потери точности, но ускорить распознавание тональности текста, а также как снизить расходы на облако с помощью метода «дистилляции знаний». Для исследования мы использовали два типа различных архитектур нейронных сетей для обработки естественного языка: BERT вместо ансамблевых моделей и FastText как небольшая модель. Комбинация этих двух нейронных сетей (BERT в качестве учителя и FastText в качестве учащегося) позволила нам добиться ускорения до 5 раз без ущерба для точности в задаче анализа тональности.
Ключевые слова: BERT, FastText, дистилляция знаний, нейронная сеть, обработка языка природы, анализ тональности.

Korovii O. S. Adaptation of distilling knowledge method in Natural Language Processing for sentiment analysis. This paper describes how to adapt an application method of " knowledge distillation " for sentiment analysis for Ukrainian and Russian languages. It is demonstrated how to minimize resources without losing much accuracy, but speeding up the text sentiment recognition, and how to decrease expenses on cloud by using the method of "knowledge distillation". For research we used two types of different neural networks architecture for natural language processing: BERT instead of ensemble models and FastText like a small model. Combination of these two neural networks (BERT as a teacher and FastText as a learner) allowed us to achieve the speedup up to 5 times and without sacrificing much accuracy in sentiment analysis task.
Keywords: BERT, FastText, distill knowledge, neural network, natural language processing, sentiment analysis

Постановка наукової проблеми: Розвиток штучного інтелекту у світі стимулює майже всі області нашого життя. Останні досягнення в області штучного інтелекту, а саме природньої обробки мови, показують актуальність вивчення даної теми. Кожен день хтось робить пошуковий запит в інтернеті, пише відгук, коментар тощо. Це генерує великий об'єм текстової інформації яку потрібно обробляти та зберігати. Але на те щоб обробити великі об'єми текстових даних, штучним інтелектом потрібно дуже багато дороговартісних ресурсів, а саме графічних процесорів. Тому використання процесу «дистиляції знань» дозволяє скоротити витрати на дороговартісні ресурси, і перекласти обчислення нейронних мереж з графічного процесора на центральний. Що в свою чергу набагато дешевше.

Аналіз досліджень. У часи розвитку технологій ми переважно отримуємо інформацію в цифровому вигляді: наукові статті, новини, публікації користувачів у соціальних мережах, електронні листи – все це приклади текстової інформації. Раніше комп'ютери не могли аналізувати природну мову як людську істоту. Але з розвитком штучних нейронних мереж (ШНМ) це стало можливим. Використання ШНМ дозволило комп'ютеру перейти до обробки природної мови на рівні людини, а в певні моменти навіть перевершити її: зрозуміти текст на рівні значення, а не лише на рівні окремих слів [1]. Але з більшими досягненнями та новими можливостями для нас з високою точністю та повнотою вирішувати всі проблеми в області обробки природної мови, ми створили кілька великих моделей, щоб обробляти велику кількість необроблених даних, а це потребувало більше обчислювальних ресурсів. Сьогодні ця проблема є важливою для різних компаній, стартапів тощо. Сьогодні багато сучасних моделей вимагають графічного процесора для етапу навчання та розгортання. Він споживає багато електроенергії і створює високий вуглецевий слід. Для навчання глибоких нейронних мереж (кількість нейронів перевищує 100 мільйонів) потрібно використовувати хмарну технологію, а саме оренду цілого кластера з високопотужними графічними процесорами. Наприклад, автори архітектури сучасної нейронної мережі під назвою Transformer використовували для запуску моделі сервер з вісьмома графічними процесорами P100 [2]. Крім того, у нас є істотна

проблема з відсутністю графічних процесорів, тому що багато людей купують потужні графічні процесори та використовують їх у блокчейні для майнінгу криптовалюти та цифрових монет. Дослідження пропонує адаптацію методу «дистиляції знань» до обробки природної мови для вирішення завдань аналізу настроїв для української та російської мов, а також зменшення обчислювальних ресурсів, збільшення швидкості обчислень та перенесення всіх обчислень з GPU на CPU. Аналіз настроїв [3-5] не нова тема, але для української та російської мов ми не знайшли жодного дослідження, пов'язаного з цією темою. Крім того, це велике завдання, щоб показати, як працює метод «дистиляції знань» у сфері обробки природної мови.

Виклад основного матеріалу й обґрунтування отриманих результатів. В оригінальній статті «distill knowledge with neural networks» [6] автори використовували ансамбль нейронних мереж як «вчителя», але в дослідженні ми замінили модель ансамблю на велику глибоку нейронну мережу, засновану на архітектурі двонаправлених репрезентацій кодера від трансформера (англ. Bidirectional Encoder Representations from Transformers – BERT) [9], яка має близько 300 мільйонів параметрів навчання. Сьогодні BERT є найсучаснішою моделлю для моделювання природної мови і може використовуватися для вирішення багатьох інших проблем обробки природної мови: аналіз настроїв, тексти класифікації, розпізнавання іменованих об'єктів, запитання та відповіді тощо. Для невеликої моделі в якості «учень» обрано архітектуру неглибокої нейронної мережі FastText [7], яка має близько 1 мільйона параметрів навчання. Архітектура FastText адаптована для вирішення завдань класифікації в обробці природної мови «учень» модель має невеликий розмір і не потребує графічних процесорів для навчання і може бути розрахована на ЦП. Завдання, над яким ми використовуємо дистиляцію знань, це аналіз настроїв тексту української та російської мов. За інформацією проекту Tatoeba [11] українська мова ще не вирішила багатьох завдань у сфері обробки природної мови. Тому ми вибираємо українську та російську, вони схожі, і багато людей розмовляють обома мовами.

У цій роботі в якості вхідних текстів були використані новини. Завантажено близько ~180 тисяч новин з відкритих інтернет-ресурсів українською та російською мовами. Проте близько ~17 тисяч новин для вивчення було позначено такими мітками: негативні, нейтральні та позитивні. Усі інші новини, тобто ~163 тисячі новин, використані як синтетичні дані. Дані були позначені великою глибокою нейронною мережею BERT для створення синтетичного набору даних і передачі знань для неглибокої нейронної мережі на основі архітектури FastText.

Bidirectional Encoder Representations from Transformers (BERT) - це глибока нейронна мережа, розроблена для попереднього навчання глибокого двонаправленого кодера на основі архітектури Transformer [12]. BERT - це глибока мовна нейронна мережа, яка попередньо навчена на немаркованих текстових даних, щоб розуміти контекст зліва направо і справа наліво (двонаправлена модель). Інновацією цієї моделі є використання кодера з так званою трансформаторною архітектурою, заснованою на технології уваги [2]. Попереднє навчання цієї мережі проводилося на великій кількості текстів різними мовами. Також був використаний новий підхід для навчання мовній моделі. Основна ідея цього підходу для попереднього навчання полягає в тому, що речення надсилається на вхід нейронної мережі, але одне слово маскується спеціальним маркером, і нейронна мережа намагається передбачити слово, яке має бути замість лексеми, тобто це різновид навчання без вчителя і не вимагає попередньо розмічених даних. Такий підхід дозволяє дати мовній моделі «розуміння на рівні контексту», а не лише на рівні слів.

FastText - невелика модель, з невеликою кількістю параметрів для навчання. Він ефективний для перевірки гіпотез у задачі класифікації тексту (у нашій задачі ми класифікуємо текст на три мітки: негативний, нейтральний, позитивний). Ефективність цієї ШНМ полягає в її невеликій кількості параметрів для навчання [13]. Об'єкти кодуються у векторах та усереднюються на вході прихованого рівня. Завдяки простій архітектурі ми можемо навчити нейронну мережу для більш ніж мільярда слів на ЦП менш ніж за 10 хвилин. Це дає перевагу різноманітним дослідженням в області штучного інтелекту для отримання базової лінії точності при навчанні ШНМ, що може бути використано для подальшого ускладнення нейронної мережі та підвищення точності її роботи. Під час навчання ця нейронна мережа має кілька властивостей, які підвищують точність її роботи. FastText можна викладати як у словах, так і в N-грамах, тобто коли на одному вході ми одночасно надаємо закодовану фразу з кількох слів і звичайних слів, як приклад речення, яке буде закодовано в біграмах: «Привіт, Я штучний інтелект», на вході ми отримаємо такий список табличок: «Привіт, я», «Я штучний», «штучний інтелект». Для трьох грам в одному знаку вже буде три слова і т. д. це збільшує кількість вхідних функцій і розмір словника, що зберігається, але водночас підвищує точність правильної класифікації тексту [13].

Для дослідницьких цілей ми використали власний набір даних з онлайн-новин українською та російською мовами. Набір даних, який використовується для навчання великої нейронної мережі –

BERT, за нашими термінами називається «учитель». Для задачі аналізу настроїв дані, попередньо підготовлені та позначені трьома мітками: негативні (Negative), нейтральні (Neutral), позитивні (Positive). Наступним кроком було розділити цю вибірку на три підвибірки:

1. для навчання - у цьому зразку навчалася нейронна мережа;
2. для перевірки - у цьому зразку ми перевіряли точність нейронної мережі та коригували її параметри в процесі навчання;
3. для тестування - у цьому зразку ми вже перевіряли точність кінцевого результату нейронної мережі.

Кількість текстів, уже позначених новинами для кожної мітки та підмножин, показано в таблиці 1.

Таблиця 1. Порівняння кількості елементів у кожній підмножині для навчання, перевірки та тестового набору даних для великої нейронної мережі – BERT.

Назва мітки	**Кількість елементів для навчання**	**Кількість елементів для валідації**	**Кількість елементів для тестування**
Negative	4627	243	269
Neutral	6334	352	351
Positive	4135	244	219

Для навчання та тестування, призначених для невеликої нейронної мережі – архітектури FastText, яка називається «учень» модель, ми створили синтетичний набір даних, який після навчання промаркована моделлю «учитель». Крім того, ми розділили синтетичний набір даних на дві підмножини: для навчання та для тестування. Усі дані розподілу на підмножини наведені в таблиці 2.

Таблиця 2. Розподіл даних за кожною міткою настроїв у синтетичному наборі даних, який був згенерований великою моделлю BERT для навчання та тестування, виготовленої для невеликої нейронної мережі – FastText, яка називається «учень».

Назва мітки	**Кількість елементів для навчання**	**Кількість елементів для тестування**
Negative	23630	1294
Neutral	117237	3102
Positive	10393	566

Щоб навчити архітектуру BERT, названа «учитель», для завдань аналізу настроїв, ми повинні змінити архітектуру глибокої нейронної мережі BERT, оскільки ця нейронна мережа працює в режимі кодера, ми додали один верхній шар прямого розповсюдження, який був необхідний для класифікації. Він складався з трьох вихідних нейронів, кожен із трьох нейронів відповідав отриманій мітці. Оскільки BERT – це глибока нейронна мережа, її практично неможливо навчити на домашньому комп'ютері, який не має графічного процесора і використовується лише короткий час. Тому я використовував безкоштовні хмарні ресурси платформи Kaggle з графічним процесором Nvidia Tesla P100 [11]. Для навчання невеликої нейронної мережі, заснованої на архітектурі FastText, яка називається «студент», ми використовували комп'ютер Apple MacBook Pro 2018, на процесорі Intel Core i7-8850H 2,60 ГГц, 12 ядер ЦП. Для навчання цієї нейронної мережі не потрібно було використовувати хмарну потужність, що є великою перевагою. Тому нам не потрібні будь-які зміни в архітектурі [13].

Тренування великої нейронної мережі відбувалося епохами (тобто кількість разів, коли навчальні дані проходять через нейронну мережу), оскільки BERT є досить глибокою нейронною мережею, заснованою на оригінальній роботі [12], для точного налаштування BERT для завдання класифікації достатньо використовувати не більше 5 епох. Як видно з графіка, показаного на рисунку 1, після 1-ї епохи, точність навчальної вибірки та валідаційної вибірки

однакові. Після 1-ї епохи точність зростає, але не суттєво. З цього графіка можна зробити висновок, що для навчання нейронної мережі архітектури BERT для завдання аналізу настроїв достатньо 2 епох. Точність після 5-ї епохи у вибірці перевірки становить 0,8235995232419546. Тепер для порівняння ми протестували нейронну мережу на тестовому зразку. Точність на тесстовій вибірці становила 0,8224076281287246. При порівнянні точності на останньому валідаційному та тестовому зразках результат відрізнявся на 0,001 пунктах, що свідчить про високу якість навчання глибокої

нейронної мережі та відсутність ефекту перенавчання. Більш детальні результати після тренування показані на рисунку 2.

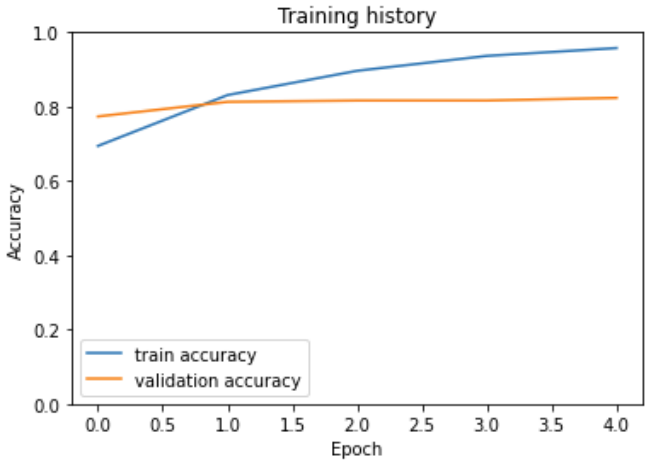

Рисунок 1. Залежність точності від вибірки навчання та валідації в кожній епосі. Епоха почала рахуватися від 0 до 4. Вісь X відображає кількість епох. Вісь Y відображає точність. Синя лінія відповідає точності тренування, після кожної епохи, обчислена точність для всього набору навчальних даних. Жовта лінія – точність валідації, тобто точність розрахунку на валідаційних зразках.

	precision	recall	f1-score
negative	0.81	0.85	0.83
neutral	0.83	0.74	0.78
positive	0.82	0.92	0.87
accuracy			0.82
macro avg	0.82	0.84	0.83
weighted avg	0.82	0.82	0.82

Рисунок 2. Звіт про класифікацію на тестовій вибірці для кожної мітки настроїв для моделі BERT. Точність є найбільшою для нейтральної мітки, а відкликання найкраще виражено у позитивній мітці. Якщо розглядати значення *f1-score* (це середнє значення між точністю і повнотою), то можна сказати, що глибока нейронна мережа найкраще ідентифікує позитивні тексти, тоді негативні, і більшість проблем викликали – нейтральні, вона зробила там найбільше помилок.

Після навчання невеликої нейронної мережі ми провели тестування на синтетичних даних, які раніше були позначені моделлю «учитель». Результати показані на рисунку 3.

	precision	recall	f1-score
negative	0.81	0.61	0.70
neutral	0.79	0.93	0.86
positive	0.73	0.42	0.53
accuracy			0.79
macro avg	0.78	0.65	0.69
weighted avg	0.79	0.79	0.78

Рисунок 3. Звіт про класифікацію на тестовій підмножині із синтетичних даних для кожної мітки настроїв для моделі FastText, яка називається «учень». Підсумкова точність моделі - 0,79081. Точність вища для негативної мітки, але повнота вище для нейтральної мітки, *f1-score* для нейтральної мітки є найбільшим, але для позитивної мітки - 0,53, що є досить поганим результатом, це вказує на те, що нейронна мережа буде найчастіше помиляються у визначенні позитивних новин.

Виходячи з результатів, наведених у таблиці 3, ми бачимо різницю між великою нейронною мережею, заснованою на архітектурі BERT, яка називається «учитель», і маленькою нейронною мережею, заснованою на архітектурі FastText, яка називається «учень». У результаті модель «вчителя» показала точність - 0,822, а модель «учня» - 0,79. Ми втратили приблизно 0,022 пунктів точності, що є хорошим результатом, коли використовували процес «дистиляції знань». Але якщо порівнювати більш детально для кожної мітки, то модель BERT має кращу точність і повноту, для позитивних і негативних

міток, і майже не помиляється, коли текст негативний, не визначає його як негативний, і навпаки. У моделі FastText було більше помилок у цьому аспекті. На основі результатів, показаних на рис. 2 і рис. 3, можна зробити висновок, що модель BERT може розуміти контекст тексту, а не тільки наявність певних слів, а модель FastText зосереджується тільки на словах.

Якщо порівняти продуктивність цих двох моделей, то модель FastText в 5 разів швидше, ніж модель BERT. Використовуючи тільки ресурси процесора Intel Core i7 (12 ядер), модель FastText показала продуктивність близько 1000 текстів в секунду, розмір тексту ~ 1 КБ. Модель BERT, використовуючи ресурси графічного процесора Nvidia Tesla P100, показала швидкість близько 200 текстів в секунду. Якщо порівняти вартість оренди сервера CPU і GPU, то різниця становить близько 680 євро на місяць. 12-ядерний CPU-сервер коштує 120 євро на місяць, а GPU-сервер з одним GPU – 800 євро на місяць [15].

Таблиця 3. Порівняння моделей «учень» і «вчитель».

	FastText («учень») мала модель	**BERT («учитель») Громіздка модель**
Negative/Neutral/ Positive *f1*-score кожної мітки	0.70/0.86/0.53	0.83/0.78/0.87
Точність	0.790	0.822
Розмір моделі	300 Mbytes	1.89 Gbytes
Споживання RAM	1.2 Gbytes	6 Gbytes (GPU)
Продуктивність, кількість новин за сек (1 article ≈ 1KB)	~ 1000 on CPU (Core i7, 12 Cores)	~ 200 on GPU (Nvidia Tesla P100), ~ 25 on CPU
Вартість оренди сервера, в місяць	120€	800€

Висновки та перспективи подальшого дослідження. У роботі було проведено дослідження з використанням методу «дистиляції знань» для вирішення проблеми природно-мовної обробки аналізу настроїв тексту. Використовувалися дві архітектури нейронних мереж: модель BERT як викладач і модель FastText як студент. Ми створили власний набір даних, щоб показати, як нейронні мережі можуть допомогти у вирішенні завдань аналізу настроїв українською та російською мовами. BERT був обраний через цю сучасну мовну модель і FastText – найшвидшу нейронну мережу для класифікації тексту. Запропонований підхід показав, як досягти кореляції між високою швидкістю обчислень і точністю, використовуючи ці два типи нейронних мереж, втративши певну точність, але отримавши високу швидкість обчислень і знизивши вартість оренди хмарних екземплярів. Використовуючи цей підхід для аналізу настроїв, ми втратили 0,022 бали точності, але збільшили швидкість обчислення нейронної мережі на ЦП у 5 разів. Для майбутньої роботи ми реалізуємо «перегін знань» для завдання розпізнавання іменованих об'єктів, таким же способом, який ми запропонували в статті для аналізу настроїв. Загалом, використання методу «дистиляції знань» дозволяє компаніям і стартапам збільшити швидкість визначення аналізу настроїв на етапі розгортання.

References.

1. Goodfellow, I., Bengio, Y. and Courville, A., 2016. Deep learning. Cambridge (EE. UU.): MIT Press.
2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L. and Polosukhin, I., 2021. Attention Is All You Need. [online] arXiv.org. Available at: https://arxiv.org/abs/1706.03762
3. Abdur Rahman, Mobashir Sadat, Saeed Siddik, "Sentiment Analysis on Twitter Data: Comparative Study on Different Approaches", International Journal of Intelligent Systems and Applications(IJISA), Vol.13, No.4, pp.1-13, 2021. DOI: 10.5815/ijisa.2021.04.01
4. Golam Mostafa, Ikhtiar Ahmed, Masum Shah Junayed, "Investigation of Different Machine Learning Algorithms to Determine Human Sentiment Using Twitter Data", International Journal of Information Technology and Computer Science(IJITCS), Vol.13, No.2, pp.38-48, 2021. DOI: 10.5815/ijitcs.2021.02.04

5. Khalid Mahboob, Fayyaz Ali, Hafsa Nizami, "Sentiment Analysis of RSS Feeds on Sports News – A Case Study", International Journal of Information Technology and Computer Science(IJITCS), Vol.11, No.12, pp.19-29, 2019. DOI: 10.5815/ijitcs.2019.12.02
6. Hinton, G., Vinyals, O. and Dean, J., 2021. Distilling the Knowledge in a Neural Network. [online] arXiv.org. Available at: https://arxiv.org/abs/1503.02531
7. C. Buciluˇa, R. Caruana, and A. Niculescu-Mizil. Model compression. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '06, pages 535–541, New York, NY, USA, 2006. ACM.
8. N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1):1929–1958, 2014.
9. Dalal AL-Alimi, Yuxiang Shao, Ahamed Alalimi, Ahmed Abdu, "Mask R-CNN for Geospatial Object Detection", International Journal of Information Technology and Computer Science(IJITCS), Vol.12, No.5, pp.63-72, 2020. DOI: 10.5815/ijitcs.2020.05.05
10. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, 2016. You Only Look Once: Unified, Real-Time Object Detection. [online] arXiv.org. Available at: https://arxiv.org/abs/1506.02640
11. Tatoeba: Collection of sentences and translations, 2021. [online] tatoeba.org. Available at: https://tatoeba.org/en/
12. Devlin, J., Chang, M., Lee, K. and Toutanova, K., 2021. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. [online] arXiv.org. Available at: https://arxiv.org/abs/1810.04805
13. Joulin, A., Grave, E., Bojanowski, P. and Mikolov, T., 2021. Bag of Tricks for Efficient Text Classification. [online] arXiv.org. Available at: https://arxiv.org/abs/1607.01759
14. Kaggle: Your Machine Learning and Data Science Community, 2021. [online] Available at: https://www.kaggle.com/
15. Scaleway. 2021. Cloud, Compute, Storage and Network models and pricing. [online] Available at: https://www.scaleway.com/en/pricing/