УДК 004.891
DOI: https://doi.org/10.17721/1812-5409.2024/1.28

Віталій ВРУБЛЕВСЬКИЙ, асп.
ORCID ID: 0009-0005-7070-9001
e-mail: vitalii.vrublevskyi@gmail.com
Київський національний університет імені Тараса Шевченка, Київ, Україна

МОДЕЛЬ ТРАНСФОРМЕРА З ВИКОРИСТАННЯМ ДЕРЕВА ЗАЛЕЖНОСТЕЙ ДЛЯ ІДЕНТИФІКАЦІЇ ПАРАФРАЗ

Побудова моделей для представлення семантики слів, речень і текстів природної мови є ключовою проблемою у комп'ютерній лінгвістиці та штучному інтелекті. Використання якісних векторних представлень слів змінило підходи до оброблення й аналізу природної мови, оскільки слова є основою мови. Дослідження векторних представлень речень також мають велике значення, оскільки вони спрямовані на захоплення семантики та значень речень. Поліпшення цих представлень допомагає краще розуміти текст на глибшому рівні і сприяє розв'язанню різноманітних завдань.

Статтю присвячено розв'язанню задачі ідентифікації парафраз, використовуючи моделі на основі архітектури трансформера. Зазначені моделі продемонстрували високу ефективність на різноманітних завданнях. Було досліджено, що їхня точність може бути підвищена шляхом збагачення моделі додатковою інформацією. Використання синтаксичної інформації, такої як теги частин мови або лінгвістичні структури, може поліпшити розуміння моделлю контексту та структури речення. Збагачення моделі таким способом дає змогу здобути ширший контекст, підвищити адаптивність і продуктивність у різних завданнях обробки природної мови, надаючи їй більшої універсальності для різних застосувань. З огляду на це було запропоновано модель на основі архітектури трансформера з використанням дерева залежностей. Досліджено її ефективність та зіставлено з іншими моделями тієї ж архітектури, використовуючи задачу ідентифікації парафраз. Продемонстровано поліпшення запропонованої моделі в точності та повноті порівняно з оригінальною моделлю (DeBERTa).

У майбутньому доцільними є дослідження використання зазначеної моделі для інших прикладних задач (як-от перевірка на плагіат, визначення авторського стилю) та в оцінці інших графових структур для репрезентації речення (напр. AMR, граф).

Ключові слова: ***оброблення природної мови, ідентифікація парафраз, машинне навчання, дерево залежностей, архітектура трансформера.***

AMS 2020 класифікація: 68T50, 05C90.

Вступ

Побудова моделей для представлення семантики слів, речень і текстів природної мови є ключовою проблемою у комп'ютерній лінгвістиці та штучному інтелекті. Такі моделі, як BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), AlBERT (Lan et al., 2020), DeBERTa (He et al., 2021) та інші змінили обробку природної мови. Вони були створені великими IT-компаніями й обчислені на масштабних обчислювальних ресурсах. Ці моделі демонструють високу точність і стали новим золотим стандартом для багатьох завдань обробки природної мови.

Дослідження векторних представлень речень має велике значення, оскільки вони спрямовані на захоплення семантики та значень речень. Поліпшення цих представлень допомагає краще розуміти текст на глибшому рівні і сприяє розв'язанню різноманітних завдань.

Використання якісних векторних представлень слів змінило підходи до оброблення й аналізу природної мови, оскільки слова є основою мови. Дослідження в цій сфері може вплинути на розвиток інших напрямків, таких як машинний переклад, класифікація тексту та пошук інформації.

Починаючи з моделей, що представляють семантику слів, дослідники вдосконалюють способи кодування контексту навколо слова. Проте сучасні методи, такі як CBOW та Skip-gram (Mikolov et al., 2013), не завжди враховують складні зв'язки між словами всередині речення, що може негативно позначитися на точності моделей, особливо у випадках аналітичних мов, де порядок слів важливий.

Оцінка векторних представлень речень є складним завданням, оскільки відсутній один ідеальний стандарт для порівняння. Для цього дослідники зазвичай використовують непрямі методи оцінки, що ґрунтуються на перевірці можливостей використання цих векторів для виконання різних прикладних завдань у лінгвістиці. У цьому дослідженні ми зосередимося на завданні ідентифікації парафраз.

Об'єктом дослідження є методи побудови векторних представлень речень природної мови.

Мета дослідження – створення методів побудови векторного представлення речення на основі моделей з архітектурою трансформера; дослідження використання графових структур репрезентації речення та їх застосування.

Для досягнення поставленої мети сформовано такі завдання: створити модель для побудови представлення речення з урахуванням його структури.

Застосування мовних моделей постійно розширюється, що підкреслює важливість досліджень у цій сфері. Це дає змогу розвивати моделі, що здатні ефективно розуміти текст у різних мовах і контекстах. Результати пропонованої роботи можуть бути застосовані для будь-якої мови з вираженою структурою речення.

Ідентифікація парафраз означає визначення, чи мають два речення однакове або схоже значення, незважаючи на їхнє різне формулювання. Прикладами парафразу можуть бути такі речення:

- *Хлопець переходив дорогу до школи, тримаючи в руках великий букет троянд.*
- *Іван йшов до школи з великим букетом квітів.*

У сфері оброблення природної мови ідентифікацію парафразу часто трактують як завдання бінарної класифікації. У цьому випадку вхідними даними є пара речень, а виходом – булеве значення, що вказує, чи є два речення парафразами, чи ні.

1. Огляд літератури. Методи виявлення парафраз значно вдосконалилися завдяки прогресу в обробленні природної мови і машинному навчанні. Розглядаючи цю еволюцію, можна виділити деякі етапи або класи методів – лексична і семантична подібність та машинне навчання.

ISSN 1728-3817

Дослідники використовували лексичну та семантичну подібність, застосовуючи тезауруси, такі як WordNet (Fellbaum, 1998) і латентний семантичний аналіз (LSA) (Landauer, Foltz, & Laham, 1998), для ідентифікації парафраз. Ці методи базувалися на ідеї, що слова зі схожими значеннями зазвичай зустрічаються в подібних контекстах.

Відомі вчені, зокрема А. Анісімов та О. Марченко, у своїх роботах поєднали онтологічну структуру WordNet та семантично-синтаксичні зв'язки між словами у реченні за допомогою методу невід'ємної факторизації тензорів для визначення семантичної близькості (Anisimov, Marchenko, & Vozniuk, 2014; Marchenko, 2016). Ці дослідження можуть бути покладені в основу ідентифікації парафраз, якщо використати отримані результати та застосувати їх на рівні речення.

З розвитком машинного навчання моделі на основі рекурентних мереж і трансформерів (такі як BERT) почали використовувати для більшості прикладних задач в обробленні природної мови. Цікаво розглянути моделі, що поєднували переваги цих обох класів.

Щоб модель краще розуміла контекст і структуру речення, наприклад, часто використовують синтаксичну інформацію, теги частин мови або лінгвістичні структури. Це може забезпечити додатковий контекст, поліпшуючи розуміння моделлю вхідного тексту.

Збагачуючи модель додатковою інформацією, вона отримує ширший контекст, підвищену адаптивність і підвищену продуктивність у широкому спектрі завдань, що робить її більш універсальною та здатною в різних програмах для розуміння та оброблення природної мови.

SyntaxBERT – це варіант BERT, спеціально розроблений для захоплення синтаксичної інформації в тексті (Bai et al., 2021). На відміну від традиційних моделей BERT, що зосереджені на контекстному розумінні, SyntaxBERT акцентує на розумінні структурних залежностей і граматичних зв'язків між словами в реченнях. Він використовує методи синтаксичного аналізу, щоб поліпшити свою здатність розуміти та представляти ці зв'язки. Це дає змогу детальніше розуміти структуру речень.

Залучаючи синтаксичну інформацію у процес навчання, SyntaxBERT прагне поліпшити завдання, які значною мірою покладаються на граматичні структури, такі як розбір, машинний переклад і відповіді на запитання. Ця модель намагається вловити не лише семантичне значення, але й синтаксичні нюанси, наявні у природній мові, допомагаючи таким способом у завданнях, що потребують глибшого розуміння композиції та структури речення.

Ми пропонуємо використати дерево залежностей та зв'язки між словами у ньому для цього та застосувати цю інформацію в шарі self – attention.

2. Модель. Існує багато способів презентації формальної структури речення. Більшість з них намагається описати взаємозв'язки між різними словами чи словосполученнями у реченні. Одним з них є дерево залежностей (рис. 1).

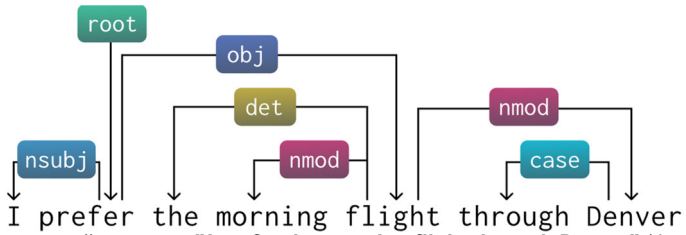

Рис. 1. Дерево залежностей речення "*I prefer the morning flight through Denver*" (Jurafsky, & Martin, 2023)

Дерево залежностей – це орієнтований граф $G = (V, A)$, що задовольняє наступні обмеження (Jurafsky, & Martin, 2023):

1. Є один кореневий вузол, який не має вхідних дуг – $root$.
2. За винятком кореневого вузла кожна вершина має рівно одну вхідну дугу.
3. Існує унікальний шлях від кореневого вузла до кожної вершини у V.

Зв'язки між словами проілюстровані над реченням спрямованими дугами від головного елемента до залежного. Кожна дуга має свій тип і тому це дерево є типізованим деревом залежностей. Вершина $root$ явно позначає корінь дерева.

Проєкт Universal Dependencies – є спробою анотувати залежності й інші аспекти граматики у понад 100 мов (Marneffe et al., 2021). Поточний перелік містить 37 зв'язків залежностей. Деякі з них подано в табл. 1.

Таблиця 1

Типи граматичних залежностей (Jurafsky, & Martin, 2023)

Тип зв'язку	**Опис**	**Приклад** (*Курсивом* виділено головний, **жирним** – залежний елемент)
NSUBJ	Nominal subject	**United** *cancelled* the flight.
OBJ	Direct object	United *diverted* the **flight** to Reno. We *booked* her the first **flight** to Miami
IOBJ	Indirect object	We *booked* **her** the flight to Miami.
NMOD	Nominal modifier	We took the **morning** *flight*.
AMOD	Adjectival modifier	Book the **cheapest** *flight*.
NUMMOD	Numeric modifier	Before the storm JetBlue cancelled **1000** *flights*.
APPOS	Appositional modifier	*United*, a **unit** of UAL, matched the fares.
DET	Determiner	**The** *flight* was cancelled. **Which** *flight* was delayed?
CASE	Prepositions, postpositions and other case markers	Book the flight **through** *Houston*.
CONJ	Conjunct	We *flew* to Denver and **drove** to Steamboat.
CC	Coordinating conjunction	We flew to Denver **and** *drove* to Steamboat.

Моделі на основі архітектури трансформера досягають хороших результатів у різноманітних завданнях. Збагачення BERT і подібних моделей додатковою інформацією може підвищити їхню продуктивність й адаптивність до різних завдань.

Це можна пояснити тим, що для деяких завдань може знадобитися інформація, пов'язана з областю або завданням, які не є представленими у тренувальному корпусі текстів. Доповнення моделі додатковою інформацією, такою як метадані, числові дані або доменно-спеціальні вбудовування, може підвищити її продуктивність у цих завданнях.

Щоб модель краще розуміла контекст і структуру речення, часто використовують синтаксичну інформацію, теги частин мови або лінгвістичні структури. Це може забезпечити додатковий контекст, поліпшуючи розуміння моделлю вхідного тексту.

Завдяки збагаченню додатковою інформацією модель отримує ширший контекст, підвищену адаптивність і підвищену продуктивність у широкому спектрі завдань, що робить її більш універсальною та здатною в різних програмах для розуміння й обробки природної мови.

Ми пропонуємо використати дерево залежностей і зв'язки між словами в ньому для цього та застосувати цю інформацію в шарі self – attention.

Як основну модель з архітектурою трансформера ми обрали DeBERTa (He et al., 2021). Вона базується на моделі RoBERTa (Liu et al., 2019) та містить дві інноваційні методики. По-перше, DeBERTa використовує механізм розмежованої уваги, де кожне слово представлено двома векторами, один кодує його зміст, а інший – його позицію. Вагові коефіцієнти уваги між словами обчислюються за допомогою роз'єднаних матриць на основі вмісту та відносних позицій. По-друге, модель застосовує розширений декодер, що використовує також абсолютні позиції токенів для передбачення замаскованих токенів під час попереднього навчання.

Для вхідної послідовності $s = (s_1, s_2 \dots s_n)$ визначимо граф $< V_s, E_s >$ та матрицю A.

$$dependencyTree(s) = < V_s, E_s > \quad (1)$$

$$a_{ij} = \begin{cases} \delta, & (s_i, s_j) \in E_s \vee (s_j, s_i) \in E_s \\ 1, & otherwise \end{cases}. \quad (2)$$

Коефіцієнт δ – один із гіперпараметрів моделі, буде підібраний під час її оптимізації.

Оновлений шар self – attention:

$$q_i = W_q\, x_i, \quad (3)$$

$$k_i = W_k\, x_i, \quad (4)$$

$$v_i = W_v\, x_i, \quad (5)$$

$$w'_{ij} = \frac{q_i^T k_j}{\sqrt{d_k}}, \quad (6)$$

$$w^*_{ij} = w'_{ij} * a_{ij,} \quad (7)$$

$$w_{ij} = softmax(\, w^*_{ij}), \quad (8)$$

$$y_i = \sum_j w_{ij} v_j, \quad (9)$$

де d_k – це розмірність вектора ключів, y_i – контекст, який використано для репрезентації послідовності відповідно до x_i, W_q, W_k, W_v – це матриці, що дають змогу підрахувати для кожного вектора його значення query, key, value.

Щоб коректно закодувати дерево залежностей, потрібно узяти до уваги особливості різних стратегій токенізації в моделях з архітектурою трансформера (рис. 2).

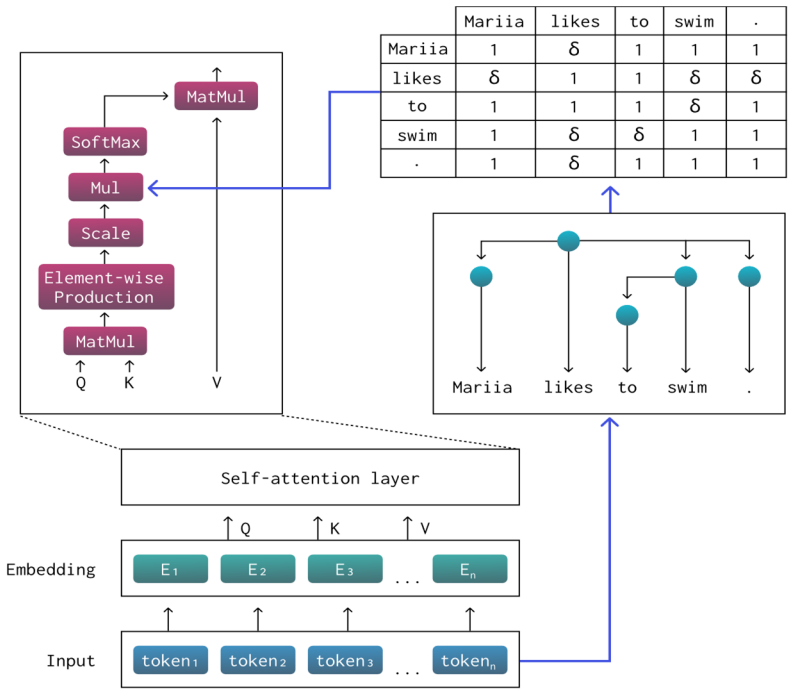

Рис. 2. Архітектура моделі трансформера, доповнена інформацією дерева залежностей

Моделі типу BERT використовують токенізатор типу Word Piece. Він розбиває слова на одиниці, які називають токенами. Так відбувається сегментація слова на менші одиниці, що допомагає ефективніше обробляти рідкісні слова, невідомі слова та морфологічно складні мови. Наприклад, слово "singing", можна розділити на "sing" і "##ing", де "##" означає, що підслово є частиною більшого слова.

Також використовуються спеціальні токен-маркери, щоб допомогти зрозуміти структуру послідовності:

- [CLS] – маркер класифікації. Він передує вхідному тексту та використовується для класифікаційних завдань.
- [SEP] – токен-розділювач. Він розділяє два різні речення або фрази в одному введеному тексті.
- [MASK] – маркер маски. Його використовують під час попереднього навчання для завдань моделювання замаскованої мови, де деякі лексеми випадково замінюються на [MASK], щоб передбачити оригінальні лексеми.

Токенізатори застосовують фіксований словник, отриманий із навчальних даних. Кожен токен у вхідному тексті зіставляється з вектором із цього словника.

Під час оброблення двох речень одночасно (у таких завданнях, як ідентифікація парафраз), моделі типу BERT використовують ідентифікатори сегментів, щоб розрізняти речення. Кожній лексемі присвоюється ідентифікатор сегмента, щоб вказати відповідне речення.

Оскільки модель під час тренування працює зазвичай з певною множиною вхідних даних одночасно (*batch*), потрібно стандартизувати вхідні послідовності до фіксованої довжини. [PAD] – маркери заповнення, додаються до послідовностей, які коротші за максимальну довжину, тоді як довші послідовності можуть бути скорочені, щоб відповідати максимальному розміру вхідних даних моделі. Для того, щоб модель могла розрізняти справжні вхідні дані, від заповнення використовується маска уваги.

Токенізатори відіграють важливу роль у тому, як моделі типу BERT обробляють і розуміють текст, даючи змогу моделі ефективно обробляти вхідні дані. Під час додавання нових ознак до моделі потрібно брати до уваги всі особливості токенізації, щоб правильно представити нові ознаки.

3. Приклад. Розглянемо приклад побудови ознаки, використовуючи дерево залежностей (див. табл. 2). Вхідне речення: "*Mariia likes to swim*". Для початку ознайомимося з базовою токенізацію моделі типу BERT для цього речення (будемо використовувати токенізатор моделі BERT Base Cased (Google-bert/Bert-base-cased…, n. d.))

Таблиця 2

Базова токенізація речення "*Mariia likes to swim*"

Слово	Токен	Токен ID	Сегмент	Маска уваги
	[CLS]	101	0	1
Mariia	Mari	17978	0	1
	##ia	1465	0	1
likes	likes	7407	0	1
to	to	1106	0	1
swim	swim	11231	0	1
.	.	119	0	1
	[SEP]	102	0	1

Надалі потрібно побудувати граф залежностей (рис. 3). Для цього можна використати бібліотеку Spacy (SPACY…, n. d.).

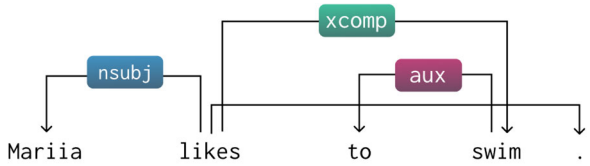

Рис. 3. Дерево залежностей для речення "*Mariia enjoys swimming*"

Базуючись на цьому графі, побудуємо матрицю зв'язності графу (напрямом для спрощення моделі знехтуємо) (див. табл. 3).

Таблиця 3

Матриця зв'язності речення "*Mariia likes to swim*"

Слово	Mariia	likes	to	swim	.
Mariia	0	1	0	0	0
likes	1	0	0	1	1
to	0	0	0	1	0
swim	0	1	1	0	0
.	0	1	0	0	0

Маючи цей граф, можна побудувати повну матрицю відповідно до формул (1) та (2), ураховуючи особливості токенізації (табл. 4).

Таблиця 4

Матриця ознаки на основі дерева залежностей для речень "*Mariia likes to swim*"

	[CLS]	*Mariia likes to swim*						[SEP]
	[CLS]	Mariia		likes	to	swim	.	[SEP]
	101	17978	1465	7407	1106	11231	119	102
101	1	1	1	1	1	1	1	1
17978	1	1	1	δ	1	1	1	1
1465	1	1	1	δ	1	1	1	1
7407	1	δ	δ	1	1	δ	δ	1
1106	1	1	1	1	1	δ	1	1
11231	1	1	1	δ	δ	1	1	1
119	1	1	1	δ	1	1	1	1
102	1	1	1	1	1	1	1	1

Під час побудови матриці було враховано можливе розбиття одного слова на декілька токенів і службові токени.

4. Результати. Було проведено експеримент для того, щоб порівняти доповнену модель трансформера з ознаками дерева залежностей з базовою моделлю.

Основні показники, що використовуватимуться для порівняння моделей:

- Точність й оцінка F1.
- Розмір моделі.

Усе тонке налаштування для моделей, описаних у цьому розділі, виконано за допомогою NVIDIA Tesla V100 із 16 ГБ оперативної пам'яті GPU та 50 ГБ системної оперативної пам'яті за допомогою платформи Google Collab (Bisong, 2019).

Умови експерименту:

- Кожна модель тонко налаштована лише протягом 10 епох.
- Використані загальні глобальні параметри для точного налаштування моделей.

Кожна модель тонко налаштована три рази і в таблицях результатів повідомлено середнє значення та стандартне відхилення.

Усі моделі були налаштовані з такими загальними параметрами:

- Розмір батча = 8.
- Тонке налаштування (fine-tunning) протягом 10 епох.
- Швидкість навчання lr = 2×10^{-5}, лінійне зменшення протягом кожного кроку.
- Weight decay = 0,01.
- Стратегія оптимізації AdamW (PyTorch).

Для тренування моделей було обрано корпус Microsoft Research Paraphrase Corpus (MRPC) (Dolan, Quirk, & Brockett, 2004). Він містить пари речень, для яких вручну було зазначено, чи є вони парафразами. Він був створений Доланом і Брокеттом у Microsoft Research і став одним із найпоширеніших датасетів для зазначеної задачі. Він містить 5801 пару речень, що взяті з різних джерел, включно зі статтями новин, статтями в енциклопедіях і вебсторінками. Щоб забезпечити якість міток, для кожної пари речень використовували кілька анотаторів, а розбіжності вирішували більшістю голосів.

Ураховуємо, що для моделі, яка використовує дерево залежності, потрібно спочатку підібрати параметр δ (табл. 5). Для цього будемо використовувати дані з корпусу MRPC (validation part).

Таблиця 5

Результати моделі DeBERTa Base + Dependency Tree під час валідації, підбір значення δ

№	Значення δ	Точність під час валідації	F1 під час валідації
1	1,2	89,71	92,81
2	1,1	88,97	92,12
3	0,9	**91,67**	**94,01**
4	0,8	90,44	93,15

За результатами експериментів бачимо, що модель з використанням дерева залежностей має найкращу точність і значення F1 (табл. 6).

Таблиця 6

Порівняння моделей

№	Назва моделі	Точність під час тестування	F1 під час тестування
1	DeBERTa Base	87,52 ± 0,26	90,80 ± 0,12
2	DeBERTa Base + Dependency Tree, δ = 0,9	**87,67 ± 0,56**	**91,12 ± 0,27**

Дискусія і висновки

Проведено аналіз використання дерева залежностей, основи для репрезентації структури речення. Аналізуючи попередні результати використання синтаксичного дерева розбору, дерева залежностей та сучасних моделей на основі архітектури трансформера, запропоновано модифікацію шару self-attention з використанням ознак на основі дерева залежностей. За результатами експериментів розкрито якість моделі. Запропонована модифікація показала кращі показники точності та F1, ніж базова модель.

Головним результатом є експериментальне підтвердження того, що дерево залежностей можуть поліпшити базові моделі з архітектурою трансформера. У майбутньому доцільним є дослідження використання зазначеної моделі для інших прикладних задач (такі, як перевірка на плагіат, визначення авторського стилю тощо) та в оцінці інших графових структур для репрезентації речення (напр., AMR, граф).

Подяка, джерела фінансування. Робота є складовою частиною наукових досліджень, які ведуться на кафедрі математичної інформатики факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка на виконання теми проєкту Міністерства освіти і науки України "Теорія і методи розробки інтелектуальних інформаційних технологій та систем" (НДФ №16КФ015-02, 01.01.2016-31.12.2025).

Список використаних джерел

Anisimov, A. V., Marchenko, O. O., & Vozniuk, T. G. (2014). Determining Semantic Valences of Ontology Concepts by Means of Nonnegative Factorization of Tensors of Large Text Corpora. *Cybern Syst Anal*, 50, 327–337 https://doi.org/10.1007/s10559-014-9621-9

Bai, J., Wang, Y., Chen, Y., Yang, Y., Bai, J., Yu, J., & Tong, Y. (2021). Syntax-BERT: Improving Pre-trained Transformers with Syntax Trees. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume* (pp. 3011–3020). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.262

Bisong, E. (2019). *Building Machine Learning and Deep Learning Models on Google Cloud Platform (1st ed.)*. Apress Berkeley, CA. https://doi.org/10.1007/978-1-4842-4470-8

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* https://doi.org/10.48550/arXiv.1810.04805

Dolan, B., Quirk, C., & Brockett, C. (2004). Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics* (pp. 350–356). COLING.

Fellbaum, C. (1998). *WordNet: An electronic lexical database.* MIT Press. https://doi.org/10.7551/mitpress/7287.001.0001

Google-bert/bert-base-cased - Hugging Face. (n. d.). https://huggingface.co/google-bert/bert-base-cased

He, P., Liu, X., Gao, J., & Chen, W. (2021). *DeBERTa: Decoding-enhanced BERT with Disentangled Attention.* https://doi.org/10.48550/arXiv.2006.03654
Jurafsky, D., & Martin, J. (2023). Speech and Language Processing. An *Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition,* 3. https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf
Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. https://doi.org/10.48550/arXiv.1909.11942
Landauer, T. K., Foltz, P. W., & Laham, D. (1998*)*. An introduction to latent semantic analysis. *Discourse Processes*, 25(2–3), 259–284. https://doi.org/10.1080/01638539809545028
Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., … Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach.* https://doi.org/10.48550/arXiv.1907.11692
Marchenko, O. O. (2016). A Method for Automatic Construction of Ontological Knowledge Bases. I. Development of a Semantic-Syntactic Model of Natural Language. *Cybern Syst Anal,* 52, 20–29. https://doi.org/10.1007/s10559-016-9795-4
Marneffe, M. C., Manning, C., Nivre, J., & Zeman, D. (2021). Universal Dependencies. *Computational Linguistics – Association for Computational Linguistics*, 47(2), 255–308. https://doi.org/10.1162/coli_a_00402
Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. https://doi.org/10.48550/arXiv.1301.3781
SPACY – Industrial-strength Natural language processing in Python. (n. d.). https://spacy.io

References

Anisimov, A. V., Marchenko, O. O., & Vozniuk, T. G. (2014). Determining Semantic Valences of Ontology Concepts by Means of Nonnegative Factorization of Tensors of Large Text Corpora. *Cybern Syst Anal*, 50, 327–337 https://doi.org/10.1007/s10559-014-9621-9
Bai, J., Wang, Y., Chen, Y., Yang, Y., Bai, J., Yu, J., & Tong, Y. (2021). Syntax-BERT: Improving Pre-trained Transformers with Syntax Trees. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume* (pp. 3011–3020). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.262
Bisong, E. (2019). *Building Machine Learning and Deep Learning Models on Google Cloud Platform (1st ed.)*. Apress Berkeley, CA. https://doi.org/10.1007/978-1-4842-4470-8
Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* https://doi.org/10.48550/arXiv.1810.04805
Dolan, B., Quirk, C., & Brockett, C. (2004). Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics* (pp. 350–356). COLING.
Fellbaum, C. (1998). *WordNet: An electronic lexical database.* MIT Press. https://doi.org/10.7551/mitpress/7287.001.0001
Google-bert/bert-base-cased - Hugging Face. (n. d.). https://huggingface.co/google-bert/bert-base-cased
He, P., Liu, X., Gao, J., & Chen, W. (2021). *DeBERTa: Decoding-enhanced BERT with Disentangled Attention.* https://doi.org/10.48550/arXiv.2006.03654
Jurafsky, D., & Martin, J. (2023). Speech and Language Processing. An *Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition,* 3. https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf
Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. https://doi.org/10.48550/arXiv.1909.11942
Landauer, T. K., Foltz, P. W., & Laham, D. (1998*)*. An introduction to latent semantic analysis. *Discourse Processes*, 25(2–3), 259–284. https://doi.org/10.1080/01638539809545028
Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., … Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach.* https://doi.org/10.48550/arXiv.1907.11692
Marchenko, O. O. (2016). A Method for Automatic Construction of Ontological Knowledge Bases. I. Development of a Semantic-Syntactic Model of Natural Language. *Cybern Syst Anal,* 52, 20–29. https://doi.org/10.1007/s10559-016-9795-4
Marneffe, M. C., Manning, C., Nivre, J., & Zeman, D. (2021). Universal Dependencies. *Computational Linguistics – Association for Computational Linguistics*, 47(2), 255–308. https://doi.org/10.1162/coli_a_00402
Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. https://doi.org/10.48550/arXiv.1301.3781
SPACY – Industrial-strength Natural language processing in Python. (n. d.). https://spacy.io

Отримано редакцією журналу / Received: 15.04.24
Прорецензовано / Revised: 17.05.24
Схвалено до друку / Accepted: 19.05.24

Vitalii VRUBLEVSKYI, PhD Student
ORCID ID: 0009-0005-7070-9001
e-mail: vitalii.vrublevskyi@gmail.com
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

TRANSFORMER MODEL USING DEPENDENCY TREE FOR PARAPHRASE IDENTIFICATION

Models to represent the semantics of natural language words, sentences, and texts are key in computational linguistics and artificial intelligence. Using quality vector representations of words has revolutionized approaches to natural language processing and analysis since words are the foundation of language. The study of vector representations of sentences is also critical because they aim to capture the semantics and meanings of sentences. Improving these representations helps understand the text at a deeper level and solve various tasks.

The article is devoted to solving the problem of identifying paraphrases using models based on the Transformer architecture. These models have demonstrated high efficiency in various tasks.

It was investigated that their accuracy can be improved by enriching the model with additional information. Using syntactic information such as part-of-speech tags or linguistic structures can improve the model's understanding of context and sentence structure. Enriching the model this way allows you to gain a broader context and improve adaptability and performance in different natural language processing tasks, making it more versatile for different applications. As a result, a model based on the Transformer architecture using a dependency tree was proposed. Its effectiveness compared to other models of the same architecture was investigated using the task of identifying paraphrases. Improvements in accuracy and completeness over the original model (DeBERTa) were demonstrated.

In the future, it is advisable to study the use of this model for other applied tasks (such as plagiarism checking and determining the author's style) and in evaluating other graph structures for sentence representation (for example, AMR graph).

K e y w o r d s : ***natural language processing, paraphrase identification, machine learning, dependency tree, Transformer architecture.***

Автор заявляє про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The author declares no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.