

МЕТОД ФОРМУВАННЯ НАБОРУ ДАНИХ ДЛЯ ПЕРЕВІРКИ ЯКОСТІ ВИВЧЕННЯ МОВНИМИ МОДЕЛЯМИ ТРАНЗИТИВНОГО ВІДНОШЕННЯ У КОНТЕКСТІ ЗАДАЧІ ЛОГІЧНОГО ВИСНОВКУ

Андрій Берко¹, Петро Здебський¹, Вікторія Висоцька^{1,2}

¹ Національний університет “Львівська політехніка”, кафедра інформаційних систем та мереж,
вул. С. Бандери, 12, Львів, Україна

² Університет Оснабрюка, Інститут комп’ютерних наук,
вул. Фрідріха Янсена, 1, Оснабрюк, Німеччина

E-mail: Andrii.Y.Berko@lpnu.ua, ORCID: 0000-0003-2892-9519

E-mail: Petro.V.Zdebskyi@lpnu.ua, ORCID: 0000-0002-0478-2308

E-mail: Victoria.A.Vysotska@lpnu.ua, ORCID: 0000-0001-6417-3689

© Берко А. Ю., Здебський П. В., Висоцька В. А., 2023

Розроблено метод формування набору даних для перевірки вивчення готовими моделями залежності транзитивності. Сформований набір даних використано для тестування якості вивчення моделями залежності транзитивності у задачі логічного висновку (NLI). Тестування набору даних розміром 10 000 зразків (MultiNLI) відбувалось на моделі RoBerta. Також досліджено, що у задачі логічного висновку клас *подібний* є більш спрямованим, ніж *протиріччя* і *нейтральний*. Бо якщо поміняти місцями передумову і гіпотезу у наборі даних, то точність моделі RoBerta зменшується на коефіцієнт 2,97, 1,17, 1,26 для класу *подібний* (0,98 @ 0,33), *нейтральний* (0,90 @ 0,77) та *протиріччя* (0,98 @ 0,78) відповідно. Час ітерації дослідження становить 0,0028 с, тому лише половина набору даних потребує приблизно 84 год обчислень. Це дослідження актуальне, оскільки можливість моделей природної мови досліджувати такі залежності, як транзитивність, яка не задана явно у наборі тренувальних даних, є важливим елементом вміння моделей узагальнювати. Виявлено, що модель RoBerta добре вивчає транзитивні залежності у задачі логічного висновку, бо на всіх зразках зі сформованого набору даних правильно класифікувала належність до класів *подібний*, *протиріччя* та *нейтральний*.

Ключові слова: опрацювання природної мови; речення; транзитивне відношення; машинне навчання; трансформери; MultiNLI; NLI; NLU; NLP; RoBerta; BERT; attention; coder-decoder.

Вступ

Завдання NLI (*natural language inference*, визначення логічних зв’язків між текстами) є хорошим орієнтиром для дослідження NLU (*natural language understanding*, розуміння природної мови), де два речення передають моделі для ідентифікації зв’язку між ними, вибравши один із трьох варіантів: *подібний* (логічний наслідок), *нейтральний* і *протиріччя*. Успіх у NLI не потребує складних навичок машинного навчання, а швидше здатності розуміти значення речень, лексичну та композиційну семантику, а також такі явища, як квантифікація, час, переконання, модальність, лексичну та синтаксичну неоднозначність. Дослідження ґрунтується лише на наборі даних MultiNLI (<https://cims.nyu.edu/~sbowman/multinli/>), який вважається найбільшим набором даних для логічного висновку природної мови та підтримує кілька мов. MultiNLI складається з пар речень, анотованих текстовою інформацією, і охоплює різні жанри тексту [1].

Передумова	Гіпотеза	Клас
The new rights are nice enough	Everyone really likes the newest benefits	neutral
This site includes a list of all award winners and a searchable database of Government Executive articles.	The Government Executive articles housed on the website are not able to be searched.	contradiction
uh i don't know i i have mixed emotions about him uh sometimes i like him but at the same times i love to	i like him for the most part, but would still enjoy seeing someone beat him.	entailment
yeah i i think my favorite restaurant is always been the one closest you know the closest as long as it's it r	My favorite restaurants are always at least a hundred miles away from my house.	contradiction
i don't know um do you do a lot of camping	i know exactly.	contradiction
well that would be a help i wish they would do that here we have got so little landfill space left that we're g	We have plenty of space in the landfill.	contradiction
yeah i know and i did that all through college and it worked too	i did that all through college but it never worked	contradiction
Calcutta seems to be the only other production center having any pretensions to artistic creativity at all, bu	Most of Mrinal Sen's work can be found in European collections.	neutral
If that investor were willing to pay extra for the security of limited downside, she could buy put options wit	The strike price could be \$8.	contradiction
3) Dare you rise to the occasion, like Raskolnikov, and reject the petty rules that govern lesser men?	Would you rise up and defeat all evil lords in the town?	neutral
The most important directions are simply up and up leads eventually to the cathedral and fortress commar	Go downwards to one of the gates, all of which will lead you into the cathedral.	contradiction
The bhakti movement of the Tamils brought a new warmth to the hitherto rigid Brahmanic ritual of Hinduism.	The Tamils' bhakti movement froze the previously warm ritual of Hinduism.	contradiction
see now in a situation like that the boys are only sixteen years old and they were sexually involved with her	Everyone involved was the same age.	contradiction
You and your friends are not welcome here, said Severn.	Severn said the people were not welcome there.	entailment
um-hum um-hum yeah well uh i can see you know it's it's it's kind of funny because we it seems like w	We don't loan a lot of money.	neutral
i'm not sure what the overnight low was	i don't know how cold it got last night.	entailment
so i have to find a way to supplement that	i need a way to add something extra.	entailment
the hologram makes up all these things and uh i mean sometimes sometimes it's funny sometimes it's no	Sometimes it is amusing to see what the hologram creates.	entailment
5 The share of gross national saving used to replace depreciated capital has increased over the past 40 ye	Gross national saving was highest this year.	neutral
So far, however, the number of mail pieces lost to alternative bill-paying methods is too small to have any r	The amount of lost mail is huge and really impacts mail volume	contradiction
you want to punch the button and go	You don't want to push the button lightly, but rather punch it hard.	neutral
Buffet and a la carte available.	It has a buffet.	entailment
Blue says Blumenthal claimed Clinton had told him that Lewinsky had made unwanted sexual advances.	Clinton said that Monica Lewinsky made unwanted sexual advances during her time as a journalist in the White House	neutral
We also have found that leading organizations strive to ensure that their core processes efficiently and effe	Leading organizations want to be sure their employees are safe.	neutral
oh-huh well no they wouldn't would they no	No, they wouldn't go there.	neutral

Рис. 1. Приклад зразків MultiNLI (<https://cims.nyu.edu/~sbowman/multinli/>)

Для дослідження застосовано такі інструменти машинного навчання, як трансформери, кодер-декодери, механізм уваги (attention), моделі BERT і RoBERTa та набір даних MultiNLI.

Аналіз останніх досліджень та публікацій

Transformer – це архітектура нейронної мережі, розроблена для завдань опрацювання природної мови (NLP). Вона стала однією із найширше використовуваних і популярних архітектур у НЛП. Ця архітектура ґрунтується на механізмі самоконтролю, який дає змогу краще зрозуміти контекст і зв'язки між словами, оскільки модель розглядає різні частини вхідної послідовності. На відміну від традиційних моделей НЛП, таких як рекурентні нейронні мережі та згорткові нейронні мережі, *Transformer* є швидшим і ефективнішим, оскільки обробляє усю послідовність паралельно. Він складається із кодера та декодера, кожен з яких складається зі стека ідентичних шарів. Кодер бере вхідну послідовність і виробляє послідовність прихованих станів, тоді як декодер бере вихід кодера та надає кінцеву вихідну послідовність. І кодер, і декодер використовують самоувагу для оброблення вхідних даних і створення вихідних даних. Однією із головних переваг *Transformer* є його здатність обробляти вхідні послідовності змінної довжини без необхідності доповнення або скорочення. Це досягається за рахунок використання позиційного кодування, яке кодує позицію кожного слова у вхідній послідовності як вектор. *Transformer* досяг найсучаснішої продуктивності в широкому діапазоні завдань НЛП, урахувавши машинний переклад, моделювання мови, відповіді на запитання та аналіз настроїв. Багато популярних моделей НЛП, таких як BERT, GPT-3 і RoBERTa, основані на архітектурі *Transformer*.

Позиційне кодування – це техніка, яку використовують у НЛП для кодування відносної позиції tokenів у послідовності. У НЛП послідовності tokenів часто обробляються такими моделями, як *Transformer*, основаними на механізмах уваги. Оскільки у цих моделях немає поняття порядку, їм потрібен додатковий сигнал для подання послідовності. Позиційне кодування досягає цього, додаючи вектор кодування до вбудовування кожного маркера, який фіксує позицію маркера в послідовності. Вектор кодування обчислюють на основі позиції маркера в послідовності та розміру простору вбудовування. Цей вектор кодування додається до вбудовування токена, що дає змогу моделі розрізняти токени на основі їхньої позиції в послідовності. Для таких завдань, як моделювання мови, машинний переклад і класифікація тексту, важливо охопити як семантику tokenів, так і їхню позицію в послідовності, й модель *Transformer* із позиційним кодуванням може досягти цього.

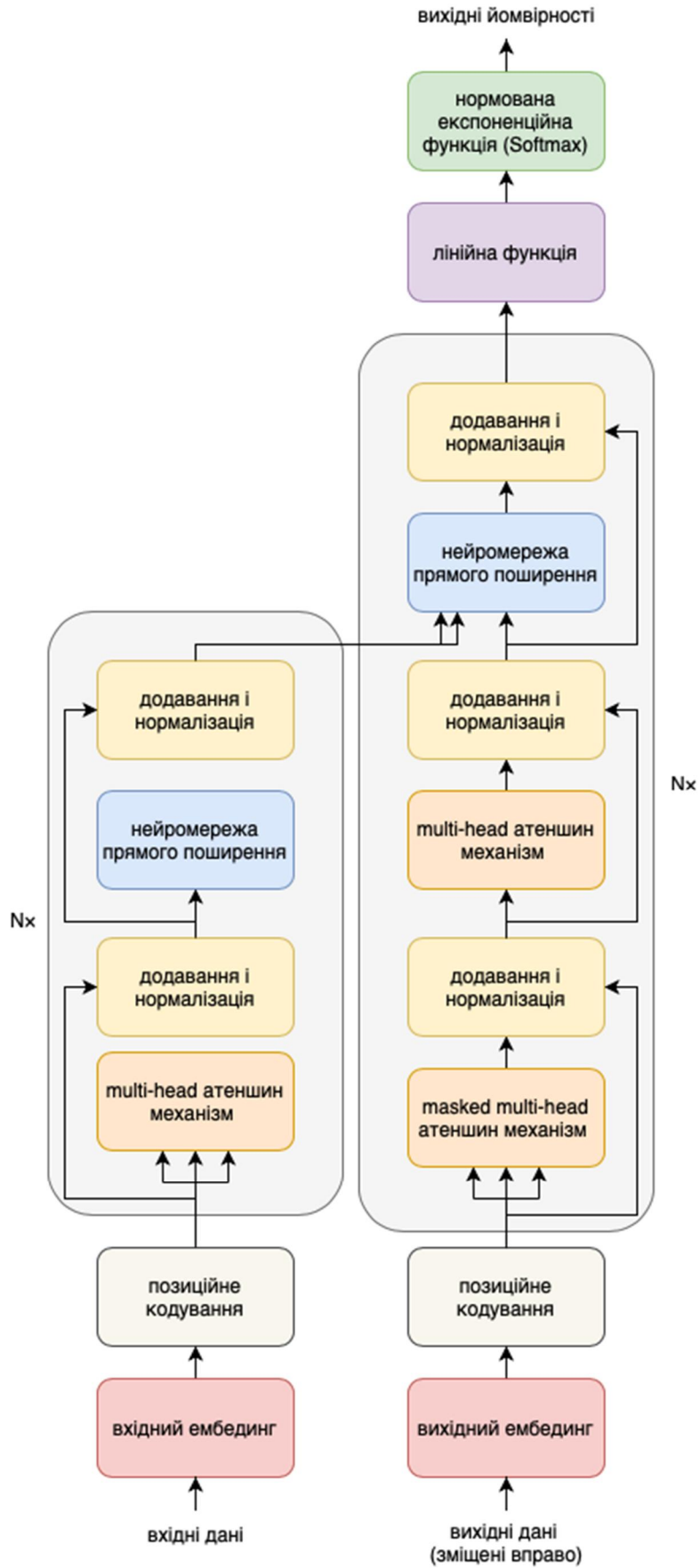


Рис. 2. Архітектура Трансформер (<https://arxiv.org/pdf/1706.03762.pdf>)

Кодер-декодер – це тип архітектури нейронної мережі, який зазвичай використовується для оброблення природної мови та для завдань машинного перекладу. Він складається з двох основних компонентів: кодера та декодера. Як показано на рис. 1, кодер приймає вхідну послідовність і перетворює її на векторне подання фіксованої довжини, яке фіксує істотну інформацію введення. Цей вектор передається на декодер, який генерує вихідну послідовність на основі вхідного вектора. Така архітектура добре підходить для завдань із різною довжиною входу і виходу, оскільки вона може обробляти послідовності змінної довжини для введення та виведення. Прикладами таких завдань можуть бути машинний переклад, конспектування тексту. Кодер містить шість ідентичних шарів, складених разом. На кожному із них два підрівні: багатоголова самоувага та повністю пов'язана мережа прямого зв'язку. Існує залишковий зв'язок навколо кожного з двох підрівнів, услід за ним рівень нормалізації. Кодер зазвичай має стек рекурентних або згорткових нейронних мереж, які обробляють вхідну послідовність по одному маркеру за раз і створюють послідовність прихованих станів. Він бере вхідну послідовність, наприклад речення на одній мові, і перетворює її на векторне подання фіксованої довжини, яке називається вектором контексту. Декодер отримує вектор від кодера та генерує вихідну послідовність. Декодер часто має залишкові зв'язки та нормалізацію, як і кодер.

Масштабований механізм уваги (*attention*) є одним із основних компонентів трансформера. Саме механізм уваги допомагає моделі краще генерувати вихідну послідовність, зосереджуючись на найвідповідніших частинах вхідної послідовності. Він обчислює зважену суму значень на основі набору пар ключ-значення, де ваги обчислюють як скалярний добуток запиту із кожним ключем із застосуванням функції *softmax* для нормалізації ваг. Крім того, щоб градієнти не стали занадто великими, скалярний добуток масштабують квадратним коренем із розмірності ключових векторів. У результаті зважена сума потім використовується як вхідна інформація для наступного рівня в мережі.

Розрахунок уваги для набору запитів зберігається в матриці Q . Ключі зберігаються в матриці K , а значення в матриці V . Формула механізму уваги за скалярним добутком:

$$Attention(Q, K, V) = softmax\left(\frac{QK^t}{\sqrt{d_k}}\right)V.$$

Використовують дві функції уваги. Перша – адитивна, а друга – мультиплікативна увага. За складністю вони подібні, але продукт швидший і ефективніший з погляду простору, оскільки реалізований за допомогою високооптимізованого коду множення матриці.

Якщо значення d_k невеликі, два підходи працюють подібно, але адитивна увага працює краще, ніж увага за скалярним продуктом без масштабування, коли значення d_k більші. Точкові добутки зростають і підштовхують функцію *softmax*, де вона має надзвичайно малі градієнти, коли значення d_k великі. Такий механізм звернення уваги дуже корисний у завданнях NLP, де вхідні послідовності бувають дуже довгими, оскільки він дає моделі змогу зосереджуватися лише на найрелевантніших частинах послідовності та, як наслідок, підвищує точність моделі. Багатоголовий механізм уваги зазвичай використовують у моделях нейронних мереж для завдань NLP, таких як машинний переклад і класифікація тексту. Це дає моделі змогу аналізувати одночасно різні частини послідовності. Такий механізм покращив здатність аналізувати складні зв'язки між частинами вхідних даних.

Краще лінійно проектувати запити, ключі та значення кількістю разів h із різними проєкціями на d_k , d_k , і d_v розміри замість того, щоб використовувати єдину увагу з d_{model} розмірними ключами, значеннями та запитами. Функцію звернення уваги реалізують паралельно на прогнозованих версіях запитів, ключів і значень, отримуючи вихідні значення, які є d_v -вимірними.

Потім для кожної голови обчислюють зважену суму вхідної послідовності, а результати об'єднують разом, щоб сформувати єдиний вихідний вектор. Після цього вихід уваги з кількох головок передається через базовий рівень нейронної мережі прямого зв'язку. Багатоголовий фокус

дає змогу моделям обмінюватися інформацією із різних підпросторів перегляду в різних положеннях. За допомогою усереднення ми контролюємо це, оскільки використовується одна голова механізму уваги.

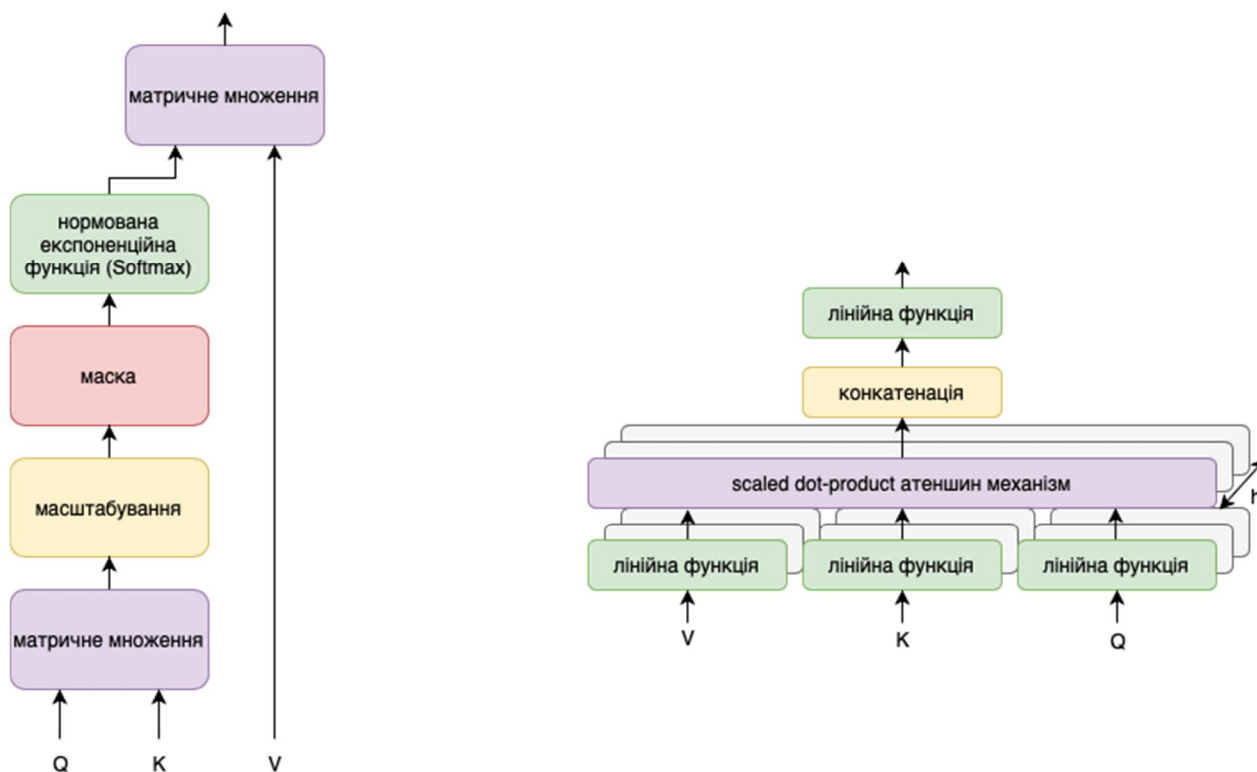


Рис. 3. Механізм уваги (<https://arxiv.org/pdf/1706.03762.pdf>)

Є три різні способи використання багатостороннього механізму уваги в Transformer:

1. У кодері-декодері звертають увагу на ключі та значення пам'яті, що передаються з виходу кодера, але запити від декодера останнього рівня. Зокрема, на кожному кроці часу в декодері механізм уваги обчислює вектор ваги уваги над закодованою вхідною послідовністю. Це дає змогу кожній позиції в декодері відвідувати всі позиції у вхідній послідовності. Це дуже схоже на загальні підходи до уваги кодера-декодера в моделях seq-to-seq.

2. У кодері Self-Attention кодер має таку саму увагу до себе. На рівні самосвідомості все йде з одного місця. У цьому підході вихід беруть з попереднього рівня кодера. Кожна голова механізму уваги звертає увагу на іншу частину вхідної послідовності та обчислює вектор ваги уваги. У кодері кожна позиція може бути в будь-якій або всіх позиціях попереднього рівня декодера.

3. Подібно самоувага декодера дає змогу декодеру перебувати у всіх положеннях декодера до цього положення. На кожному кроці часу декодер звертається до попередньо згенерованих маркерів у цільовій послідовності, щоб обчислити вектор ваги механізму уваги. Це реалізовано всередині масштабованої уваги точки, і всі значення у softmax маскуються введенням, щоб запобігти лівому потоку інформації.

Трансформатор може охопити складні відносини між входом і виходом, використовуючи багатоголовий механізму уваги, тому це одна з найуспішніших моделей у NLP [3].

Загалом у структурі моделі BERT є два етапи. По-перше, це попереднє навчання, а по-друге – точне налаштування. Модель навчається неконтрольованим способом (дані без міток) у різних завданнях під час попереднього навчання. Модель ініціалізується навченими параметрами, а потім усі параметри налаштовуються за допомогою контрольованого підходу (дані з мітками) для точного

налаштування. Для кожного завдання існують різні точно налаштовані моделі, незважаючи на те, що вони ініціалізуються ідентичними попередньо навченими параметрами. Відмінною особливістю такої моделі є те, що для різних завдань використовується та сама архітектура. Це означає, що існує лише невелика відмінність між остаточною архітектурою та попередньо навченою архітектурою. Архітектура моделі BERT – це багатопартийний кодер Transformer, який є двонаправленим. БД для BERT має той самий розмір моделі, що й модель GPT першого покоління OpenAI. Однак в GPT Transformer обмежена самосвідомість, тому він може дивитися лише ліворуч у контексті, тоді як BERT має двонаправлену самосвідомість. BERT навчається виконанню завдання під назвою “Модельовання замаскованої мови” (Masked Language Modeling, MLM): йому дають речення з деякими словами, випадково замаскованими, і просять передбачити замасковані слова на основі контексту – інших слів у реченні. Це допомагає моделі вивчити зв’язок між словами та зрозуміти значення речення. BERT також навчається виконувати завдання під назвою “Передбачення наступного речення” (NSP): йому дають пару речень і просять передбачити, чи є друге речення наступним у документі після першого речення, чи ні. Подання вхідних даних може представляти одне речення та пару речень, щоб модель працювала з різними наступними завданнями. Послідовність у цьому контексті – це послідовність токенів, переданих у модель, яка може складатися з одного або двох речень разом. Вбудовування, використовуване в BERT, називають вбудовуванням WordPiece. Це спеціальний маркер на початку кожної послідовності. Є два способи розрізнення речень: перший полягає у використанні спеціального символу для розділення, а інший – у додаванні до кожного маркера вивченого вбудовування, яке вказує, чи належить воно до того, чи до іншого речення [4, 11].

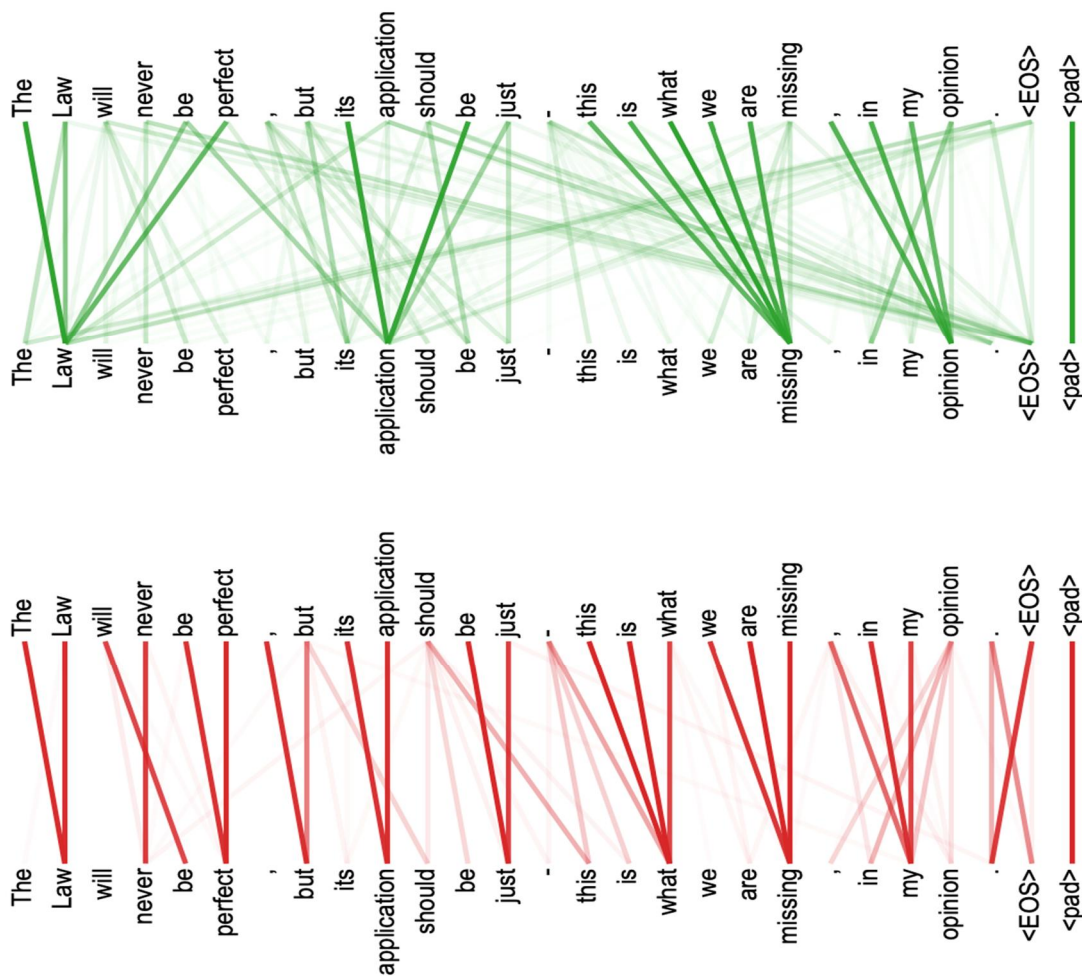


Рис. 4. Візуалізація механізму уваги

RoBERTa (надійно оптимізований підхід попереднього навчання BERT) є варіантом моделі BERT (двонаправлене представлення кодера з трансформаторів). Але він має величезний набір даних для попереднього навчання порівняно з BERT (160 ГБ текстових даних). Цей підхід використовує динамічне маскування, яке випадковим способом маскує різні токени в кожну епоху навчання. RoBERTa навчається виконувати завдання з моделювання мови в масках замість передбачення наступного речення. У першій реалізації довільна заміна та маскування виконуються один раз на початку та зберігаються під час навчання, але на практиці дані дублюються, тому для кожного речення маска не завжди однакова. NSP або Next Sentence Prediction – це завдання двійкової класифікації, щоб передбачити, чи два речення йдуть одне за одним у тексті. Приклади для навчання створені вилученням послідовних речень із навчальних текстових даних, тоді як заперечення сформовано з використанням речень із різних документів. Завдання “Передбачення наступного речення” використовують для покращення виконання завдань, як-от висновок природної мови, які потребують розуміння зв’язку між реченнями [5, 12].

Набір даних MultiNLI (Multi-Genre Natural Language Inference) – це популярний еталонний набір даних для завдань розуміння природної мови в машинному навчанні. Він містить 433 000 прикладів, і це робить його одним із найбільших наборів даних для висновків природною мовою. Складність MultiNLI вища завдяки використанню даних із десяти жанрів письмової та розмовної англійської мови. Це дає змогу оцінити моделі для майже повної складності мови та забезпечує очевидні налаштування для оцінювання адаптації предметної області. Набір даних MultiNLI складається із пар речень із різних жанрів, зокрема художньої літератури, урядових звітів і розмовної мови. Кожна пара речень позначена одним із трьох типів зв’язку: втягнення, протиріччя або нейтральне. Завдання NLP залежать від розуміння природної мови або NLU для досягнення успіху [12–21]. Було докладено багато зусиль для вдосконалення завдань прикладного розуміння природної мови, щоб модель могла успішно справлятися з цими проблемами. Загалом модель повинна бути точною в NLU, а також у додаткових завданнях машинного навчання, таких як доступ до пам’яті чи структуроване передбачення. Унаслідок цього доволі складно правильно оцінити, як моделі NLP розуміють значення мови. Відношення наслідування вказує на те, що із першого речення логічно випливає друге, тобто якщо перше речення істинне, то друге речення також має бути істинним. Відносини протиріччя означають, що перше речення суперечить другому, тобто якщо перше речення істинне, то друге речення повинно бути хибним. Нейтральний зв’язок означає, що між двома реченнями немає логічного зв’язку. Методологія збирання даних у MultiNLI дуже схожа на SNLI. Він містить пару речень із попереднього джерела, а потім анотатора попросили скласти нове речення. Набір даних було створено, щоб стимулювати дослідження розуміння природної мови в кількох жанрах, а не лише в одному жанрі, як це було з багатьма попередніми наборами даних. Це робить моделі складнішими та реалістичнішими для ефективної роботи на цьому наборі даних.

Текст для речень припущень MultiNLI походить із десяти джерел повністю доступного тексту. Тому він має бути різноманітним і представляти американську англійську (рис. 5). Пари речень у наборі даних MultiNLI взято з десяти різних жанрів, серед яких телефонні розмови, путівники та урядові документи. Це гарантує, що набір даних охоплює широкий спектр тем і стилів написання.

Золоту відмітку присвоювали кожній парі речень, які представляють більшість голосів між відміткою, присвоєною початковим анотатором, і чотирма відмітками, присвоєними анотаторами, які перевіряли. Деяка невелика кількість речень не мала консенсусу. Такі речення не слід використовувати в стандартних оцінках і вводити в розподіл, але вони мають тире як золоту позначку. Як показано на рис. 5, набір даних містить пару речень, жанр, золоту мітку та всі мітки, призначені анотаторами [1].

Клас анотатор 1	Клас анотатор 2	Клас анотатор 3	Клас анотатор 4	Клас анотатор 5	Жанр	Фінальний клас	Речення 1 (передумова)	Речення 1 оброблене	Речення 2 (гіпотеза)	Речення 2 оброблене
neutral	entailment	neutral	neutral	neutral	slate	neutral	The new rights are nice eno	(((The (new rights)) are (r	Everyone really likes the	(Everyone (really (likes (th
contradiction	contradiction	contradiction	contradiction	contradiction	government	contradiction	This site includes a list of all	(((This site)) (includes ((((The Government Exec	(((The (Government (Exec
entailment	entailment	entailment	entailment	entailment	telephone	entailment	uh i don't know i i have mixe	(((((uh (i ((do n't)) (know	{ I like him for the most p	{ I (((((like him)) (for (the
contradiction	contradiction	contradiction	contradiction	contradiction	telephone	contradiction	yeah i i think my favorite res	{ yeah ((i i)) (think ((my (fa	My favorite restaurants	{ (My (favorite restaurants))
contradiction	contradiction	contradiction	contradiction	contradiction	telephone	contradiction	i don't know um do you do	{ i ((do n't)) (know (um (dc	I know exactly.	{ I ((know exactly . .))
contradiction	contradiction	contradiction	contradiction	contradiction	telephone	contradiction	well that would be a help i w	{ well (that (would (be (a	We have plenty of spac	{ We ((have (plenty (of (sp
contradiction	contradiction	contradiction	contradiction	contradiction	telephone	contradiction	yeah i know and i did that al	{ yeah ((i know)) and) (i (i	did that all through col	{ I ((did (that all)) (throug
neutral	neutral	entailment	neutral	neutral	travel	neutral	Calcutta seems to be the or	(((((Calcutta (seems (to (Most of Mrinal Sen's wr	{ (Most (of ((Mrinal (Sen 's
contradiction	contradiction	contradiction	contradiction	contradiction	slate	contradiction	If that investor were willing t	(((((that investor)) were (The strike price could t	{ (The (strike price)) (cou
neutral	neutral	neutral	neutral	neutral	slate	neutral	3) Dare you rise to the occa	(((3 -RRB-)) (Dare (you (Would you rise up and	{ ((Would you)) ((rise up)
contradiction	contradiction	entailment	contradiction	contradiction	travel	contradiction	The most important directio	(((The ((most important)) di	Go downwards to one c	{ (Go ((downwards (to (one
contradiction	contradiction	contradiction	contradiction	entailment	travel	contradiction	The bhakti movement of the	(((((The (bhakti movement))	The Tamils' bhakti mov	{ (((The (Tamils ')) (bhakti
contradiction	contradiction	contradiction	contradiction	contradiction	telephone	contradiction	see now in a situation like th	(((((see now)) (in ((((a	Everyone involved was	{ (Everyone involved)) (wa
entailment	entailment	entailment	entailment	entailment	fiction	entailment	You and your friends are not	(((((You and)) (your friends	Severn said the people	{ (Severn (said ((the peopl
contradiction	neutral	contradiction	neutral	neutral	telephone	neutral	um-hum um-hum yeah well	(((((um-hum (um-hum yeah	We don't loan a lot of r	{ We ((do n't)) (loan ((a lc
entailment	entailment	entailment	entailment	entailment	telephone	entailment	i'm not sure what the overni	{ i (('m not)) (sure (what (I don't know how cold i	{ I ((do n't)) (know (how
entailment	entailment	entailment	entailment	entailment	telephone	entailment	so i have to find a way to su	{ so (i (have (to (find (a w	I need a way to add sor	{ I ((need ((a way)) (to (ac
entailment	entailment	entailment	entailment	entailment	telephone	entailment	the hologram makes up all t	(((((((the hologram)) (r	Sometimes it is amusin	{ (Sometimes (it ((is (amusi
neutral	neutral	neutral	neutral	neutral	government	neutral	5 The share of gross nation	(((5 (The share))) (of ((gn	Gross national saving v	{ (Gross (national saving))
contradiction	contradiction	contradiction	contradiction	contradiction	government	contradiction	So far, however, the number	(((So far)) (, (however , ((The amount of lost mail	{ (The amount of lost mail)) (of (lost n
neutral	neutral	contradiction	entailment	neutral	telephone	neutral	you want to punch the butt	{ you (want (to ((punch (You don't want to push	{ You ((do n't)) (want (to (
entailment	entailment	entailment	entailment	entailment	travel	entailment	Buffet and a la carte availa	(((((Buffet and)) (a la))	It has a buffet.	{ It ((has (a buffet .) .))
neutral	neutral	entailment	neutral	neutral	slate	neutral	Blue says Blumenthal claim	{ Blue ((says (Blumenthal (Clinton said that Monic	{ Clinton (said (that ((Mor
contradiction	neutral	neutral	neutral	neutral	government	neutral	We also have found that lea	{ We (also ((have (found (t	Leading organizations v	{ (Leading organizations)) (
neutral	neutral	neutral	neutral	neutral	telephone	neutral	oh uh-huh well no they wou	{ oh ((uh-huh (well (no ((t	No, they wouldn't go th	{ No (, (they ((would n't)

Рис. 5. Приклади з MultiNLI (<https://www.cs.princeton.edu/courses/archive/spring20/cos598C/lectures/lec18-annotation-artifacts.pdf>)

Під час аналізу літератури виявлено й інші спроби сформулювати задачу, схожу на задачу транзитивності. У статті “Does BERT Know that the IS-A Relation Is Transitive?” перевірено вміння моделі BERT вивчати залежності транзитивності у контексті відношення “є” (IS-A). Тобто якийсь один предмет чи процес є підмножиною іншого предмета або процесу. Наприклад, слон є твариною, отже, множина слонів є підмножиною тварин. Для цього використано семантичний словник WordNet, у якому є онтологічні зв’язки між англійськими словами. Один зі зв’язків між словами – зв’язок “є” (IS-A), який використано у проаналізованій роботі. Там застосували останній шар моделі Bert, щоб отримати вкладення слів (word embedding). Після цього знайшли найближче слово із тренувального набору даних, користуючись евклідовою відстанню між векторами. Очікувалось, що найближче слово відображатиме відношення “є”. Було виявлено, що модель Bert у 82 % випадків правильно класифікує, що “A є B”, якщо “A є B” і “B є B”. [<https://aclanthology.org/2022.acl-short.11.pdf>]

Описаний вище підхід відрізняється від запропонованого у цій роботі, бо він використовує WordNet для формування нового набору даних, а у цій роботі використано набір даних MultiNLI як набір даних, на основі якого формується новий. Цей набір даних спеціалізований для задачі логічного висновку. Також у цій роботі протестовано саме вміння моделі вивчати транзитивні залежності у задачі логічного висновку, а не у контексті онтологічної належності слів до ширших понять, як у WordNet. Крім цього, протестована у цій роботі модель спеціалізована на задачі логічного висновку, бо раніше уже натренована компанією Facebook на наборі даних MultiNLI.

Формулювання мети та постановка задачі

Мета роботи – визначити здатність мовних моделей вивчати транзитивні залежності у задачі логічного висновку (NLI), а завдання – сформулювати набір даних на основі наявного набору даних та запропонованих правил розмічування набору даних, для тестування якості моделі RoBERTa. Для задачі логічного висновку сформовано набори даних, один із яких – MultiNLI, описаний раніше. Також є готова модель з архітектурою RoBERTa, яка натренована на цьому наборі даних. У результаті цього вона навчилася класифікувати речення на три класи: подібний, протиріччя і нейтральний. Детальніше: мета цієї роботи – дослідити, чи вона також вивчила фундаментальну залежність транзитивності, яка повинна виконуватись у випадку логічної імплікації. Тобто, якщо модель знає, що із

речення 1 впливає речення 2, а із речення 2 впливає речення 3, то вона повинна генералізувати і зрозуміти, що із речення 1 впливає речення 3. Якщо це записати у вигляді формального закону транзитивності для імплікації: $A \rightarrow B$ і $B \rightarrow C$, то $A \rightarrow C$. Прикладів із такими залежностями не повинно бути у початковому наборі даних, на якому тренується модель, щоб модель могла вивчити такі залежності. У цій роботі на основі набору даних MultiNLI сформовано новий набір даних для тестування якості вивчення транзитивних залежностей. Визначено правила, за якими генерувався новий набір даних на основі набору даних MultiNLI.

Також у цій роботі перевірено спрямованість кожного із класів у наборі даних MultiNLI. У задачі NLI логічний висновок є напрямленим відношенням [2]. Наприклад, якщо із речення 1 впливає речення 2, то це не означає, що із речення 2 впливає речення 1. Менша спрямованість у класах *протиріччя* і *нейтральний*. Можемо перевірити зв'язок у зворотному напрямку. Замість того, щоб робити висновок з першого речення на друге, поміняємо їх місцями і зробимо висновок з другого речення на перше. Якщо точність готової моделі, такої як RoBERTa, зменшуватиметься для якогось класу, можемо зробити висновок, що відношення для деяких класів є напрямленим, ніж інші.

Модель, яка буде використовуватися, має назву RoBERTa, вона досягає кращих результатів на різних тестах без тонкого налаштування для GLUE. Це величезна модель глибокого навчання, навчена Facebook і дотренована на згаданому раніше наборі даних (MultiNLI). Це вдосконалена модель BERT, яку покращено завдяки навчанню моделі на більшій кількості даних протягом тривалішого періоду. У цій роботі модель не тренувалась, оскільки ми поставили мету дослідити, наскільки якісно вона вивчає транзитивні відношення, сформувавши новий набір даних для її тестування. Основним завданням було саме визначення правил, на основі яких формуватиметься новий набір даних.

Виклад основного матеріалу

На основі знань предметної області створено такі правила для перевірки транзитивності:

(Entailment, Label)Label,
(Contradiction, Contradiction)Entailment,
(Contradiction, Entailment)Contradiction,
(Contradiction, Label)Neutral,
(Neutral, Label)Neutral,

де *Entailment* – клас *подібний* (логічний наслідок); *Label* – мітка; *Contradiction* – клас *протиріччя*; *Neutral* – клас *нейтральний*. Відношення *Entailment* є орієнтованим відношенням.

Ці правила будемо використовувати, щоб сформувані новий набір даних. Візьмемо для прикладу правило *(Contradiction, Entailment)Contradiction*. Це означає: якщо у першій парі речень є клас *протиріччя* (contradiction), а у другій парі речень клас *подібний* (entailment), то можна сформувані приклад для нового набору даних, взявши перше речення із першої пари і друге із другої, і поставити між ними клас залежності *протиріччя*. Тобто, якщо “А contradiction Б” і “Б entailment В”, то “А contradiction В”.

У текстовому висновку (наслідування) завдання є передумовою, а відношення гіпотези вважається спрямованим. Наприклад, візьмемо “Минулого року я був у Оснабрюку” як передумову та речення “Я був за кордоном рік тому”. У цьому прикладі чітко бачимо, що відношення є спрямованим, можемо вивести гіпотезу з передумови, але не можемо зробити навпаки, тому що якщо хтось був за кордоном, це не означає, що він був у Оснабрюку. Лише деякі автори використовували спрямованість співвідношення логічний наслідок: це означає, що якщо $T \rightarrow H$, мало ймовірно, що зворотний $H \rightarrow T$ також виконується. З логічного погляду відношення наслідування подібне до імплікації, яка, всупереч еквівалентності, не є симетричною [2, 8, 9]. З усіх цих правил лише перше можна враховувати строго. Якщо є логічний наслідок у першій парі та логічний наслідок у другій, тоді має бути відношення логічного наслідку між передумовою першої пари та гіпотезою другої пари.

Наприклад, перша пара: “Я був у Оснабрюку” і “Я був у Німеччині”, друга пара: “Я був у Німеччині” та “Я відвідав Німеччину”, потім “Я був у Оснабрюку” і “Я відвідав Німеччину” – наслідок. Аналогічно ми можемо пояснити для інших міток.

У табл. 1, 2 неправильні класифікації виділені сірим фоном. Тут можна використовувати метрику точності, оскільки класи збалансовані. Збалансованості класів досягнуто тим, що цей набір даних ми згенерували основі правил і для кожного із класів взято однакову кількість прикладів із оригінального набору даних (MultiNLI).

Сформовано набір даних розміром 10 тисяч прикладів на основі набору даних MultiNLI. У табл. 1 наведено приклад пари речень (передумови і гіпотези) із MultiNLI, а у табл. 2 сформований набір даних на основі правил, описаних раніше. Тобто якщо у MultiNLI друге речення у якомусь прикладі тотожне із першим у іншому, можна ці приклади з’єднати і встановити залежність із першого речення першого прикладу до другого речення другого прикладу. Мітка класу, яка міститиметься у результуючому наборі даних, визначається за правилами, описаними вище.

Після тестування моделі RoBERTa на результуючому наборі даних було виявлено, що вона правильно класифікує усі приклади. Тобто можемо зробити висновок, що транзитивні залежності модель вивчила якісно.

Таблиця 1

Класифікація оригінальних даних

Перша пара				Друга пара			
Передумова	Гіпотеза	Передбачувана мітка	Справжня мітка	Передумова	Гіпотеза	Передбачувана мітка	Справжня мітка
Боюся, що ні, панове	Я так не думаю	логічний наслідок	логічний наслідок	Я так не думаю	Ви шукаєте не в тому місці	нейтральний	нейтральний
Боюся, що ні, панове	Я так не думаю	логічний наслідок	логічний наслідок	Я так не думаю	Я поняття не маю	нейтральний	проти-річчя
Ти знаєш і він ти знаєш я не знаю	Я не знаю	логічний наслідок	логічний наслідок	Я не знаю	Я знаю	проти-річчя	проти-річчя
Зовсім ні – або, принаймні, я так не думаю	Я так не думаю	логічний наслідок	логічний наслідок	Я так не думаю	Ви шукаєте не в тому місці	нейтральний	нейтральний
Зовсім ні – або, принаймні, я так не думаю	Я так не думаю	логічний наслідок	логічний наслідок	Я так не думаю	Я поняття не маю	нейтральний	проти-річчя
Ага, це правда, це правда, так	Це вірно	логічний наслідок	логічний наслідок	Це вірно	Це правильно	логічний наслідок	логічний наслідок
Цілком	Абсолютно	логічний наслідок	логічний наслідок	Абсолютно	Немає сумніву, що я це зроблю	нейтральний	нейтральний
Цілком	Абсолютно	логічний наслідок	логічний наслідок	Абсолютно	Безумовно	логічний наслідок	логічний наслідок

Таблиця 2

Класифікація створених прикладів даних із транзитивними відношеннями

Передумова	Гіпотеза	Передбачувана мітка	Справжня мітка
Боюся, що ні, панове	Ви шукаєте не в тому місці	нейтральний	нейтральний
Боюся, що ні, панове	Я поняття не маю	протиріччя	нейтральний
Ти знаєш і він, ти знаєш, я не знаю	Я знаю	протиріччя	протиріччя
Зовсім ні – або, принаймні, я так не думаю	Ви шукаєте не в тому місці	нейтральний	нейтральний
Зовсім ні – або, принаймні, я так не думаю	Я поняття не маю	протиріччя	нейтральний
Ага, це правда, це правда, так	Це правильно	логічний наслідок	логічний наслідок
Цілком	Немає сумніву, що я це зроблю	нейтральний	нейтральний
Цілком	Безумовно	логічний наслідок	логічний наслідок

Також перевіримо, що отримаємо, якщо поміняти місцями перше і друге речення (табл. 3, рис. 6). Друге речення розглядатиметься як передумова, а перше – як гіпотеза [10]. Набір даних є величезною випадковою підмножиною, що містить 10 000 зразків, які взято з набору даних (час розрахунку становив більше ніж 2 год).

Таблиця 3

Метрика класифікації від гіпотези до передумови

Точність	Передумова @ гіпотеза	Гіпотеза @ передумова (у зворотній послідовності)	Зменшення на
Загальна	0,95	0,63	1,51
Для класу <i>подібний</i>	0,98	0,33	2,97
За класом <i>нейтральний</i>	0,90	0,77	1,17
Для класу <i>протиріччя</i>	0,98	0,78	1,26

Як бачимо (табл. 3), точність зменшилася на коефіцієнт 2,97, 1,17, 1,26 для класу *подібний*, *нейтральне* та *протиріччя* відповідно. Звідси можна зробити висновок, що *подібний* клас є найспрямованішим відношенням, потім *протиріччя*, а потім *нейтральний*. Ці числа насправді інтуїтивно зрозумілі, тому що якщо одне з речень суперечить іншому, це зазвичай означає, що воно може бути в зворотній послідовності від другого речення до першого. Для класу *нейтральний* точність знижується ще менше порівняно з *протиріччям*, оскільки нейтральні речення зазвичай ще

частіше є нейтральними в іншому напрямку. Що стосується класу *подібний*, напрям класу дуже сильний, і можемо підтвердити це цифрами, тому що точність стає нижчою приблизно утричі, якщо поміняти речення місцями.

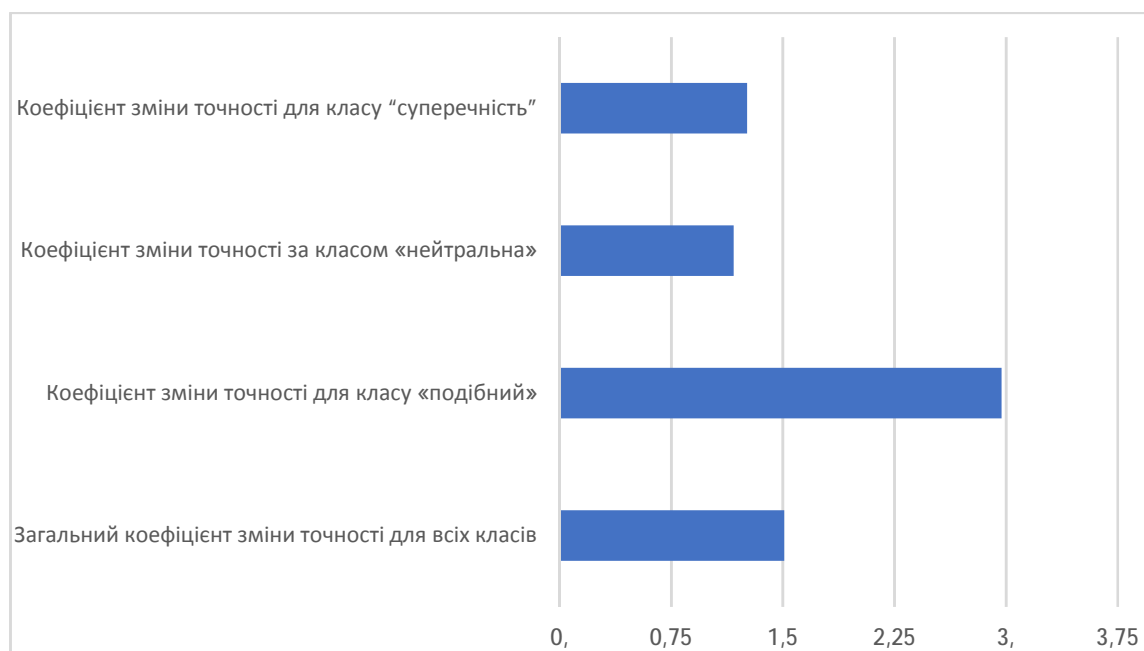
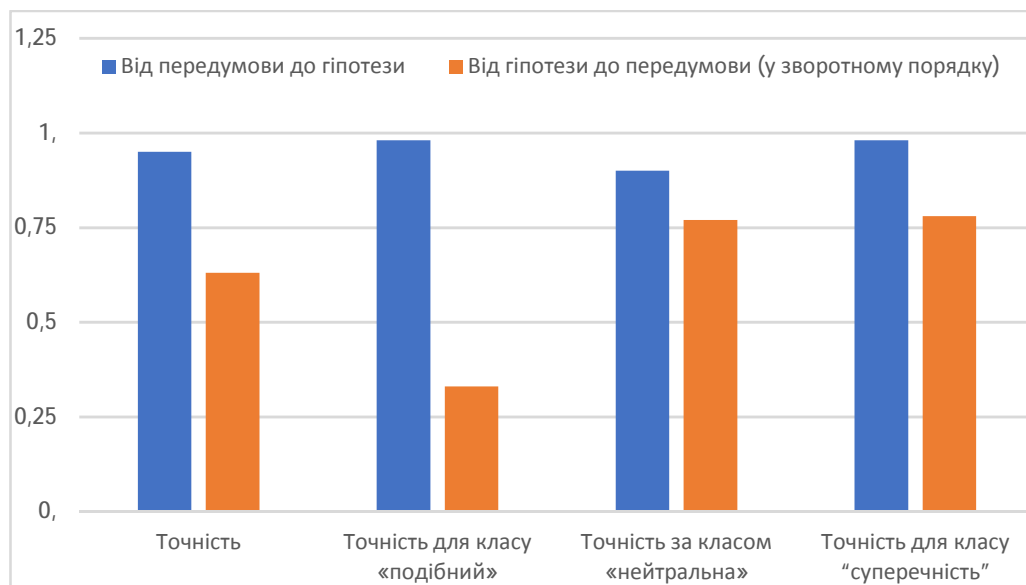


Рис. 6. Результати перевірки моделі

Паралельно виконано вимірювання часу виконання програми (рис. 7). Це актуально у зв'язку з активним переходом з Web 2.0 на Web 3.0 з урахуванням як семантики контенту, так і методів штучного інтелекту опрацювання цього контенту для повернення користувачу права власності на свої дані та контент. Паралельно будь-які дії в інтернеті повинні забезпечувати не лише точність шуканого результату, але й оперативність його отримання. Боротьба між такими антагонізмами, як точність та швидкість в інтернет-просторі, є вічним питанням. Крім того, більшість користувачів мають середньостатистичні комп'ютери для роботи в інтернеті, тому проаналізовано швидкодію запропонованого методу на персональному комп'ютері.

Ідея полягає в тому, щоб перевірити, чи є гіпотези (другі речення), які є передумовами (перші речення) в інших зразках. Це дасть змогу створити набір даних, щоб перевірити, наскільки добре моделі вловлюють транзитивне відношення. Використовується сильна відповідність, але також можна застосовувати нечітку відповідність. Вибірку набору даних перевірено, оскільки набір даних величезний, і обчислення для всього набору даних займе багато часу.

$$TimeElapsed = (IterationsNumber \cdot IterationTime)^2$$

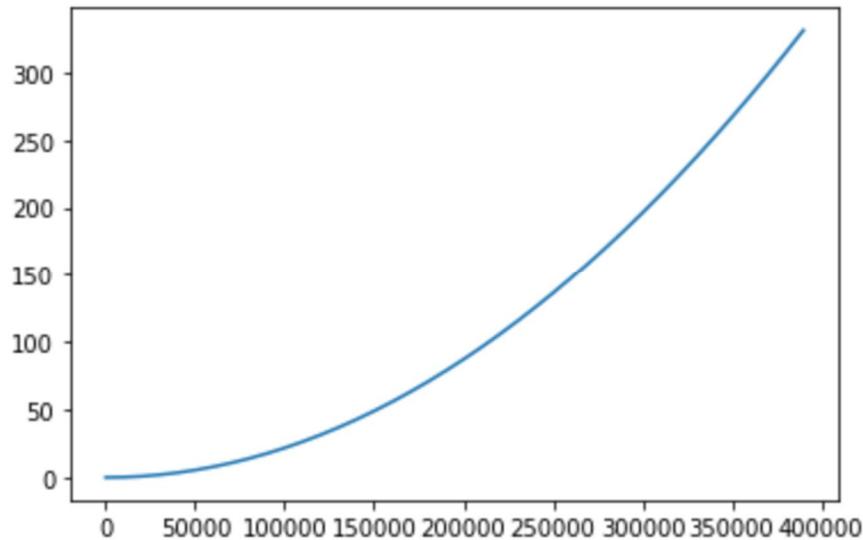


Рис. 7. Графік часової складності виконання коду

Час ітерації на комп'ютері становив 0,0028, тому ми можемо побудувати графік часу обчислення. Отже, навіть для половини набору даних потрібно 84 години обчислень (рис. 7).

Висновки

Робота націлена на орієнтований на дані підхід до підвищення точності моделі за рахунок поліпшення якості даних замість покращення архітектури моделі. Ідея полягає в тому, щоб покращити набір даних за допомогою відносин транзитивності, щоб допомогти моделі машинного навчання вивчати такі залежності, але в результаті виявилось, що модель добре вивчає такі залежності без додаткового дотренування. Разом зі збагаченням набору даних досліджено, наскільки добре попередньо навчена модель вивчає такі зв'язки. Отже, дослідження можна розділити на дві основні частини: формування набору даних (10 000 зразків) і дослідження моделі машинного навчання Roberta, попередньо навченої на таких наборах даних. Виявлено, що наявна модель Roberta добре вловлює транзитивні залежності, оскільки правильно класифікувала всі приклади із нового, сформованого набору даних. Також досліджено, що відношення “подібний” є спрямованішим, ніж “суперечність” і “нейтральне”.

Точність зменшилася на коефіцієнт 2,97, 1,17, 1,26 для класів *подібний*, *нейтральний* та *протиріччя* відповідно. Звідси можна зробити висновок, що *подібний* клас є найспрямованішим відношенням, відтак *суперечність*, а потім *нейтральність*. Ці числа насправді інтуїтивно зрозумілі, оскільки якщо одне з речень суперечить іншому, це зазвичай означає, що воно може бути в зворотній послідовності від другого речення до першого. Для класу *нейтральний* точність знижується ще менше порівняно з *протиріччям*, оскільки нейтральні речення зазвичай ще частіше є нейтральними в іншому напрямку. Що стосується класу *подібний*, напрям дуже сильний, і можемо підтвердити це цифрами, тому що точність стала нижчою приблизно утричі.

Розроблений прототип демонструє виконану роботу та основний функціонал вебдодатка. Час ітерації на комп'ютері становив 0,0028, тому можна побудувати графік часу обчислення. Отже, навіть для половини набору даних нам потрібно 84 год обчислень.

References

1. Adina Williams, Nikita Nangia, Samuel R. Bowman, A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference, 2018. <https://doi.org/10.48550/arXiv.1704.05426>
2. Doina Tatar, Gabriela Serban, Mihis Andreea, Textual Entailment as a Directional Relation, 2008. URL: <https://search.informit.org/doi/abs/10.3316/INFORMIT.836390534395451>
3. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, Attention Is All You Need, 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
4. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019. <https://doi.org/10.48550/arXiv.1810.04805>
5. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019. <https://doi.org/10.48550/arXiv.1907.11692>
6. William McDaniel Albritton, ICS 241: Discrete Mathematics II, Waterloo, 2015
7. Petro Zdebskyi, Vasyl Lytvyn, Yevhen Burov, Zoriana Rybchak, Petro Kravets, Olga Lozynska, Roman Holoshchuk, Solomiya Kubinska, Alina Dmytriv, Intelligent System for Semantically Similar Sentences Identification and Generation Based on Machine Learning Methods, *CEUR workshop proceedings*, Vol. 2604, 317–346, 2020. <https://ceur-ws.org/Vol-2604/paper25.pdf>
8. Aikaterini-Lida Kalouli, Annebeth Buis, Livy Real, Explaining Simple Natural Language Inference, 2019. DOI: 10.18653/v1/W19-4016
9. Tongfei Chen, Zhengping Jiang, Adam Poliak, Keisuke Sakaguchi, Benjamin Van Durme, Uncertain Natural Language Inference, 2020. <https://doi.org/10.48550/arXiv.1909.0304>
10. Adam Poliak, A Survey on Recognizing Textual Entailment as an NLP Evaluation, 2020. <https://doi.org/10.48550/arXiv.2010.03061>
11. Hanwen Zha, Zhiyu Chen, Xifeng Yan, Large Batch Optimization for Deep Learning: Training BERT in 76 minutes, 2021. <https://doi.org/10.48550/arXiv.1904.00962>
12. Leo Z. Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, Noah A. Smith, Probing Across Time: What Does RoBERTa Know and When?, 2021. <https://doi.org/10.48550/arXiv.2104.07885>
13. Zdebskyi P., Berko A., Vysotska V. (2023). Investigation of Transitivity Relation in Natural Language Inference, *CEUR Workshop Proceedings*, Vol. 3396, 334–345.
14. Bisikalo O., Vysotska V., Burov Y., Kravets P. (2020). Conceptual Model of Process Formation for the Semantics of Sentence in Natural Language, *CEUR workshop proceedings*, Vol. 2604, 151–177.
15. Vysotska V., Holoshchuk S., Holoshchuk R. (2021). A comparative analysis for English and Ukrainian texts processing based on semantics and syntax approach. *CEUR Workshop Proceedings*, Vol. 2870, 311–356.
16. Rogushina J. (2023). Ontological Approach in the Smart Data Paradigm as a Basis for Open Data Semantic Markup. *CEUR Workshop Proceedings*, Vol. 3403, 12–27.
17. Hryhorovych V. (2023). Calculation of the Semantic Distance between Ontology Concepts: Taking into Account Critical Nodes, *CEUR Workshop Proceedings*, Vol. 3396, 206–216.
18. Albota S. (2023). Creating a Model of War and Pandemic Apprehension: Textual Semantic Analysis. *CEUR Workshop Proceedings*, Vol. 3396, 228–243.
19. Taran A. (2023). Corpus Analysis of Word Semantics. *CEUR Workshop Proceedings*, Vol. 3396, 373–384.
20. Basyuk T., Vasilyuk A., Lytvyn V. (2019). Mathematical model of semantic search and search optimization. *CEUR Workshop Proceedings*, Vol. 2362, 96–105.
21. Sharonova Natalia, Kyrychenko Iryna, Gruzdo Iryna, Tereshchenko Glib (2022). Generalized Semantic Analysis Algorithm of Natural Language Texts for Various Functional Style Types. *CEUR Workshop Proceedings*, Vol. 3171, 16–26.

**DATA SET FORMATION METHOD FOR CHECKING THE QUALITY
OF LEARNING LANGUAGE MODELS OF THE TRANSITIVE RELATION
IN THE LOGICAL CONCLUSION PROBLEM CONTEXT**

Andriy Berko¹, Petro Zdebskyi¹, Victoria Vysotska^{1,2}

¹ Lviv Polytechnic National University, Information Systems and Networks Department,
12, S. Bandery str., Lviv, Ukraine

² Osnabrück University, Institute of Computer Science, 1, Friedrich-Janssen str., Osnabrück, Germany
E-mail: Andrii.Y.Berko@lpnu.ua, ORCID: 0000-0003-2892-9519
E-mail: Petro.V.Zdebskyi@lpnu.ua, ORCID: 0000-0002-0478-2308
E-mail: Victoria.A.Vysotska@lpnu.ua, ORCID: 0000-0001-6417-3689

© *Berko A., Zdebskyi P., Vysotska V., 2023*

Summary. A method for data set formation has been developed to verify the ability of pre-trained models to learn transitivity dependencies. The generated data set was used to test the quality of learning the transitivity dependencies in the task of natural language inference (NLI). Testing of a data set with a size of 10,000 samples (MultiNLI) used to test the RoBERTa model. It was found that this model is good at studying transitive dependencies in the task of logical inference because all samples from the formed dataset were correctly classified as belonging to the class similar, contradiction and neutral. It was also investigated that in the task of logical inference, the class similarity is more directed than contradiction and neutral. Because if the premise and hypothesis in the data set are swapped, the accuracy of the RoBERTa model decreases by a factor of 2.97, 1.17, 1.26 for the similar (0.98 @ 0.33), neutral (0.90 @ 0.77), and contradiction (0.98 @ 0.78) classes, respectively. The study iteration time is 0.0028 seconds, so only half of the data set requires approximately 84 hours of collection. This research is relevant because the ability of natural language models to explore such dependencies as transitivity, which is not explicitly specified in the training data set, is an important element of the model's ability to generalize. It was found that RoBERTa's model is good at studying transitive dependencies in the logical inference task because it correctly classified belonging to the class similar, contradiction, and neutral on all samples from the generated data set.

Key words: natural language processing; sentences; transitive relation; machine learning; transformers; MultiNLI; NLI; NLU; NLP; RoBERTa; BERT; attention; coder-decoder.