Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка
Навчально-науковий інститут філології
кафедра української мови та прикладної лінгвістики

РОЗРОБКА МОДЕЛІ ДЛЯ ВИКОНАННЯ МОРФОЛОГІЧНОГО ТЕГУВАННЯ УКРАЇНСЬКОМОВНИХ ТЕКСТІВ НА ОСНОВІ ПРЕТРЕНОВАНОГО ТРАНСФОРМЕРА GPT-3.5

Кваліфікаційна робота
студента 4 курсу
освітньої програми
«Прикладна(комп'терна) лінгвістика та англійська мова»
спеціальності – 035.10 Філологія (прикладна лінгвістика)
галузі знань – 03 гуманітарні науки
Кирила Сергійовича Кучинського
Науковий керівник:
к.т.н. доц. Микола КОСТІКОВ
Науковий консультант:
ас. каф. Валентина РОБЕЙКО

«Допущено до захисту»
Протокол засідання
кафедри української мови та прикладної лінгвістики
протокол № 15 **від «** 6 **»** 06 **2024 року**
завідувач кафедри _____(підпис)
к.філол.н., доц. Сергій РІЗНИК

КИЇВ — 2024

ЗМІСТ

АНОТАЦІЯ

Ця робота присвячена розробці моделі для автоматичного частиномовного тегування українських текстів на базі трансформера GPT-3.5. Ця робота є актуальною через зростаючу потребу в автоматизації обробки українськомовних текстів, включно з аналіом сучасного розмовного мовлення.

Об'єктом дослідження є методи та інструменти POS-тегування для автоматизації обробки текстових даних.

Предметом є порівняння ефективності різних інструментів для POS-тегування з натренованою моделлю GPT-3.5.

Метою дослідження є розробка моделі для автоматичного тегування частин мови в українському тексті на основі GPT-3.5 та оцінка її ефективності.

Завдання включають аналіз наявних інструментів POS-тегування, тренування моделі GPT-3.5 на українських текстах та порівняння результатів.

Методологічні підходи базуються на сучасних досягненнях глибокого навчання та трансформерних моделей.

Новизна дослідження полягає у впровадженні GPT-3.5 для обробки українського тексту.

У Розділі 1 описано сучасний стан галузі автоматичної морфологічної розмітки.

Розділ 2 присвячений розробці моделі для виконання автоматичного тегування частин мови на основі претренованого трансформера GPT-3.5. Описано процес формування тренувального набору даних, розробки порівняльного застосунку, процедуру файн-тюнінгу моделі та результати тренування.

У Розділі 3 розглядаються методи тестування розробленої моделі. Зокрема, описано процес збору текстів для тестування та проведення порівняльного аналізу з іншими інструментами. Особлива увага приділяється проблемам обробки текстів з помилками, нестандартним регістром, чергуванням мов та граматичною омонімією.

Результати роботи підтверджують, що впровадження GPT-3.5 для обробки українського тексту значно покращило якість автоматичного тегування, зокрема для текстів розмовного мовлення.

Ключові слова: POS-тегування, GPT-3.5, Українська мова, Код-світчинг, Лінгвістичний аналіз, Обробка природної мови (NLP), Файн-тюнінг, Морфологічна розмітка

ABSTRACT

This paper is devoted to the development of a model for automatic part-of-speech tagging of Ukrainian texts based on the GPT-3.5 transformer. This work is relevant due to the growing need to automate the processing of Ukrainian-language texts, including the analysis of modern colloquial language.

The object of research is POS tagging methods and tools for automating text data processing.

The subject is to compare the efficiency of various POS-tagging tools with the trained GPT-3.5 model.

The purpose of the study is to develop a model for automatic tagging of parts of speech in Ukrainian text based on GPT-3.5 and to evaluate its effectiveness.

The tasks include analyzing existing POS tagging tools, training the GPT-3.5 model on Ukrainian texts, and comparing the results.

The methodological approaches are based on modern advances in deep learning and transformational models.

The novelty of the study lies in the implementation of GPT-3.5 for Ukrainian text processing.

Section 1 describes the current state of the art in automatic morphological tagging.

Section 2 describes the process of forming a training dataset, developing a comparative application, the procedure of model fine-tuning, and the training results.

Section 3 discusses the methods of testing the developed model. In particular, it describes the process of collecting texts for testing and conducting a comparative analysis with other tools. Particular attention is paid to the problems of processing texts with errors, non-standard case, language alternation, and grammatical homonymy.

The results confirm that the implementation of GPT-3.5 for Ukrainian text processing has significantly improved the quality of automatic tagging, in particular for spoken texts.

Keywords: POS-tagging, GPT-3.5, Ukrainian language, Code-switching, Linguistic analysis, Natural language processing (NLP), Fine-tuning, Morphological tagging

ПЕРЕЛІК СКОРОЧЕНЬ

NLP (Natural Language Processing) – Обробка природної мови

POS (Part of Speech) Tagging – Тегування частин мови

LLM (Large Language Model) – Велика мовна модель

RNN (Recurrent Neural Network) – Рекурентна нейронна мережа

LSTM (Long Short-Term Memory) – Довготривала короткочасна пам'ять.

BiLSTM (Bidirectional Long Short-Term Memory) – Біспрямована довготривала короткочасна пам'ять

BERT (Bidirectional Encoder Representations from Transformers) – Біспрямовані представлення енкодера з трансформерів

DAWG (Directed Acyclic Word Graph) – Орієнтований ациклічний граф слів

OOV (Out Of Vocabulary) – Поза словниковий запас

GloVe (Global Vectors for Word Representation) – Глобальні вектори для представлення слів

GPT (Generative Pre-trained Transformer) – Генеративний попередньо навчений трансформер

RoBERTa (Robustly Optimized BERT Approach) – Робастно оптимізований підхід BERT

ВСТУП

Із появою великих мовних моделей (LLM), таких як GPT-3, BERT та їхніх наступників, у галузі обробки природної мови (NLP) відбулася зміна парадигми [47]. Ці моделі, навчені на великих і різноманітних корпусах, продемонстрували значні можливості у вирішенні цілої низки лінгвістичних завдань, перевершуючи ефективність традиційних моделей, розроблених для конкретних застосувань [42]. Одним із таких фундаментальних завдань у NLP є розпізнавання частин мови (Part-of-Speech, POS), що передбачає позначення слів у реченні відповідною частиною мови (наприклад, іменниками, дієсловами, прикметниками). Історично POS-тегування спиралося на статистичні моделі та системи, що базуються на правилах, але поява LLM дає нагоду переоцінити, як ці моделі справляються з цим завданням.

У цьому проєкті досліджується ефективність LLM у POS-маркуванні та з'ясовується, чи дають їхні складні архітектури та великих обсяг навчальних даних перевагу над традиційними моделями. На відміну від традиційних POS-тегерів, які часто обмежуються синтаксичними знаннями, LLM володіють багатими фоновими знаннями і контекстним розумінням, отриманими в результаті навчання на великих наборах текстових даних. Ця багата база знань дозволяє LLM потенційно розуміти нюанси лінгвістичних патернів і контекстуальні тонкощі, які простіші моделі можуть не врахувати.

Крім того, адаптивність LLM до різних лінгвістичних контекстів викликає питання щодо глибини їхнього розуміння мови. Оцінюючи LLM на завданні POS-тегування, ми можемо отримати уявлення про їхнє синтаксичне та семантичне розуміння, а також про те, чи їхнє виконання цього завдання свідчить про глибшу, більш узагальнену лінгвістичну компетенцію.

У цьому дослідженні ми прагнемо порівняти продуктивність LLM з традиційними моделями POS-тегування, використовуючи стандартні набори даних і метрики оцінювання. Ми припускаємо, що LLM з їхніми вдосконаленими архітектурами та попередньо навченими знаннями

перевершать традиційні моделі, демонструючи свій потенціал як більш інтелектуальні та універсальні інструменти для лінгвістичного аналізу. Це дослідження сприятиме нашому розумінню можливостей і обмежень LLM, надаючи більш чітку картину їхньої ролі в майбутньому NLP.

Вивчаючи продуктивність LLM у POS-тегуванні, це дослідження намагається відповісти на ширші питання про їхнє розуміння мови та здібності до узагальнення. Воно також має на меті висвітлити трансформаційний потенціал LLM в комп'ютерній лінгвістиці.

РОЗДІЛ 1. СУЧАСНИЙ СТАН ГАЛУЗІ ЧАСТИНОМОВНОГО ТЕГУВАННЯ

У цьому розділі ми розглянемо сучасний стан галузі частиномовного тегування. Це важлива область обробки природної мови, що стосується автоматичного визначення частин мови для кожного слова у тексті. Ми розглянемо основні принципи та підходи до POS-тегування, включаючи статистичні методи, правило-орієнтовані методи та методи машинного навчання. Також зробимо огляд сучасних інструментів для POS-тегування української мови, таких як Pymorphy, Stanza, SpaCy та Flair з RoBERTa, та обговоримо їхні характеристики і застосування. У цьому контексті ми проведемо аналітичний огляд праць, що висвітлюють різні аспекти використання великих мовних моделей (LLM) у завданнях POS-тегування.

1.1. Поняття частиномовного тегування

POS-тегування (частиномовне тегування) – це процес автоматичного визначення частин мови для кожного слова в тексті та призначення відповідних тегів, що вказують на граматичну категорію слова (наприклад, іменник, дієслово, прикметник тощо). Цілі та задачі POS-тегування включають ідентифікацію частини мови для кожного слова у реченні та покращення розуміння тексту для таких задач, як синтаксичний аналіз, машинний переклад, інформаційний пошук та інші [8].

1.1.1. Застосування POS-тегування

Застосування POS-тегування включає кілька ключових областей. У лінгвістичному аналізі воно допомагає в дослідженні структури мови та граматики, дозволяючи краще розуміти, як слова функціонують у різних контекстах. В інформаційному пошуку та вилученні даних POS-тегування підвищує точність пошукових систем, допомагаючи їм краще розуміти запити користувачів і відповідно знаходити релевантну інформацію. У машинному перекладі POS-тегування полегшує процес перекладу, забезпечуючи правильне узгодження частин мови між різними мовами, що підвищує якість і точність перекладів. POS-тегування також

застосовується для покращення текстової аналітики, автоматичного реферування текстів, сентимент-аналізу та інших задач, що вимагають детального розбору мовних структур [18].

1.1.2. Принципи та підходи до частиномовного тегування

Принципи та підходи до маркування частин мови (POS-тегування) включають три основні методи [17].

Перший з них — правило-орієнтовані методи, що базуються на наперед заданих граматичних правилах, які допомагають визначати POS-теги на основі морфологічних характеристик та взаємозв'язків між словами. Цей метод неможливий без роботи лінгвістичних експертів для створення ефективних правил [10]. Такі системи здатні досить точно визначати POS-теги, особливо в академічних текстах, де вживається стандартна мова і чітке дотримання граматичних норм. З іншого боку, наскільки ефективною буде така система, значною мірою залежить від якості та повноти граматичних правил, які вона використовує. Чим детальніше і всеохоплююче правило, тим краще воно може обробити варіації в тексті. Тому ключовим аспектом розробки правило-орієнтованих систем є неперервна робота над оновленням правил, щоб адаптувати їх до мінливих лінгвістичних реалій та нових мовних явищ [10].

Другий — статистичні методи, які використовують математичні моделі для визначення ймовірностей різних тегів. Приховані марківські моделі моделюють послідовності тегів частин мови, тоді як моделі максимальної ентропії оптимізують класифікацію, зберігаючи максимальну невизначеність до моменту надходження даних [11].

Приховані марківські моделі використовують ідею про те, що частини мови (теги) є прихованими станами, а самі слова є спостереженнями, які генеруються цими станами. Вони моделюють послідовності тегів частин мови [21].

До основних компонентів належать:

- Множина станів S: це теги частин мови.

- Множина спостережень O: це слова в реченні.

- Матриця ймовірностей переходу $A = \{a_{ij}\}$: $a_{ij} = P(s_j \mid s_i)$, де s_i та s_j є станами (тегами).

- Матриця ймовірностей емісії $B = \{b_{jk}\}$: $b_{jk} = P(o_k \mid s_j)$, де o_k є спостереженням (словом), а s_j є станом.

- Початковий розподіл $\pi = \{\pi_i\}$: $\pi_i = P(s_i)$, де s_i є початковим станом.

Формально, ймовірність послідовності тегів $S = s_1, s_2, \ldots, s_T$ для даної послідовності слів $O = o_1, o_2, \ldots, o_T$ визначається як:

$$P(S \mid O) = \pi_{s_1} \prod_{t=2}^{T} a_{s_{t-1}s_t} \prod_{t=1}^{T} b_{s_t o_t}$$

Моделі максимальної ентропії використовують принцип максимальної невизначеності для класифікації, зберігаючи максимальну невизначеність до моменту надходження даних. Основною ідеєю є визначення ймовірності тегів на основі сукупності ознак (features), які можуть бути словами, попередніми тегами, суфіксами, префіксами тощо [38].

Модель максимальної ентропії визначає ймовірність тега y для слова x як:

$$P(y \mid x) = \frac{1}{Z(x)} \exp\left(\sum_i \lambda_i f_i(y, x)\right)$$

де $f_i(y, x)$ — функції ознак, λ_i — ваги функцій ознак та $Z(x)$ — нормалізаційний фактор, який забезпечує, що всі ймовірності в сумі дають 1.

Нормалізаційний фактор обчислюється наступним чином:

$$Z(x) = \sum_{y'} \exp\left(\sum_i \lambda_i f_i(y', x)\right)$$

Третій метод для розпізнавання частин мови – використання глибокого навчання, зокрема нейронних мереж, навчених на великих анотованих корпусах, таких як Penn Treebank. Цей підхід ефективно вловлює контекстуальні залежності в текстах та POS-розмітці завдяки використанню сучасних нейронних мереж, таких як рекурентні та трансформерні [40].

Рекурентні нейронні мережі (RNNs) спеціалізуються на обробці послідовностей даних. Це дозволяє моделям враховувати попередні елементи послідовності, що особливо корисно для розпізнавання частин мови, де контекст попередніх слів може впливати на інтерпретацію поточного слова. Різновид RNN, відомий як LSTM (Long Short-Term Memory), здатний зберігати інформацію протягом довгих інтервалів, що дозволяє моделям враховувати далекі залежності в тексті [46]. Bidirectional LSTM (BiLSTM) йде далі, обробляючи послідовність у двох напрямах — вперед і назад, — що забезпечує врахування як попереднього, так і наступного контексту для кожного слова [9].

Блок LSTM складається з комірки пам'яті та трьох «вентилів» (англ. gates):

- «Забувальний вентиль» (англ. forget gate): визначає, яка інформація буде видалена зі стану комірки.

$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$, де:

- W_f: Вагова матриця «вентилю».
- h_{t-1}: Прихований стан з попереднього кроку.
- x_t: Вхідні дані на поточному кроці.
- b_f: Фактор зсуву.
- σ: Сигмоїдна функція активації, виводить значення між 0 та 1.

- «Входовий вентиль» (англ. input gate) визначає, які значення в стані комірки оновлювати.

$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$

$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$, де:

- i_t: Вихід вхідного вентилю на часовому кроці t, вирішує, які значення у стані комірки потрібно оновити.
- W_i: Вагова матриця.
- $\tilde{C}_t$: Значення-кандидати для оновлення стану комірки.
- W_C: Вагова матриця для генерації значень-кандидатів.
- b_i, b_C: Умови зміщення для генерації значень-кандидатів.
- $\tanh$: Гіперболічна тангенціальна функція активації, виводить значення між -1 та 1.

- «Виходовий вентиль» (англ. output gate) : визначає наступний прихований стан, який впливає на наступний або вихідний рівень,

впливаючи на обробку наступного слова або остаточне рішення щодо тегу.

$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$

$h_t = o_t * \tanh(C_t)$

$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$, де:

- o_t: Активація вихідного вентилю на часовому кроці t, визначає наступний прихований стан.
- W_o: Вагова матриця для вихідного вентиля.
- b_o: Член зсуву для вихідного вентиля.
- h_t: Прихований стан для поточного кроку часу.
- C_t: Оновлений стан комірки на кроці t.
- C_{t-1}: Попередній стан комірки.

BiLSTM розширює традиційні LSTM, додаючи другий рівень, який обробляє вхідну послідовність у зворотному напрямку. Таке налаштування дозволяє моделі мати як минулий, так і майбутній контекст у кожній точці послідовності, що корисно для таких завдань, як POS-тегування, де контекст з обох боків слова часто має вирішальне значення для визначення його правильного тегу.

Результати прямого і зворотного проходів зазвичай об'єднуються на кожному кроці, забезпечуючи набір функцій, які фіксують інформацію як з минулого, так і з майбутнього контексту:

$h_t = [\overrightarrow{h}_t; \overleftarrow{h}_t]$

Після цього ця комбінована функція подається в щільний шар з активацією функції softmax, щоб передбачити POS-тег для кожного токена:

$y_t = \text{softmax}(W_y \cdot h_t + b_y)$, де:

- W_y: Вагова матриця для вихідного шару, відображає приховані стани у вихідний простір.
- b_y: Член зсуву для вихідного шару.
- y_t: Вихід на часовому кроці t, вектор ймовірностей можливих POS-тегів, обчислений за допомогою функції softmax.

Трансформерні мережі [45], зокрема архітектура BERT (Bidirectional Encoder Representations from Transformers), підходять для розпізнавання частин мови завдяки своїй здатності паралельно обробляти весь текстовий вхід, що дозволяє моделі захоплювати глобальний контекст [15]. Трансформери використовують механізм уваги, який дозволяє кожному

слову в тексті «уважно» стежити за іншими словами, таким чином точно визначаючи залежності між словами незалежно від їх відстані в тексті [45].

Створення інструментів для POS-тегування за допомогою нейронних мереж проходить в декілька етапів. Спочатку текст нормалізується і токенізується, а потім анотується на основі існуючих корпусів. Після цього модель навчається на цих анотованих наборах даних, коригуючи свої вагові параметри, щоб забезпечити максимальну точність при визначенні частин мови. Під час навчання модель оптимізує свою функцію втрат, яка зазвичай мінімізує різницю між прогнозованими і фактичними тегами [39].

Переваги використання нейронних мереж для POS-тегування включають високу точність розпізнавання, здатність обробляти складні мовні залежності та контексти, а також гнучкість у застосуванні до різних мов і текстових корпусів. Цей підхід значно покращує якість розпізнавання частин мови завдяки здатності нейронних мереж аналізувати складні патерни у великих обсягах текстових даних. Ці методи інколи комбінуються для створення більш точних та адаптивних систем POS-тегування, які можуть ефективно обробляти різноманітні мовні відмінності [5].

1.2. Огляд наявних інструментів морфологічної розмітки для української мови для української мови

1.2.1. Pymorphy

Pymorphy3 – це морфологічний аналізатор та синтезатор з відкритим вихідним кодом для української та інших слов’янських мов, що використовує великі, ефективно закодовані лексикони, створені на основі даних OpenCorpora та LanguageTool [2].

Pymorphy розроблено як модульну систему з розподілом між аналізом відомих слів і генерацією морфологічних форм для слів, що не входять до словника. Її архітектура включає низку блоків аналізатора, які можна налаштовувати або змінювати, що забезпечує можливості для адаптації у морфологічному аналізі.

Pymorphy спирається на великі лексикони, такі як OpenCorpora, що містить приблизно 5 мільйонів словоформ. Ці лексикони закодовано у вигляді прямого ациклічного графа слів (DAWG), що значно зменшує

використання пам'яті, зберігаючи при цьому високу швидкість доступу для морфологічного аналізу [2].

Спрямований ациклічний граф слів (англ. Directed Acyclic Word Graph, DAWG) – це структура даних, що використовується для представлення наборів рядків у компактній формі. Це тип детермінованого ациклічного скінченного автомата, де кожен унікальний підрядок набору рядків представлений шляхом від спільної початкової точки (кореня). Ця структура особливо корисна для задач, що включають великі словники і потребують швидкого пошуку, таких як перевірка орфографії, передбачення слів та інші програми обробки тексту [14].

Слова, що не входять до словника, Pymorphy обробляє за допомогою серії заздалегідь визначених правил, які можна налаштовувати. Ці правила враховують префікси, суфікси та інші морфологічні маркери слів, щоб призначити частиномовний тег або створити правдоподібні морфологічні форми [2].

Pymorphy підходить до POS-тегування за допомогою набору блоків аналізатора, які застосовують правила для визначення можливих граматичних категорій і флексій слів, яких немає в словнику.

Система використовує комбінацію лексичної інформації та морфологічних моделей, щоб обмежити кількість можливих результатів аналізу для кожного слова, особливо для слів OOV (англ. out of vocabulary). Наприклад, якщо слово не знайдено в словнику, Pymorphy висуне гіпотезу про його POS на основі суфікса слова, відомих морфологічних правил і контексту, в якому воно з’являється. Цей метод допомагає зменшити неоднозначність тегів POS, що часто є проблемою для мов, які мають великий набір можливих морфологічних тегів.

Морфологічний аналізатор Pymorphy може повертати кілька можливих аналізів слів. Проблема вибору правильного аналізу зі списку можливих варіантів називається усуненням неоднозначності. Як правило, щоб вибрати правильний аналіз, необхідно враховувати контекст слова. Морфологічний аналізатор Pymorphy приймає окремі слова як вхідні дані, тому він не може надійно усунути неоднозначність результату. Однак він може надати оцінку умовної ймовірності P(аналіз|слово) [2].

Для оцінки умовної ймовірності P (аналіз|слово) для російських слів Pymorphy використовує корпус частково позбавлений граматичної омонімії

OpenCorpora [12] і припускає, що P(аналіз|слово) = P(тег|слово). Умовна ймовірність оцінюється для слів, які мають багаторазовий аналіз відповідно до pymorphy2, але мають випадки з одним аналізом в корпусі OpenCorpora; оцінка є оцінкою максимальної правдоподібності зі згладжуванням за Лапласом.

$$P_{\mathrm{MLE}}(\mathrm{tag}|\mathrm{word}) = \frac{\mathrm{count}(\mathrm{word}, \mathrm{tag}) + 1}{\mathrm{count}(\mathrm{word}) + B(\mathrm{word})}$$

де count – частотність.

Щоправда, для української мови ймовірності OOV слів розподіляються рівномірно, оскільки «на момент написання статті не існує вільно доступного українського корпусу, подібного до OpenCorpora.» [2]

Наразі вже існує аналог OpenCorpora для української [12], який є роботою Дмитра Чаплинського, Юлії Романишин та інших і є конвертатором розмічених корпусу з проєкту LanguageTool [24] у формат OpenCorpora. Цей проєкт було згадано у статті [2] як такий, що знаходиться в розробці.

Проте згідно з файлом «CHANGES.rst» у репозиторії бібліотеки [35], усунення неоднозначності зі згладжуванням не було додано для української.

1.2.2. Stanza

Stanza – це інструментарій для обробки природної мови (NLP), який використовує повністю нейронний конвеєр лінгвістичного аналізу та підтримує широкий спектр мов і завдань NLP. Архітектура Stanza особливо ефективна для обробки таких завдань, як токенізація, тегування частин мови (POS) та морфологічних ознак, серед інших [36].

Механізм POS-тегування Stanza починається з первинної обробки вхідного тексту, який спочатку токенізується і групується в речення. Це досягається за допомогою моделі, яка передбачає, чи означає символ кінець токена, речення або є частиною багатослівного токена.

Для самого POS-тегування Stanza використовує двонаправлену мережу довготривалої короткочасної пам'яті (Bi-LSTM), яка є різновидом

рекурентної нейронної мережі (RNN), що може обробляти послідовності даних (наприклад, слова в реченні) як в прямому, так і в зворотному напрямку [36].

Крім цього, Stanza використовує біафний скоринг [16], за допомогою якого встановлюються типи залежностей між словами в реченні – процес, який не тільки покращує POS-тегування, але й допомагає в розборі структури речення [36].

Механізм глибокої біафної уваги покращує аналіз речень на їхні синтаксичні структури, зокрема дерева залежностей. Він включає два основних компоненти: біафну увагу для прогнозування дуг та біафні класифікатори для прогнозування міток [16, 50].

Для кожної пари слів у реченні модель біафної уваги обчислює оцінку, яка вказує на ймовірність того, що одне слово є синтаксичною головою іншого. Ця оцінка виводиться з використанням репрезентацій потенційних головних та залежних слів, трансформованих глибокою нейронною мережею.

Після прогнозування дуг інший біафний класифікатор прогнозує тип синтаксичного зв'язку (мітку), який існує між головою та залежним. Це включає схожий підхід глибокого навчання, але зосереджений на типах відносин (наприклад, суб'єкт, об'єкт тощо), а не на їх існуванні [16].

Прогнозування дуг: $s^i_{\text{arc}} = (\mathbf{R}^{(1)}\mathbf{r}_i) + (\mathbf{R}^{(2)}\mathbf{r}_j)$

де $\mathbf{r}^i$ та $\mathbf{r}^j$ — вихідні вектори BiLSTM, що представляють слова, а $\mathbf{R}^{(1)}$ та $\mathbf{R}^{(2)}$ — трансформаційні матриці, що застосовуються в біафінній трансформації для прогнозування дуг.

Прогнозування міток: $s^i_{\text{label}} = \mathbf{r}_j^T\mathbf{U}^{(1)}\mathbf{r}_i + (\mathbf{r}_j \oplus \mathbf{r}_i)^T\mathbf{U}^{(2)} + \mathbf{b}$

де $\mathbf{U}^{(1)}$ та $\mathbf{U}^{(2)}$ — параметричні матриці для біафінної трансформації, специфічної для прогнозування міток, а $\mathbf{b}$ — вектор зсуву.

Хоча модель глибокої біафної уваги перш за все націлена на синтаксичний аналіз залежностей, збагачені репрезентації слів та інформація про синтаксичну структуру, яку вона генерує, можуть опосередковано покращити тегування частин мови. В інтегрованій моделі, де тегування частин мови та синтаксичний аналіз залежностей поєднуються, синтаксичний контекст, наданий аналізатором залежностей, може допомогти в розрізненні тегів частин мови. Знання синтаксичної ролі

слова у своїй фразі або реченні може допомогти з проблемою граматичної омонімії [16].

1.2.3. Spacy

POS тегування за допомогою SpaCy використовує згорткові нейронні мережі (CNN) для оброблення тексту; цей процес включає перетворення слів у векторні представлення (ембедінги) за допомогою попередньо навчених моделей, таких як GloVe, а також застосування контекстуалізованих ембеддінгів для забезпечення точного аналізу контексту, після чого модель прогнозує POS тег для кожного слова, враховуючи його позицію та взаємодію з іншими словами в реченні, що дозволяє досягти високої точності у визначенні частин мови [41, 49].

З виходом spaCy v3.0 було додано нові функції та компоненти. Найсуттєвішою новою функцією, безсумнівно, стали трансформерні пайплайни. Нові пайплайни на основі трансформерних нейронних мереж підвищують точність spaCy. Інтеграція моделей-трансформерів у конвеєр spaCy [41]. До v3.0 конвеєр spaCy виглядав наступним чином:

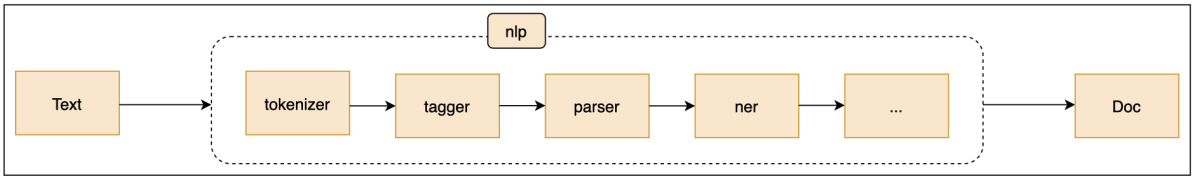


Рис. 1. Компоненти пайплайну SpaCy, базованого на векторному підході (до v3.0) [28].

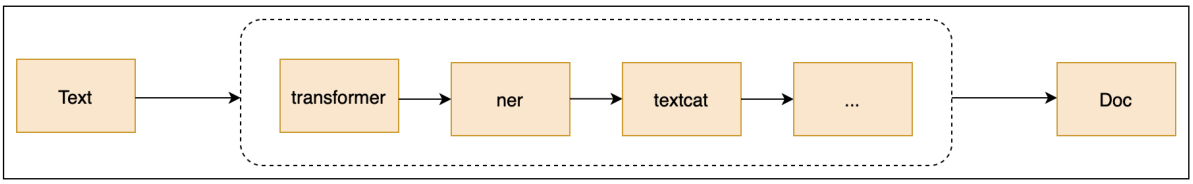


Рис. 2. Компоненти пайплайну spaCy, базованого на трансформерному підході (v3.0) [28].

З ілюстрації можна помітити, що елементи «токенізатор» та «тегер» були замінені трансформерним шаром. Незважаючи на новий

трансформерний пайплайн, з випуском версії 3.0 все ще підтримуються векторні моделі spaCy v2.

1.2.4. Flair

POS тегування за допомогою Flair працює за принципом використання двонаправлених LSTM мереж та контекстуалізованих ембеддінгів, які називаються «Flair embeddings»; ці ембеддінги враховують контекст кожного слова шляхом обробки тексту з обох боків, створюючи вектори, що змінюються в залежності від контексту, після чого модель використовує ці контекстуалізовані векторні представлення для прогнозування POS тегів [7].

1.3. Використання LLM для частиномовної розмітки

Великі мовні моделі (LLM) є прогресом у сфері обробки природної мови. Поява LLM докорінно змінила підхід до завдань обробки мови та їх виконання. Ці моделі, такі як GPT від OpenAI та BERT від Google, розроблені для розуміння та генерації мовлення.

У контексті POS-тегування традиційні моделі зазвичай покладаються на анотовані набори даних і створені вручну функції та розмічені датасети для класифікації слів за відповідними частинами мови. Хоча певною мірою ефективні, рідкісні слова, малоресурсні мови, неоднозначний контекст, граматична омонімія та динамічна природа мови становлять для них проблему [22]. LLM, з іншого боку, використовують свої архітектури глибокого навчання для динамічного створення контекстних ембедингів, які відображають не лише синтаксичні, а й семантичні властивості слів у реченні. Це дозволяє їм потенційно ефективніше розпізнавати значення слів і проводити тегування з високою точністю навіть для вищезгаданих проблемних випадків [29].

RoBERTa (Robustly optimized BERT approach) — це тип трансформерної моделі на основі архітектури BERT, розроблена дослідниками з Facebook AI, яка використовує методи попереднього навчання на великих текстових корпусах з оптимізацією гіперпараметрів і видаленням маскованих токенів для покращення продуктивності; при

виконанні POS тегування за допомогою RoBERTa модель застосовує свої багатошарові контекстуалізовані ембеддінги для аналізу тексту, де кожне слово представлено вектором, який враховує його контекст у реченні [25].

Адаптивність LLM виходить за межі звичайного POS-тегування. Наприклад, під час виконання завдання сентимент-аналізу великі мовні моделі можуть більш точно визначати настрої нюансованих ідіоматичних сполук і контекстно-залежних підказок, перевершуючи продуктивність традиційних моделей, які можуть покладатися на функції поверхневого рівня або підходи на основі ключових слів [29]. У машинному перекладі LLM показали значні покращення у плавності та точності, виробляючи переклади, які є контекстуально та граматично правильними [26, 48].

Загалом, великі мовні моделі втілюють зсув до більш цілісного та контекстно-орієнтованого розуміння мови. Їхня здатність інтегрувати синтаксичні, семантичні та світові знання дозволяє їм виконувати широкий спектр завдань обробки мови з високою точністю та гнучкістю, що робить їх незамінними інструментами NLP. Оскільки дослідження та розробки в LLM продовжують розвиватися, очікується, що їхній вплив на сферу NLP зростатиме, прокладаючи шлях для більш інтелектуальних та інтуїтивно зрозумілих технологій на основі мови [26, 43].

1.3.1. Аналітичний огляд праць

У статті «Language Models are Unsupervised Multitask Learners» [37] висвітлено кілька ключових висновків про здатність мовних моделей виконувати різноманітні завдання та те, як вони спонукаються до цього.

Навчаючись моделювати, мовні моделі можуть виконувати багато завдань, що по суті залежить не лише від вхідних даних, але й від самого завдання. Це формалізовано в налаштуваннях багатозадачності та метанавчання, де кондиціонування завдання може бути реалізовано на архітектурному або алгоритмічному рівнях. Крім того, мовні моделі, особливо великі, такі як GPT, можуть виконувати різноманітні завдання в zero-shot режимі, тобто вони можуть обробляти нові завдання без будь-якого явного нагляду чи модифікації параметрів, специфічних для завдання.

Цей метод дозволяє навчити одну модель виконувати багато різних завдань на основі структури та вмісту вхідної послідовності. Для zero-shot навчання моделі надається підказка, яка неявно пропонує завдання. Наприклад, щоб створити підсумок, моделі можна надати статтю, за якою слідує підказка «TL;DR:». Здатність виконувати конкретні завдання може бути викликана наданням відповідного контексту та прикладів у вхідній послідовності.

На завершення, дослідження «Language Models are Unsupervised Multitask Learners» [37] демонструє, що за наявності достатньої потужності мовні моделі можуть навчитися виконувати широкий спектр завдань без явного нагляду, включно з завданням POS-тегування.

У статті «Evaluating large language models for the tasks of PoS tagging within the Universal Dependency framework» [27] автори досліджують можливості великих мовних моделей (LLM) у виконанні тегування частин мови для бразильської португальської мови за допомогою розміченого корпусу UD. Дослідження оцінює три моделі: GPT-3, LLaMA-7B і Maritaca.

У дослідженні використовувався датасет Porttinari, різносторонній корпус для бразильської португальської мови, особливо зосереджуючись на його журналістській частині. Завдання передбачало анотування перших 1010 речень із цього набору даних, використовуючи перші 10 речень як приклади для LLM у завданні тегування частин мови.

Моделям було надано вхідну інструкцію, щоб спонукати їх виконувати частиномовне тегування, звертаючи увагу на обробку скорочень і комбінованих слів. Параметр температури, який контролює випадковість відповідей моделі, був підібраний для кожної моделі, щоб оптимізувати баланс між точністю та запам’ятовуванням.

Експерименти показали, що GPT-3 досягла найвищих показників продуктивності, потім LLaMA-7B, а потім Maritaca. Модель GPT-3 продемонструвала значну перевагу F-міри та точності, особливо при налаштуванні температури 0,0. LLaMA показала найкращі результати при температурі 0,2, а Maritaca — 0,1.

Точність, recall та F-міра для кожного POS-тега були записані. GPT-3 показав високу точність у більшості тегів, з особливо високою ефективністю в таких категоріях, як NOUN, PROPN і VERB. Однак деякі

теги, такі як SCONJ (підрядні сполучники) і AUX (допоміжні дієслова), показали меншу точність. LLaMA та Maritaca також продемонстрували достатню продуктивність, але мали проблеми з певними тегами та загалом показали нижчі показники порівняно з GPT-3.

У дослідженні зроблено висновок, що LLM, зокрема GPT-3, є ефективними інструментами для POS-тегування бразильською португальською мовою в рамках корпусу UD. Результати свідчать про те, що LLM можуть служити початковими тегерами, забезпечуючи цінний бейзлайн, який можна вдосконалити за допомогою перевірки людиною. Дослідження підкреслює потенціал LLM у зменшенні потреби у великих наборах даних, анотованих вручну, і припускає, що ці моделі продовжуватимуть удосконалюватися, що створить ще кращу продуктивність у майбутньому.

У статті «Comparing LLM prompting with Cross-lingual transfer performance on Indigenous and Low-resource Brazilian Languages» [6] автори прагнуть оцінити продуктивність великих мовних моделей (LLM) у завданнях тегування частин мови для низькоресурсних мов (LRL). Дослідження зосереджено на 12 мовах із низьким ресурсом із Бразилії, 2 з Африки та 2 мовах із високим рівнем ресурсу (HRL), таких як англійська та бразильська португальська. Дослідження порівнює результати LLM, зокрема GPT-4, із методами міжмовної передачі (transfer learning), які використовує XLM-R, багатомовна модель.

У рамках вищезгаданого дослідження, моделі GPT-4 було запропоновано виконати завдання POS-тегування за допомогою структурованого підходу, де моделі було надано опис завдання разом із прикладами. Запити були розроблені відповідно до структури універсальних залежностей (UD) для узгодженості. Модель XLM-R використовувалася для виконання zero-shot перенесення з англійської та бразильської португальської на цільові мови з низьким ресурсом. Крім того, дослідники виконали адаптивнй файн-тюнінг, використовуючи обмежену кількість одномовних даних із корпусу Біблії, щоб покращити її продуктивність на цільових мовах.

Дослідження показало, що продуктивність GPT-4 і XLM-R значно відрізнялася в різних мовах.

Хоча точність прогнозування POS-тегів для таких високоресурсних мов, як англійська та португальська, була високою (близько 98%), продуктивність значно впала для мов із низьким ресурсом. GPT-4 досягла кращих результатів, ніж базові методи перехресного zero-shot перенесення, але все ще мала проблеми з деякими мовами через відсутність доступу до них під час попереднього навчання.

Файн-тюнінг значно покращив продуктивність XLM-R на кількох мовах із низьким ресурсом. Точність зросла на 3-12 процентних пунктів для шести з семи мов, де доступні дані біблійного корпусу. Це демонструє ефективність файн-тюнінгу навіть невеликих обсягів одномовних даних.

Поширені помилки включали неправильне тегування слів через орфографічну схожість з англійськими словами. Наприклад, GPT-4 позначив допоміжне дієслово мови Karo «okay» як англійське вставне слово.

GPT-4 загалом перевершила метод міжмовної передачі (cross-lingual transfer), але не була безпомилковою. Середня точність POS-тегування для бразильських LRL із GPT-4 становила 33,8% порівняно з 27,1% для міжмовної передачі з португальської за допомогою XLM-R.

У дослідженні було зроблено висновок, що LLM, такі як GPT-4, можуть слугувати корисними початковими тегерами для мов із низьким ресурсом, особливо якщо їх доповнювати перевіркою людиною. Методи міжмовного перенесення, особливо в поєднанні з мовною адаптацією, є багатообіцяючими, але потребують подальшого вдосконалення. Дослідження підкреслює потребу в більших ресурсах для низькоресурсних мов і припускає, що навіть невелика кількість анотованих даних може значно підвищити ефективність моделей.

Висновки до Розділу 1

Огляд сучасного стану сфери POS-тегування демонструє, що ця область значно еволюціонувала завдяки прогресу в машинному навчанні та розробці великих мовних моделей (LLM). Статистичні методи, правило-орієнтовані методи та нейронні мережі пропонують різні підходи до автоматичного визначення частин мови, кожен з яких має свої переваги та недоліки. Інструменти, такі як Stanza, SpaCy та Flair з RoBERTa,

показують високу морфологічного тегування українськомовних текстів. Проведений аналітичний огляд праць підтверджує, що великі мовні моделі, такі як GPT-3 та RoBERTa, демонструють високу продуктивність і точність у завданнях POS-тегування, перевершуючи інші підходи в низькоресурсних випадках. Це сприяє подальшому розвитку NLP та його застосуванню у різних сферах, що вимагають точного і глибокого розуміння мовних структур.

РОЗДІЛ 2. РОЗРОБКА МОДЕЛІ ДЛЯ АВТОМАТИЧНОГО ТЕГУВАННЯ ЧАСТИН МОВИ В УКРАЇНСЬКОМУ ТЕКСТІ НА ОСНОВІ A GPT-3.5

У попередній роботі [3], що слугує основою для цього проєкту, була розроблена модель тегування на основі базової моделі GPT-3, адаптована за допомогою процедури файн-тюнінгу. Однак, враховуючи стрімкий розвиток технологій GPT, модель минулого року та використаний API для її файн-тюнінгу набули статусу застарілих (legacy). Для забезпечення ефективності подальших досліджень було прийнято рішення перейти до використання найновішої доступної для файн-тюнінгу моделі на час написання – GPT-3.5.

На жаль, нам не вдалося отримати доступ до файн-тюнінгу моделі-флагмана GPT-4, незважаючи на заздалегідь поданий запит кількома місяцями раніше.

За інформацією від OPENAI [32], Completions API, що було застосовано у попередній роботі, вже не відповідає сучасним вимогам, подібно до версії GPT-3, яка була попередньо використана. Таким чином, необхідно здійснити міграцію коду та нашого тренувального датасету до актуальної версії Chat Completions API.

Основна різниця між Completions API та Chat Completions API полягає у їхній функціональності. Completions API зосереджувався на генеруванні завершень тексту, тоді як Chat Completions API орієнтований на ведення інтерактивних діалогів. Legacy моделі, що використовували Completions API, мали обмежені можливості для діалогової взаємодії, тоді як нові моделі, що використовують Chat Completions API, здатні підтримувати контекстуально багаті та послідовні розмови, забезпечуючи більш природну та ефективну взаємодію з користувачем [32].

2.1. Формування набору даних для дотренування

Отже, формат датасету змінився з

`{"prompt": "<prompt text>", "completion": "<ideal generated text>"}` на `{"messages": [{"role": "system", "content": "Системне повідомлення, інструкція"}, {"role": "user", "content": "Повідомлення користувача"}, {"role": "assistant", "content": "Бажана відповідь"}]}`, тобто додалося системне повідомлення для моделі.

Для дотренування було використано 1/10 частину датасету Universal Dependencies [45] для української мови. Цей датасет є найбільш доступним і поширеним для української мови і був використаний і при тренуванні інших моделей з якими ми проводимо порівняння у цьому проєкті, як от Flair та RoBERTa.

Датасет UD Ukrainian [45] представляє собою анотовані файли розширення .conllu, поділені на частини для тренування та валідації. Українські дані датасету UD відповідають формату CoNLL-U з такими особливостями:

1. Межі документів вказуються як newdoc_id = ...
2. Межі параграфів на рівні речень вказуються як newpar_id = ...
3. Заголовки документів вказуються як doc_title = ...
4. Автори документів вказуються як author = ...
5. Джерела документів вказуються як source = ...
6. Транслітерація на кшталт чеської присутня як translit = ...

Було вирішено для дотренування використовувати лише пару «речення–частиномовні теги». У цьому випадку в ‘user message’ подаються неанотовані речення, а в ‘assistant message’ – теги частин мови, розділені комою й пробілом. Згідно з актуальною документацією OpenAI API [39], для файн-тюнінгу GPT-3.5 необхідно використовувати .jsonl датасет у наступному форматі:

```
{"messages": [{"role": "system", "content": "Системне повідомлення, інструкція"], {"role": "user", "content": "Повідомлення користувача"}, {"role": "assistant", "content": "Бажана відповідь"}]}
```

Отже, аби привести UD Ukrainian [45] до потрібного для файн-тюнінгу GPT-3.5 вигляду, було написано скрипт мовою програмування Python, повна версія якого знаходиться в github-репозиторії[1] проєкту.

Згаданий скрипт виконує кілька операцій для обробки та конвертації даних з формату CoNLL-U до JSONL та працює наступним чином:

У функції conllu_to_json вхідний файл у форматі CoNLL-U зчитується рядок за рядком. Кожен рядок перевіряється на початкові символи для визначення коментарів, які пропускаються. Порожні рядки

[1] https://github.com/kiryaa865/uni-thesis4

використовуються як роздільники між реченнями. Заповнені рядки розбиваються на поля, і якщо є їх достатня кількість (не менше 4), витягуються слова та частини мови. Кожне речення представляється у форматі словників з ключами «word» та «pos» (слово та частина мови відповідно). Отримані речення зберігаються у форматі JSON у вихідний файл.

У наступній частині коду обробляється отриманий JSON-файл. Кожне речення розбирається на окремі слова та частини мови, які об'єднуються в рядки. Для кожного речення створюється новий запис у форматі, придатному для подальшого використання в якості набору даних для моделі. Кожен запис містить повідомлення користувача (вхідний текст), системне повідомлення та повідомлення асистента (вихідний текст), де вхідний текст складається з токенів речення, а вихідний містить теги частин мови, розділені пробілами. Отриманий оброблений набір даних зберігається у новий JSON файл.

Після цього функція 'convert_json_to_jsonl' приймає вихідний JSON файл і конвертує його до формату JSONL. Для цього вихідний файл зчитується, а потім кожен елемент у файлі перетворюється у рядок JSONL та записується у вихідний файл з розширенням '.jsonl'. Кожен рядок файлу містить окремий запис.

У випадку набору тренувальних даних для генеративної моделі для тегування частин мови, 'user message' є вхідним рядком, який містить лише текст та не має ніякої додаткової інформації. За допомогою 'system message' у датасеті ми встановлюємо інструкцію моделі.

Текст інструкції: «You are Postagger, an expert bot designed to perform Part-of-Speech (POS) tagging accurately for any language, including but not limited to Ukrainian and English. You will provide precise POS tags for all input text, and attempt to tag all words correctly. Keep an eye out for misspellings and attempt to tag them appropriately based on context. Always respond only with the tagged text and nothing else. The tagset you will use includes: NOUN, PUNCT, VERB, ADJ, ADP, ADV, PRON, CCONJ, DET, PART, PROPN, SCONJ, NUM, AUX, INTJ, SYM. The response format is token: pos/n.»

Український переклад наступний: «Ви Postagger, бот-експерт, призначений для точного передбачення тегів частин мови (POS) для

будь-якої мови, включаючи, але не обмежуючись, української та англійської. Ви надасте точні POS теги для всього введеного тексту та спробуєте правильно позначити всі слова. Слідкуйте за орфографічними помилками та намагайтеся позначати їх належним чином відповідно до контексту. Завжди відповідайте лише тегованим текстом і більше нічим. Набір тегів, який ви використовуватимете, включає: NOUN, PUNCT, VERB, ADJ, ADP, ADV, PRON, CCONJ, DET, PART, PROPN, SCONJ, NUM, AUX, INTJ, SYM. Формат відповіді: pos/n.»

Для тексту інструкції було обрано англійську мову через припущення про те, що LLM мають певну мовну упередженість через велику частку англійської мови у тренувальних датасетах [13].

У тексті інструкції було зосереджено увагу на орфографічних помилках та надано завдання тегувати абсолютно всі слова, навіть невідомі, виходячи з контекстної інформації.

У результаті було отримано набір тренувальних даних у форматі .jsonl, що має 5000 рядків у форматі «вхідний текст – теги, розділені пробілами». Використаний тренувальний набір даних склав 1/10 частину датасету UD Ukrainian IU.

У якості експерименту для дослідження продуктивності й інструкції-орієнтованості файн-тюнінгу до датасету було додано 10 текстів, написаних з українсько-англійським код-світчингом, а саме текстів, у яких присутня лексика англійської мови транскрибована українською. Для них було виконано експертну розмітку із додатковими поясненнями.

Один із згаданих текстів у датасеті має наступний вигляд:

["messages": [["role": "system", "content": "[текст вищезгаданого системного повідомлення]"}, {"role": "user", "content": "ні ні, це мої ішус з ним. я рзумію що це не так але я кент гет овер майселф"}, {"role": "assistant", "content": "INTJ INTJ PUNCT PRON PRON NOUN(eng. «issues») ADP PRON PUNCT PRON VERB(ed. «розумію») SCONJ PRON PART ADV CCONJ PRON VERB(eng. «can't») VERB(eng. «get») ADP(eng. «over») PRON((eng. «myself»))"}]}

У випадку одруківки у вхідному тексті було зроблено додаткову позначку із вказанням орфографічно-коректної форми, а для транскрибованих англійських слів було вказано відповідник латинською абеткою.

Тренувальний набір даних у повному обсязі знаходиться у github-репозиторії[2] проєкту.

2.2. Файн-тюнінг моделі

Для дотренування (файн-тюнінгу) GPT-3.5 на отриманому наборі даних було написано скрипт тренування, що знаходиться у github-репозиторії проєкту. Оскільки файн-тюнінг GPT-3.5 доступний лише через OpenAI API [31], було застосовано його.

Гіперпараметри, які були використані для файн-тюнінгу моделі, наведені нижче:

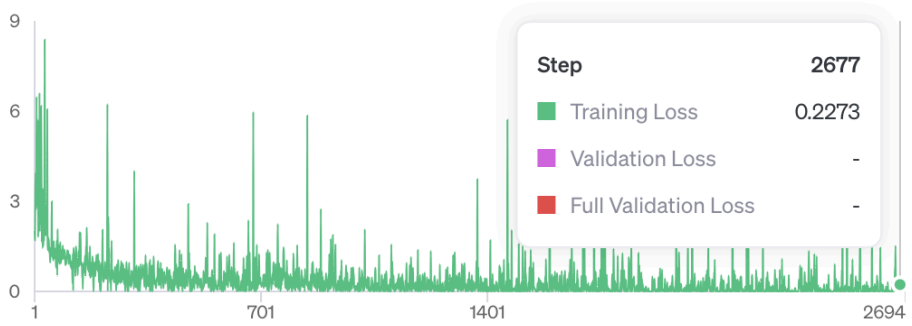
1. Розмір пакета (batch_size): 1. Цей параметр вказує на кількість прикладів даних, яка обробляється моделлю за один раз під час навчання.

2. Множник швидкості навчання (learning_rate_multiplier): 16. Цей параметр визначає коефіцієнт, який використовується для множення швидкості навчання базової моделі під час процесу файн-тюнінгу. Він дозволяє масштабувати швидкість навчання на основі конкретних вимог до файн-тюнінгу та характеру набору даних. Для використаного набору даних та поставленого завдання обраний показник показав себе найкраще.

Кількість епох (n_epochs): 3. Цей параметр вказує на загальну кількість проходів моделі через весь набір даних під час навчання.

2.3. Результати та метрики тренування

Втрата під час тренування склала 0,2273, що свідчить про можливе перенавчання моделі на тренувальному наборі даних [20].



[2] https://github.com/kiryaa865/uni-thesis4

Рис. 3. Графік втрати під час файн-тюнінгу моделі GPT-3.5

Висновки до Розділу 2

У Розділі 2 було описано процес файн-тюнінгу моделі для автоматичного тегування частин мови в українському тексті на основі сучасної версії трансформера GPT-3.5. Зважаючи на застарілість попередніх моделей і API, було здійснено міграцію до новішого Chat Completions API, який краще підходить для ведення інтерактивних діалогів.

Для створення датасету у форматі JSONL було розроблено скрипт, який конвертує дані з формату CoNLL-U, що використовується в датасеті Universal Dependencies для української мови. Цей процес включав обробку та конвертацію анотованих файлів у формат, придатний для файн-тюнінгу GPT-3.5. Системне повідомлення було сформульовано англійською мовою для зниження ризику мовної упередженості, зосереджуючи увагу на тегуванні всіх слів, включаючи орфографічні помилки та транскрибовані англійські слова.

З метою дотренування було використано 1/10 частину датасету UD Ukrainian IU та додано тексти з українсько-англійським код-світчингом, що забезпечило кращу адаптацію моделі до тегування текстів в сучасному неформальному українському мовленні, зокрема до розпізнавання частин мови в текстах, написаних змішаною мовою зі сленгом, одруківками та транскрибованими іншомовними словами.

У результаті дотренування було отримано значення втрати 0.22, яке означає, що модель успішно опрацювала та запам’ятала тренувальні приклади.

РОЗДІЛ 3. ТЕСТУВАННЯ ТА ПОРІВНЯННЯ МОДЕЛЕЙ ЧАСТИНОМОВНОГО ТЕГУВАННЯ

Для тестування отриманої моделі було вирішено порівняти її з найбільш поширеними конвенційними засобами автоматичної розмітки частин мови. До нашого списку ввійшли: Pymorphy3, Stanza, Spacy, Flair [17] та RoBERTa [23]. Останні дві моделі були натреновані на датасеті UD Ukrainian, 1/10 частина якого була використана для файн-тюнінгу GPT-3.5 для частиномовної розмітки.

У цьому розділі порівняємо результати роботи моделей, які представляють всі головні підходи до POS-тегування:

1) Pymorphy3, що працює за допомогою граматичних правил та статистики;
2) Spacy на архітектурі CNN;
3) Flair на архітектурі RNN (LSTM);
4) Stanza на архітектурі RNN (Bi-LSTM);
5) моделі з трансформерною архітектурою GPT-3.5 та RoBERTa.

Детально проаналізуємо та спробуємо пояснити отримані результати POS-тегування за допомогою експертного лінгвістичного аналізу.

Для оцінки якості роботи дотренованої моделі для тегування текстів широкого вжитку (літературні тексти публіцистичного та наукового жанрів), було вирішено обрахувати рівень узгодженості з моделями-флагманами. Spacy, Flair, Stanza, RoBERTa – state-of-the-art моделі з відомою заявленою точністю POS-тегування близько 98% [17].

Для цього було зібрано тексти різних жанрів розміром в 7139 токенів та оброблено їх за допомогою створеної програми-агрегатора. Для обрахунку узгодженості було створено скрипт, що знаходиться в github-репозиторії[3] проєкту. Результати узгодженості наступні:

	GPT-3.5	Stanza	SpaCy	Flair	RoBERTa	Pymorphy3
GPT-3.5		91.539431	91.245272	91.077182	91.329318	68.244852
Stanza	91.539431		97.534669	97.324555	97.212495	68.987253
SpaCy	91.245272	97.534669		96.834291	97.562684	68.637064

[3] https://github.com/kiryaa865/uni-thesis4

Flair	91.077182	97.324555	96.834291		96.834291	68.454966
RoBERTa	91.329318	97.212495	97.562684	96.834291		68.581034
Pymorphy3	68.244852	68.987253	68.637064	68.454966	68.581034	

Табл. 1. – Результати обрахунку узгодженості моделей в тегуванні вибірки літературних текстів

З даних таблиці можна дійти висновку, що моделі Stanza, SpaCy, Flair та RoBERTa передбачають однакові теги в ≈ 97% випадків. Дотренована GPT-3.5 відхиляється від цього числа на ≈ 6%, а Pymorphy3 – на 29%.

Відповідно, результати тегування дотренованої моделі на текстах загального вжитку є співставними з результатами сучасних моделей для POS-тегування з відхиленням в ≈ 6%.

Таблиця з результатами тегування згаданого тексту всіма моделями, що порівнюються, знаходиться в Додатку 1 роботи.

З порівняльної таблиці помітно, що більшість відмінностей трапилися в тегуванні займенників. Частою особливістю дотренованої моделі є передбачення «PRON» (займенник) у випадках, коли решта моделей окрім Pymorphy3 передбачають тег «DET» (артикль, вказівник). Така поведінка хоч і не є помилковою, потребує допрацювання.

Загалом, дотреновану модель можна застосовувати для тегування літературних текстів на рівні з іншими сучасними моделями.

3.1. Збір текстів для тестування

Для практичного тестування якості моделей, що порівнюються, було вирішено обрати тексти, відмінні від тих, з яких складається тренувальний набір даних. Для цього було створено форму для збору текстів. Для поглибленого тестування було обрано саме тексти повідомлень із месенджерів, оскільки вони репрезентують справжній стан сучасного розмовного українського писемного мовлення з орфографічними, лексичними та іншими помилками, а також суржиком і зміною мов [1].

Інтернет-мовлення максимально наближене до розмовного стилю не тільки через легкість комунікації, але й завдяки свідомому чи несвідомому порушенню правил літературної мови. Це є характерною рисою не лише

українського мовлення в месенджерах, але й загальносвітовою тенденцією до спрощення та свідомого імітування помилок у неформальному усному та письмовому спілкуванні. Такі прояви неграмотності, свідомі чи несвідомі, найкраще ілюструють поточний стан розвитку сучасного мовлення і відображають ставлення суспільства до цього явища. За словами О. Тараненка, сьогодні в українській мові спостерігається чітка тенденція до зниження рівня літературної мови, її орозмовлення та нехтування мовними нормами [4].

Тестування засобів POS-тегування на текстах повідомлень із месенджерів є оптимальним, оскільки дозволяє помітити слабкі сторони кожної з моделей завдяки високій відносній частоті лінгвістичних категорій, що становлять проблему в галузі частиномовного тегування загалом.

POS-тегування текстів онлайн-комунікацій є актуальним для дослідження та тестування, оскільки воно в тому чи іншому вигляді є однією зі складових пошукових систем, включно з інтернет пошуковими системами, системами машинного перекладу та ідентифікації іменованих сутностей в тому чи іншому вигляді. Тобто покращення систем частиномовного тегування може покращити й інші системи обробки природньої мови [4].

Двадцяти дев’яти респондентам було запропоновано надати тексти власних повідомлень, у яких наявний хоча б один артефакт із наданого списку. До нього входили:

1. Вкраплення іншомовних слів транскрибовані українською.
2. Вкраплення іншомовних слів латинською/іншою абеткою.
3. Сленг
4. Одруківки

Також респонденти мали можливість самостійно вказати особливості наданих ними текстів, якщо їх немає в списку запропонованих. Цією можливістю скористалися двоє респондентів, які додали категорії навмисних одруківок та навмисних орфографічних помилок. Статистику відповідей форми наведено на Рисунку 4.

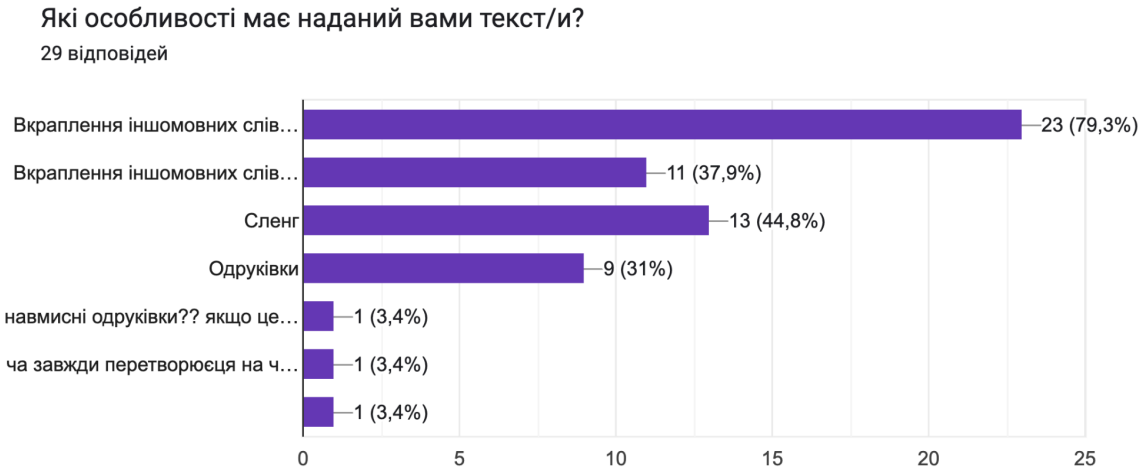


Рис. 4. – Статистика відповідей форми для збору текстів

Усього було зібрано 436 текстів. Усі респонденти надали згоду на обробку та використання текстів у межах цього проєкту.

Додатково було створено та додано 10 текстів для тестування проблеми граматичної омонімії, оскільки в наданих респондентами текстах таких випадків було недостатньо для поглибленого аналізу цієї проблеми.

Повний файл із текстами знаходиться в Додатку А. Загальна кількість токенів склала 8114.

3.2. Розробка застосунку для порівняльного аналізу

Для зручності аналізу було створено інструмент, що зводить результати тегування всіх засобів, що порівнюються, в одну таблицю та підсвічує розбіжності. Засіб було створено за допомогою мови програмування Python. Для зручності його поширення та спільного використання, додаток було адаптовано для веб-застосування за допомогою фреймворку Streamlit, який дозволяє запускати застосунки на хмарних обчислювальних потужностях. Повний програмний код знаходиться в репозиторії проєкту[4]. Код працює наступним чином:

[4] https://github.com/kiryaa865/uni-thesis4

Спочатку імпортуються необхідні бібліотеки та модулі, включаючи stanza, spacy, pymorphy3, flair та transformers, для забезпечення різних технік розмітки частин мови. Бібліотека openai використовується для взаємодії з API OpenAI для розмітки частин мови через асистента.

Функція ‘install_spacy_model’ забезпечує завантаження та завантаження моделі spacy для української мови, кешуючи модель для оптимізації продуктивності. Функція ‘load_models’ ініціалізує та повертає екземпляри кількох моделей розмітки частин мови, включаючи Stanza, SpaCy, Flair та модель RoBERTa з донавчанням, кешуючи результати, щоб уникнути повторних завантажень.

Функції ‘stanza_pos’, ‘spacy_pos’, ‘tag_ukrainian_text’, ‘tag_flair_text’ та ‘tag_roberta_text’ виконують розмітку частин мови на заданому тексті за допомогою відповідних моделей, повертаючи списки кортежів, що містять токени та їхні відповідні теги частин мови. Для моделей Flair та RoBERTa додаткова обробка забезпечує правильну токенізацію та форматування результату.

Функція ‘loop_until_completed’ багаторазово опитує статус виконання OpenAI Assistant до його завершення, невдачі або потреби в додаткових діях. Функція print_thread_messages отримує та виводить повідомлення з потоку OpenAI на консоль, відображаючи результати асистента.

В інтерфейсі Streamlit користувач вводить текст для аналізу, а результати розмітки частин мови від різних моделей захоплюються та обробляються функцією ‘capture_printed_output’. Ця функція захоплює друковані результати та структурує їх для порівняння. Функція ‘parse_output’ організовує результати розмітки у DataFrame, забезпечуючи послідовне порівняння токенів серед різних моделей шляхом вирівнювання токенів та їх відповідних тегів.

Функція ‘highlight_discrepancies’ застосовує схему підсвічування до DataFrame, щоб візуально вказати на розбіжності в тегах частин мови серед моделей, використовуючи жовті підсвічування на основі певних критеріїв згоди між моделями.

Управління станом Streamlit забезпечує відповідну реакцію інтерфейсу на взаємодії користувача, такі як відображення результатів розмітки частин мови або довідника, що пояснює різні теги частин мови.

Загалом, інструмент полегшує детальне порівняння та аналіз результатів розмітки частин мови, забезпечуючи підтримку кількох моделей та зручний інтерфейс для лінгвістичних досліджень та аналізу.

Фінальний продукт має вигляд веб-додатку з інтуїтивним інтерфейсом. у верхній частині вікна програми знаходиться поле для вводу тексту користувачем, нижче – кнопка «Почати» при натисканні якої запускається скрипт та виводиться результат. Після виведення результату стає доступною кнопка «Показати довдник з тегами», натискання на яку активує таблицю із тегами, що використовуються аналізованими моделями та їхніми значеннями українською мовою.

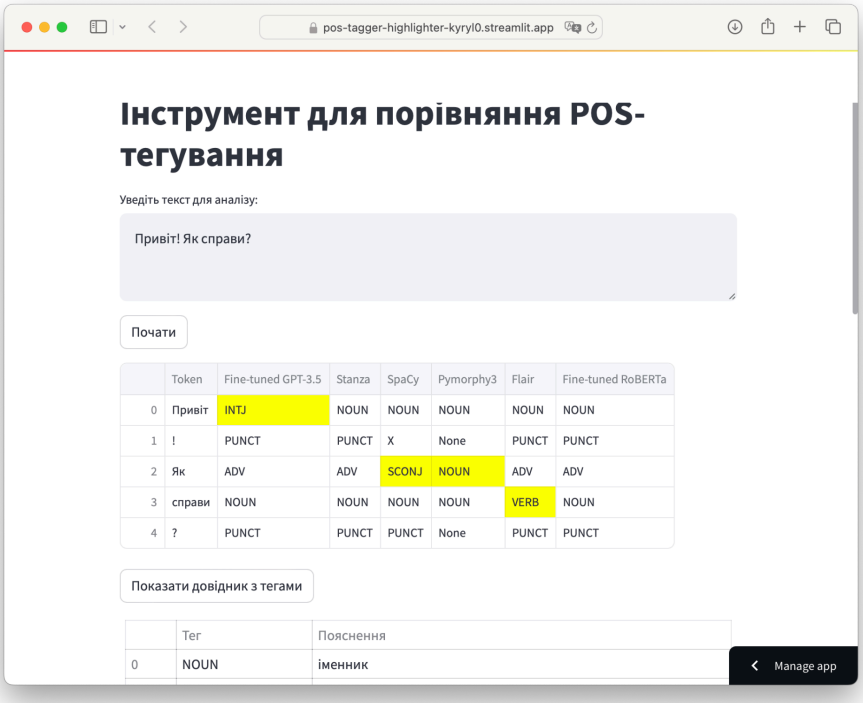


Рис. 5. – Інтерфейс додатку для порівняння частиномовного тегування.

Додаток було опубліковано в мережі Інтернет [33]. Він є єдиним способом поширення дотренованої моделі, оскільки доступ до неї надається лише за індивідуальним ключем OPENAI API, що був використаний для файн-тюнінгу моделі. Цей ключ є вразливою інформацією, яку не можна поширювати у відкритому вигляді. У коді

програми ключ було приховано за допомогою функції ‘streamlit.secrets’ [42].

3.4. Опис процедури порівняльного аналізу та оцінки

Усі зібрані тексти було оброблено за допомогою створеної програми-агрегатора. При обробці було використано системне повідомлення відмінне від того, що використовувалося під час файн-тюнінгу. До нього було додано речення «You will provide precise POS tags for all input text, and attempt to tag all words correctly, even those initially marked as 'X' in the dataset. Keep an eye out for these words and attempt to tag them appropriately based on context.», аби спонукати модель до тегування абсолютно всіх слів, навіть тих, які у тренувальному датасеті були теговані як 'X' (невизначено) при цьому звертаючи увагу на контекст.

Український переклад доданого речення наступний: «Ви надасте точні теги POS для всього вхідного тексту та спробуєте правильно позначити тегами всі слова, навіть ті, які спочатку позначені як «X» у наборі даних. Зверніть увагу на ці слова та спробуйте позначити їх належним чином відповідно до контексту.»

Отримані порівняльні таблиці було автоматично об’єднано в єдину таблицю. Таблицю було завантажено до хмарного сховища та поширено з дозволом для редагування.

Першим етапом аналізу була оцінка фактичної точності усіх інструментів POS-тегування на проблемних категоріях слів. До категорій, точність розмітки яких оцінювалася для усіх моделей, належать «Граматична омонімія» та «Орфографічні помилки/одруківки/ненормований регістр».

Оцінка категорій «Вкраплення іншомовних слів транскрибовані українською» та «Вкраплення іншомовних слів латинською/іншою абеткою» проводилася лише для дотренованої GPT-3.5, оскільки решта інструментів не були створені для виконання такого виду розмітки та їхня фактична точність для цієї категорії є майже випадковою (з виключенням калькованих слів іншомовного походження, які всі моделі здатні ідентифікувати завдяки українському морфологічному складу). Хоча score цього завдання також був обрахований для всіх моделей, він не є

показовим, а всі коректні тегування зроблені рештою моделей для цієї категорії слів можна вважати випадковістю.

Оцінка проводилася двома експертами з крос-перевіркою, тобто було отримано два варіанти розмітки від двох експертів та узгоджено відмінності.

Процедура оцінки виконання POS-тегування виконувалася за допомогою color-coding. Кожному з завдань було призначено два кольори для позначення коректного та некоректного тегування.

Назви категорій оцінювання та призначені кольори у випадках правдивого та неправдивого тегування відповідно:

- «Граматична омонімія»: зелений та червоний;
- «Орфографічні помилки (помилки орфографії, одруки навмисні та випадкові, ненормований регістр)»: блакитний та оранжевий;
- «Вкраплення іншомовних слів транскрибовані українською, включно зі сленгом»: рожевий та сірий;
- «Вкраплення іншомовних слів латинською/іншою абеткою»: жовтий та чорний;
- «Українськомовний сленг, обсценна лексика, жаргонізми»: темно-зелений та темно-фіолетовий.

Задля уникнення некоректних обрахунків, кольори було узгоджено експертами та створено макроси з прив’язкою до згаданих кольорів та шорткатів на клавіатурі.

Фінальна версія таблиці, що була розмічена експертами доступна в Додатку А проєкту.

Після експертної розмітки було створено код для обчислення загальної точності та точності виконання POS-розмітки на кожному з завдань. Код, написаний мовою програмування Python, знаходиться у github-репозиторії[5] проєкту та працює наступним чином:

Надається доступ до Google Sheets та Google Drive за допомогою даних сервісного облікового запису, зчитаних із JSON-файлу. Авторизація здійснюється через клієнт gspread. Далі задається адреса електронної таблиці за її ідентифікатором та обирається аркуш для обробки.

[5] https://github.com/kiryaa865/uni-thesis4

Для отримання метаданих електронного документа, включно з ідентифікатором аркуша, ініціалізується сервіс Google Sheets API. Після цього виконується запит для отримання форматування клітинок і значень, зокрема інформації про колір фону кожної клітинки.

Ініціалізується словник для зберігання кількості кольорів для кожного заголовка. Відбувається пошук позицій заголовків у першому рядку аркуша і додавання відповідних позицій до словника. Після цього перебираються всі рядки аркуша, окрім першого, аби підрахувати кількість кольорів у кожному рядку для відповідних заголовків. Колір кожної клітинки визначається за значеннями червоного, зеленого та синього (RGB) компонентів, які зберігаються в словнику.

Після підрахунку кольорів для кожного заголовка обраховуються відсоткові співвідношення для конкретних кольорових комбінацій. Функція ’calculate_percentage’ обчислює відсоток одного кольору від суми двох кольорів, якщо знаменник не дорівнює нулю.

Для кожного заголовка виконується розрахунок комбінацій кольорів, використаних для експертного маркування. Після цього відображаються результати у відсотковому вигляді для кожного заголовка, зокрема точність виконання POS-тегування кожної з моделей на проблемних випадках лінгвістичної омонімії, англо-українських комбінацій, одруківок та капіталізації, англійських слів, а також нецензурної лексики та сленгу. У Таблиці 2 наведені обраховані значення точності для згаданих випадків.

Категорія	GPT-3.5	Stanza	SpaCy	Pymorphy3	Flair	RoBERTa
Граматична омонімія	70.59%	49.25%	55.22%	48.48%	62.12%	73.13%
Вкраплення англійською (транскрибовані)	84.49%	19.05%	18.36%	19.63%	19.82%	13.87%
Одруківки, нестандартний регістр	89.14%	58.40%	48.43%	44.86%	64.86%	68.29%
Вкраплення англійською	97.62%	0.48%	0.72%	0.24%	0.24%	0.97%

Нецензурна лексика, сленг	93.39%	53.72%	49.59%	39.67%	43.80%	50.41%

Табл. 2. – Результати точності частиномовного тегування моделей на проблемних випадках

3.5. Порівняльний аналіз проблеми одруків, орфографічних помилок та нестандартного регістру

Актуальність дослідження цих проблем полягає у тому, що текстові дані у реальному світі рідко бувають повністю правильними з орфографічної точки зору. Наприклад, у соціальних мережах, чатах та електронних листах користувачі часто допускають помилки або свідомо використовують нестандартні форми написання слів. Це створює додаткові виклики для моделей POS-тегування, оскільки вони повинні бути здатні коректно ідентифікувати та позначати частини мови навіть у випадках, коли текст містить помилки або нестандартизоване використання регістру.

Результати порівняння точності моделей на цьому завданні, отримані за допомогою скрипта, наступні:

Інструмент	Точність, %
Fine-Tuned GPT-3.5	89.14
Fine-Tuned RoBERTa	68.29
Flair	64.86
Stanza	58.40
SpaCy	48.43
Pymorphy3	44.86

Табл. 3. – Точність виконання POS-тегування для одруків, орфографічних помилок та нестандартного регістру

Помітна кореляція зниження точності тегування зі зниженням складності архітектури інструменту, використаного для POS-тегування.

Великі мовні моделі з трансформерною архітектурою GPT-3.5 та RoBERTa показали найвищу точність, а Pymorphy, що працює за допомогою лінгвістичних правил – найнижчу.

Усі випадки одруків у тестувальному файлі можна поділити на зі стандартним та з нестандартним регістром.

Нижче наведений фрагмент одного з текстів, що аналізувався разом із розміткою експертів. Цей фрагмент можна віднести до випадків одруків зі стандартним регістром. Текст у повному обсязі з експертною розміткою доступний у Додатку А.

Token	Fine-tuned GPT-3.5	Stanza	SpaCy	Pymorphy 3	Flair	Fine-tuned RoBERTa
пхати	VERB	VERB	VERB	VERB	VERB	VERB
у	ADP	ADP	ADP	PREP	ADP	ADP
сеьн	PRON	NOUN	NOUN	NOUN	NOUN	NOUN
щось	PRON	PRON	PRON	NPRO	PRON	PRON
знаючи	VERB	VERB	VERB	GRND	VERB	VERB

Табл. 4. – Фрагмент маркованої порівняльної таблиці №1

У тестувальному тексті було допущено одрук у слові «себе» у реченні «... пхати у сеьн щось знаючи» (із контексту: «...пхати у себе щось, знаючи…»). Очевидно, що помилку було зроблено через високу швидкість набору тексту. На українській розкладці клавіатури букви «б» та «ь» й «н» і «е» знаходяться поряд, що стало причиною одруку.

Із усього списку засобів частиномовного тегування, коректний тег було надано лише дотренованою моделлю GPT-3.5. Можна припустити, що модель використала як контекстні підказки, так і знання про найбільш частотні одруки в українському письмовому мовленні для прогнозування тега завдяки претренованим вагам.

Решта інструментів видали некоректний тег «NOUN» (іменник). Можна припустити, що таке передбачення тега було зроблене через нульову флексію та закінчення слова на -н, що в українській мові властиво лише іменникам.

Нижче наведений фрагмент одного з текстів, що аналізувався, разом із розміткою експертів. У ньому наявні випадки одруків як зі стандартним так і з нестандартним регістром.

Token	Fine-tuned GPT-3.5	Stanza	SpaCy	Pymorphy 3	Flair	Fine-tuned RoBERTa
я	PRON	PRON	PRON	NPRO	PRON	PRON
аьсолютно	ADV	ADV	ADV	ADVB	ADV	ADV
НЕНАВТДЖУУУУ	VERB	VERB	PROPN	NOUN	VERB	VERB

Табл. 5. – Фрагмент маркованої порівняльної таблиці №2

У наведеному вище фрагменті наявне слово «аьсолютно», яке є помилковим варіантом написання коректної форми «абсолютно» та виникло через одрук. Цей випадок не став проблемою для жодної з моделей, оскільки слово було збережено майже у його оригінальному вигляді, з «неушкодженою» флексією, яка дала змогу коректно передбачити частиномовний тег.

Щодо «НЕНАВТДЖУУУУ», воно є ненормативною формою слова «ненавиджу» з наявним одруком («Т» замість «И») та написане повністю у верхньому регістрі. Сегмент «УУУУ» є гемінацією (гіперболічним повтором). Незважаючи на гемінацію та наявність одруку в корені слова, усі моделі, окрім Stanza та Pymorphy, змогли коректно визначити частиномовний тег. Помилкове передбачення тега «NOUN» (іменник) замість «VERB» (дієслово) бібліотекою Pymorphy, викликане її архітектурою, що заснована на формальних лінгвістичних правилах. Слово не знайшлося в словнику, а придатного правила не існувало. Така поведінка Pymorphy є типовою та частотною при тегуванні OOV-слів за статистичними даними таблиці, що доступна в Додатку А.

Передбачення тега «PROPN» (власне ім’я) замість «VERB» (дієслово) POS-тегером SpaCy обумовлене верхнім регістром слова та є частотною помилкою цього інструменту, виходячи з даних порівняльної таблиці, що доступна в Додатку А.

Ще однією категорією орфографічних помилок, які були наявні в тестувальному файлі, є навмисна чи ненавмисна асиміляція, подібна до фонетичного представлення слова. Такі випадки траплялися як самостійно так і в комбінації з використанням верхнього регістру, що становить окрему проблему для деяких з інструментів, що порівнювалися.

Нижче наведена частина одного з аналізованих речень, у якій присутнє слово з фонетичною асиміляцією, вираженою на орфографічному рівні.

Token	Fine-tuned GPT-3.5	Stanza	SpaCy	Pymorphy3	Flair	Fine-tuned RoBERTa
я	PRON	PRON	PRON	NPRO	PRON	PRON
хочю	VERB	VERB	ADV	NOUN	VERB	VERB
навчитися	VERB	VERB	VERB	VERB	VERB	VERB
малювать	VERB	VERB	VERB	VERB	VERB	VERB

Табл. 6. – Фрагмент маркованої порівняльної таблиці №3

Слово «хочю» є ненормативним орфографічним варіантом слова «хочу». При аналізі SpaCy та Pymorphy3 допустилися неправильного тегування.

Бібліотека Pymorphy протегувала слово як «NOUN» (іменник), що є властивою для Pymorphy помилкою при частиномовній розмітці слів, що не входять до її словника.

SpaCy припустилася помилки, протегувавши це слово як «ADV» (прислівник). Можна припустити, що неправильне передбачення частиномовного тега трапилося через закінчення слова на «-чю», яке в українській мові притаманне лише обмеженій категорії застарілих прислівників, таких як «ніччю».

Нижче наведена частина одного з аналізованих речень, у якій присутнє слово з фонетичною асиміляцією, вираженою на орфографічному рівні та з використанням верхнього регістру, що також становить проблему для деяких з інструментів, що порівнювалися.

Token	Fine-tuned GPT-3.5	Stanza	SpaCy	Pymorphy3	Flair	Fine-tuned RoBERTa
він	PRON	PRON	PRON	NPRO	PRON	PRON
реально	ADV	ADV	ADV	ADVB	ADV	ADV
просто	ADV	PART	PART	ADVB	PART	PART
ВІДЧЮВАВ	VERB	ADV	PROPN	NOUN	VERB	VERB
цей	DET	DET	DET	NPRO	DET	DET
вайб	NOUN	NOUN	NOUN	NOUN	NOUN	NOUN

Табл. 7. – Фрагмент маркованої порівняльної таблиці №4

У випадку слова «ВІДЧЮВАВ», яке є ненормовою формою «відчував» у верхньому регістрі, 3 з 6 інструментів, що порівнювалися, змогли передбачити правильний частиномовний тег. Це вдалося моделям GPT-3.5, Flair та RoBERTa.

Stanza помилково передбачила тег «ADV» (прислівник) замість VERB (дієслово), що не можна пояснити морфологією цього слова. Можливо, помилка була викликана верхнім регістром.

Передбачення тега «PROPN» (власне ім’я) замість «VERB» (дієслово) POS-тегером SpaCy обумовлене верхнім регістром слова та є частотною помилкою цього інструменту, виходячи з даних порівняльної таблиці.

Помилкове тегування «NOUN» (іменник) замість «VERB» (дієслово) бібліотекою Pymorphy викликане її архітектурою, що базується на формальних лінгвістичних правилах. Слово не було знайдено в словнику, а відповідне правило не існувало. Така поведінка Pymorphy є типовою та частотною при тегуванні OOV-слів, що підтверджується статистичними даними таблиці, що доступна в Додатку А.

У таблиці, що знаходиться нижче, наведено фрагмент речення із порівняльної таблиці, повна версія якої доступна в Додатку А.

Token	Fine-tuned GPT-3.5	Stanza	SpaCy	Pymorphy3	Flair	Fine-tuned RoBERTa

нема	VERB	VERB	VERB	NOUN	VERB	VERB
у	ADP	ADP	ADP	PREP	ADP	ADP
світі	NOUN	NOUN	NOUN	NOUN	NOUN	NOUN
речей	NOUN	NOUN	NOUN	NOUN	NOUN	NOUN
які	PRON	DET	DET	NPRO	DET	DET
я	PRON	PRON	PRON	NPRO	PRON	PRON
люблю	VERB	VERB	VERB	VERB	VERB	VERB
малювати	VERB	VERB	VERB	VERB	VERB	VERB
біше	ADV	ADV	ADV	ADVB	ADV	ADV
ніж	SCONJ	SCONJ	SCONJ	CONJ	SCONJ	SCONJ

Табл. 8. – Фрагмент маркованої порівняльної таблиці №5

У цьому прикладі присутнє слово «біше», яке є орфографічно зредукованою формою слова «більше». Незважаючи на спрощення в основі словоформи, всі моделі, що порівнювалися, правильно позначили це слово тегом «ADV» (прислівник). Це відбулося завдяки «неушкодженому» закінченню « -ше» притаманному українським прислівникам.

3.6. Порівняльний аналіз проблеми чергування мов

Проблема чергування мов (код-світчингу), є значущою темою досліджень у сфері обробки природної мови, зокрема у завданнях частиномовного тегування. Код-світчинг відбувається, коли у межах одного висловлювання чи тексту використовуються слова або фрази з різних мов. Це явище є досить поширеним у багатомовних суспільствах та онлайн-спільнотах, де користувачі часто переходять з однієї мови на іншу. Така мовна практика створює додаткові виклики для алгоритмів автоматичної обробки тексту, оскільки моделі повинні коректно ідентифікувати та позначати частини мови у контекстах, де змішуються різні мовні системи.

Актуальність дослідження цієї проблеми підкреслюється зростанням кількості мультимовних ресурсів і даних у цифровому середовищі.

Зокрема, в сучасному українському мовному дискурсі українська та англійська мови часто використовуються одночасно та формують калькування й суржик [1]. Важливо мати ефективні інструменти для обробки таких текстів.

Результати точності на цьому завданні, отримані за допомогою обрахунку коректних та некоректних тегувань наступні:

Інструмент	Точність тегування у випадках мовного чергування з використанням транскрибування	Точність тегування у випадках мовного чергування без використання транскрибування
GPT-3.5	84.49%	97.62%
Stanza	19.05%	0.48%
SpaCy	18.36%	0.72%
Pymorphy3	19.63%	0.24%
Flair	19.71%	0.24%
RoBERTa	13.87%	0.97%

Табл. 9. – Точність виконання POS-тегування для випадків мовного чергування з та без використання транскрибування

Надані оцінки для всіх інструментів, окрім дотренованої GPT-3.5 слугують виключно бейзлайном оцінки якості виконаного експериментального дотренування GPT-3.5 для тегування випадків такого типу.

Під час виконання розмітки та оцінки всі випадки було згруповано до цих двох категорій: «З використанням транскрибування» та «Без використання транскрибування».

Ці групи в свою чергу можна умовно поділити на дві категорії: з порушенням синтаксичної структури властивої українській та без її порушення. Збереження української синтаксичної структури в текстах з код-світчингом відбувається шляхом збереження українських службових синтаксичних конструкцій і використання поодиноких іншомовних вставок, що є іншомовними відповідниками відвічно українських слів чи конструкцій.

Ще однією підкатегорією є поодинокі випадки словотвірної кальки, створених шляхом адаптації іншомовного слова до української морфології [5] (додавання українських афіксів) напр. «вона сервить» від англ. «she's serving». У цьому прикладі було додано українську флексію, що властива дієсловам другої особи однини теперішнього часу, яка стала «перекладом» англійського суфікса «-ing» [39].

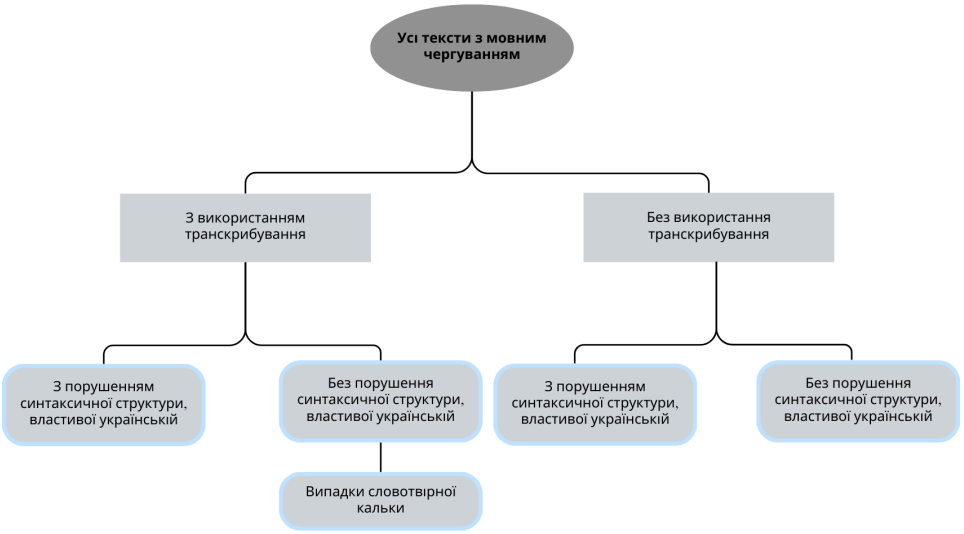


Рис. 6. – Блок-схема категорій випадків міжмовного чергування

Прикладом категорії «Зміна мови посеред українськомовного тексту без застосування транскрибування українською з порушенням синтаксичної структури, що властива українській мові», є наступний фрагмент тестувальної таблиці:

Token	GPT-3.5	Stanza	SpaCy	Pymorphy3	Flair	RoBERTa
речі	NOUN	NOUN	NOUN	NOUN	NOUN	NOUN
які	PRON	DET	DET	NPRO	DET	DET
я	PRON	PRON	PRON	NPRO	PRON	PRON
б	PART	AUX	AUX	PRCL	AUX	AUX
віддала	VERB	VERB	VERB	VERB	VERB	VERB
аби	SCONJ	SCONJ	SCONJ	CONJ	SCONJ	SCONJ
бути	VERB	VERB	AUX	NOUN	VERB	AUX

less	ADV					
oblivious	ADJ					
and	CCONJ		CCONJ			CCONJ
obvious	ADJ					

Табл. 10. – Фрагмент маркованої порівняльної таблиці №6

Речення «Речі, які я б віддала, аби бути less oblivious and obvious.» є калькою сталої англійської конструкції «The things I would give up for/to…». Синтаксис української частини цього речення хоч і є коректним, не властивий українській мові. Відвічно українським відповідником, який правильно передає значення може бути речення «Я би віддала все, щоб стати менш непомітною та очевидною».

Поведінка моделей при тегуванні цього типу код-світчингу є щоразу майже однаковою: іншомовні слова (у цьому випадку англійські) тегуються дотренованою GPT-3.5 із високою точністю, а решта моделей повертають або тег «X» (невідомо) або None.

Щоправда, цікавою приміткою є те, що моделі SpaCy та RoBERTa щоразу правильно призначають тег для сполучника «and» при цьому не передбачаючи жодних тегів для решти англійських слів. Таку поведінку можна пояснити тим, що при адаптації цих моделей для української залишилися деякі ваги найчастотніших англійських слів, або ж цей артефакт виникнув з українських тренувальних даних.

Ще одним прикладом вартим уваги є наступне речення:

Token	GPT-3.5	Stanza	SpaCy	Pymorphy3	Flair	RoBERTa
ні	PART	CCONJ	PART	PRCL	CCONJ	PART
ну	PART	PART	PART	PRCL	PART	PART
не	PART	PART	PART	PRCL	PART	PART
знаю	VERB	VERB	VERB	VERB	VERB	VERB
чят	NOUN	X	NOUN		NOUN	NOUN
it	PRON					
gets	VERB					

better	ADJ					
ше	PART	PART	ADV	PRCL	PART	PART
можна	VERB	ADV	ADV	ADJF	ADV	ADV
жить	VERB	VERB	VERB	VERB	VERB	VERB

Табл. 11. – Фрагмент маркованої порівняльної таблиці №7

У цьому випадку англійськомовне вкраплення було інкорпороване в середині речення, а не в кінці. Також присутні інші письмові особливості, як от: навмисна орфографічна помилка у слові «чят» замість «чат», яку не змогли обробити Stanza та Pymorphy та фонетичне спрощення, виражене на письмі «ше» замість «ще», яке коректно протегували всі моделі.

Прикладом категорії «Зміна мови посеред українськомовного тексту з застосуванням транскрибування українською з порушенням синтаксичної структури, що властива українській мові», є наступний фрагмент тестувальної таблиці:

Token	**GPT-3.5**	**Stanza**	**SpaCy**	**Pymorphy3**	**Flair**	**RoBERTa**
усе	PRON	PRON	PRON	PRCL	PRON	PRON
сиплеться	VERB	VERB	VERB	VERB	VERB	VERB
на	ADP	ADP	ADP	PRCL	ADP	ADP
очах	NOUN	NOUN	NOUN	NOUN	NOUN	NOUN
айм	AUX	NOUN	NOUN		NOUN	PROPN
рілі	ADV	NOUN	ADJ	NOUN	ADJ	PROPN
нот	PART	NOUN	NOUN	NOUN	NOUN	NOUN
гона	AUX	NOUN	PROPN	NOUN	NOUN	PROPN
мейк	VERB	NOUN	NOUN	NOUN	NOUN	PROPN
іт	PRON	X	X	NOUN	PART	PROPN
зіс	DET	NOUN	ADP		VERB	X
тайм	NOUN	NOUN	NOUN	NOUN	NOUN	NOUN
чят	INTJ	NOUN	NOUN		NOUN	VERB

Табл. 12. – Фрагмент маркованої порівняльної таблиці №8

Частина «айм рілі нот гона мейк іт зіс тайм» є транскрибованою версією «I’m really not gonna make it this time.», питомим українським відповідником якої може бути речення «Я справді не встигну цього разу.» або ж «Я справді не переживу це цього разу.». З таблиці можна помітити, що дотренована GPT-3.5 змогла правильно передбачити всі теги, незважаючи на використання транскрибованих англійських слів.

Закономірним є тегування транскрибованих англійських слів тегами «NOUN» та «PROPN» (іменник та власне ім’я) усіма інструментами, відмінними від GPT-3.5, що в цьому прикладі обумовлене відсутністю афіксів, які би вказували на належність цих слів до інших граматичних категорій.

Тегування «ADJ» (прикметник) слова «рілі» (анг. really) моделлю Flair швидше за все пов’язане з комбінацією «-лі» в кінці слова, що є властивою для українських прикметників, як от «білі».

Також необхідно зазначити, що слово «чят» (чат) у контексті цього речення дотренована GPT-3.5 ідентифікувала як «INTJ» (вигук), що не є помилковим, як і тегування «NOUN» (іменник) рештою моделей, хоч і цілком може бути випадковим. З точки зору сучасної інтернет-комунікації в англомовному сегменті, «chat» цілком можна віднести до вигуків. В інтернет-стрімингу «chat» змістило своє значення з прямого, що означало «чат» спершу до значення колективного звернення стрімера (людини, що проводить інтернет-трансляцію) до своїх глядачів, а з часом «chat» почало вживатися «через слово» навіть не публічними людьми у значенні схожому на вигук, тобто вираження емоцій або ж у значенні звернення ні до кого зокрема (‘chat’ is generically being used to refer to nobody in particular) [44]. Щодо нового значення слова «chat» в контексті інтернет-комунікації наразі ведуться дискусії.

Тема англомовної інтернет-комунікації є широко досліджуваною. На момент написання, Google Scholar [19] знаходить 9180 результатів за запитом «Twitch slang», тому, можливо, це значення слова «chat» невдовзі стане дослідженим.

Прикладом категорії «Зміна мови з української на іноземну на основі транскрибування зі збереженням питомої синтаксичної структури» є наступний сегмент речення, наведений в порівняльній таблиці нижче.

Token	GPT-3.5	Stanza	SpaCy	Pymorphy3	Flair	RoBERTa
дуже	ADV	ADV	ADV	ADVB	ADV	ADV
багато	DET	DET	DET	ADVB	DET	DET
оюдей	NOUN	NOUN	NOUN	NOUN	NOUN	NOUN
есюмд	VERB	NOUN	PROPN	NOUN	VERB	NOUN
шо	SCONJ	SCONJ	SCONJ	NPRO	SCONJ	SCONJ
я	PRON	PRON	PRON	NPRO	PRON	PRON
аутістік	ADJ	NOUN	VERB	NOUN	NOUN	NOUN
…	…	…	…	…	…	…
ніде	ADV	ADV	ADV	ADVB	ADV	ADV
не	PART	PART	PART	PRCL	PART	PART
стейтед	VERB	NOUN	NOUN	NOUN	VERB	NOUN
шо	SCONJ	SCONJ	SCONJ	NPRO	SCONJ	SCONJ
я	PRON	PRON	PRON	NPRO	PRON	PRON
аутістік	ADJ	NOUN	VERB	NOUN	NOUN	NOUN

Табл. 13. – Фрагмент маркованої порівняльної таблиці №9

У Таблиці 13 наведене речення зі зміною мови на англійську з застосуванням транскрибування. Зміна мови відбувається для поодиноких слів, при цьому зберігається питома українська синтаксична структура речення. Були вжиті слова «есюмд» (англ. – assumed, укр. – припустити), «аутістік» (англ. – autistic, укр. – аутист) та «стейтед» (англ. – stated, укр. – зазначала). Коректно позначила ці слова дотренована GPT-3.5 та Flair, яка змогла ідентифікувати транскрибовані англійські дієслова.

Прикладом категорії «поодинокі випадки словотвірної кальки, створених шляхом адаптації іншомовного слова до української морфології (додавання українських афіксів)» є наступний фрагмент:

Token	GPT-3.5	Stanza	SpaCy	Pymorphy3	Flair	RoBERTa
було	VERB	AUX	AUX	VERB	AUX	AUX
б	PART	AUX	AUX	PRCL	AUX	AUX

краще	ADV	ADV	ADV	ADVB	ADV	ADV
якби	SCONJ	SCONJ	SCONJ	CONJ	SCONJ	SCONJ
я	PRON	PRON	PRON	NPRO	PRON	PRON
таки	PART	PART	PART	PRCL	PART	PART
міскеріейжнулася	VERB	VERB	VERB	VERB	VERB	VERB

Табл. 14. – Фрагмент маркованої порівняльної таблиці №10

Слово «міскеріейджнулася» є калькою з англійського «miscarriage» (викидень), що зазнало словотвірної модифікації. В англійській мові значення пасивної дії першої особи передається аналітично, а при калькуванні в українську для маркування цього граматичного значення було додано конструкцію «-ласям». Завдяки питомо українській флексії всі моделі успішно розпізнали частину мови як «VERB» (дієслово).

3.7. Порівняльний аналіз проблеми граматичної омонімії

Граматична омонімія є важливою темою у сфері лінгвістики та обробки природної мови, яка стосується явища, коли одне й те саме слово має різні граматичні значення або виконує різні синтаксичні функції в залежності від контексту. Це створює значні виклики для автоматичних систем обробки тексту, зокрема для завдань частиномовного (POS) тегування, оскільки правильно визначити частину мови омонімічного слова можна лише з урахуванням його контексту.

Граматична омонімія може проявлятися в різних формах, таких як омонімія між різними частинами мови (наприклад, слово «біг» може бути іменником у значенні «дія бігти», а також дієсловом у минулому часі від «бігти»), а також омонімія між різними граматичними формами одного і того ж слова (наприклад, форма «лісу» може бути родовим відмінком іменника «ліс» або давальним відмінком іменника «ліс»).

Вивчення граматичної омонімії та її врахування у розробці моделей для POS-тегування є важливим кроком для підвищення точності систем автоматичної обробки тексту.

Результати тестування точності розмітки для цього завдання наступні:

Інструмент	Точність розмітки випадків граматичної омонімії
RoBERTa	69.57%
GPT-3.5	67.14%
Flair	57.97%
SpaCy	55.07%
Stanza	49.28%
Pymorphy3	48.53%

Табл. 15. – Точність виконання POS-тегування для випадків граматичної омонімії

З результатів помітно, що найвищу точність тегування мають LLM RoBERTa та GPT-3.5. Результати хоч і вищі за решту моделей, проте не задовільні та вказують на перспективу напрямку майбутніх досліджень.

Нижче наведений фрагмент із порівняльної таблиці з присутньою проблемою граматичної омонімії.

Token	GPT-3.5	Stanza	SpaCy	Pymorphy3	Flair	RoBERTa
гуляли	VERB	VERB	VERB	VERB	VERB	VERB
кругом	ADP	NOUN	NOUN	ADVB	ADP	ADP
озера	NOUN	NOUN	NOUN	NOUN	NOUN	NOUN
.	PUNCT	PUNCT	PUNCT	None	PUNCT	PUNCT
Випадком	NOUN	NOUN	NOUN	ADVB	NOUN	ADV
там	ADV	ADV	ADV	PRCL	ADV	ADV
опинилися	VERB	VERB	VERB	VERB	VERB	VERB

Табл. 16. – Фрагмент маркованої порівняльної таблиці №11

Слово «кругом», яке в цьому контексті є прийменником, моделі Stanza та Spacy тегували як іменник, що є помилковим. Pymorphy

призначив слову тег «ADVB» (прийменник), що також не є коректним. Токен «Випадком» коректно позначили лише Pymorphy та RoBERTa.

У наступній таблиці наведений фрагмент тестувальної таблиці, у якій слово використане двічі з різними граматичними значеннями.

Token	GPT-3.5	Stanza	SpaCy	Pymorphy3	Flair	RoBERTa
коло	NOUN	ADP	ADP	NOUN	ADP	ADP
мене	PRON	PRON	PRON	NOUN	PRON	PRON
стоїть	VERB	VERB	VERB	VERB	VERB	VERB
Вчитель	NOUN	NOUN	NOUN	NOUN	NOUN	NOUN
.	PUNCT	PUNCT	PUNCT	None	PUNCT	PUNCT
Вчитель	NOUN	NOUN	NOUN	NOUN	NOUN	NOUN
намалював	VERB	VERB	VERB	VERB	VERB	VERB
коло	NOUN	ADP	ADP	NOUN	ADP	ADP
на	ADP	ADP	ADP	PRCL	ADP	ADP
дошці	NOUN	NOUN	NOUN	NOUN	NOUN	NOUN
.	PUNCT	PUNCT	PUNCT	None	PUNCT	PUNCT

Табл. 17. – Фрагмент маркованої порівняльної таблиці №12

Слово «коло» у першому випадку з контексту має значення прийменника, в другому – іменника. В обидвох випадках моделі передбачили однакові теги. Отже, жодна з моделей не змогла розпізнати, що слово було вжито в різних граматичних значеннях.

3.8. Порівняльний аналіз проблеми обсценної лексики, жаргонізмів та сленгу

Проблема обсценної лексики, жаргонізмів та сленгу є значущою у завданнях частиномовного тегування. Обсценна лексика включає слова та вирази, що вважаються непристойними або образливими. Жаргонізми — це спеціалізовані терміни, які використовуються у вузьких професійних або соціальних групах, а сленг — неформальні вирази, що широко

вживаються в розмовній мові. Кожна з цих категорій представляє виклики для автоматичних систем обробки тексту.

Обсценна лексика часто не включається у стандартні мовні корпуси та словники, що ускладнює її розпізнавання та правильне тегування моделями. Жаргонізми, через свою спеціалізованість та швидку еволюцію, можуть бути маловідомими або незрозумілими для загальновживаних мовних моделей. Сленг, зважаючи на свою неформальність та варіативність, також часто виходить за межі стандартних лінгвістичних норм і правил.

Актуальність дослідження цих проблем обумовлена тим, що сучасні комунікаційні платформи, соціальні мережі та інтернет-форуми часто містять велику кількість обсценної лексики, жаргонізмів та сленгу. Для забезпечення точності обробки таких текстів, моделі частиномовного тегування повинні бути адаптовані до розпізнавання та правильного тегування цих мовних явищ.

Інструмент	**Точність тегування випадків обсценної лексики, жаргонізмів та сленгу**
GPT-3.5	93.39%
Stanza	53.72%
RoBERTa	50.41%
SpaCy	49.59%
Flair	43.80%
Pymorphy3	39.67%

Табл. 18. – Точність виконання POS-тегування для випадків обсценної лексики, жаргонізмів та сленгу

З таблиці помітно, що дотренована GPT-3.5 показала найвищу (93.39%) точність зі значним відривом на цьому завданні. На другому та третьому місцях – Stanza та RoBERTa з точностями 53.72% та 50.41% відповідно.

У Таблиці 19 наведений фрагмент порівняльної таблиці, повна версія якої доступна в Додатку А. У фрагменті наведене обсценне слово «бидло».

Token	GPT-3.5	Stanza	SpaCy	Pymorphy3	Flair	RoBERTa
бидло	NOUN	VERB	NOUN	NOUN	VERB	NOUN

Табл. 19. – Фрагмент маркованої порівняльної таблиці №14

При тегуванні моделі Stanza та Flair припустилися помилки, передбачивши тег «VERB» (дієслово) замість «NOUN» (іменник). Це було спричинено закінченням «-ло», яке є характерним дієсловам середнього роду минулого часу.

Token	GPT-3.5	Stanza	SpaCy	Pymorphy3	Flair	RoBERTa
(ред.) н*ху*	ADV	NOUN	PROPN	NOUN	NOUN	NOUN
(ред.) б*ят*	INTJ	VERB	VERB	NOUN	X	X

Табл. 20. – Фрагмент маркованої порівняльної таблиці №15

У Таблиці 20, що наведена вище, наведені результати тегування двох обсценних слів. Перше слово правильно протегувала лише дотренована GPT-3.5, що передбачила тег «ADV» (прислівник), друге — GPT-3.5 та Pymorphy, які тегували слово тегами «INTJ» (вигук) та «NOUN» (іменник). Решта моделей некоректно передбачили теги «NOUN» (іменник) у першому випадку та «VERB» (дієслово) в другому.

До категорії також увійшли випадки жаргонізмів. У таблиці 21, яка є фрагментом порівняльної таблиці, повна версія якої доступна в Додатку А, наведений приклад тегування слів інтернет жаргону.

Token	GPT-3.5	Stanza	SpaCy	Pymorphy3	Flair	RoBERTa
гігі	INTJ	NOUN	ADJ	ADJF	NOUN	PROPN
сподоб	VERB	NOUN	ADJ	NOUN	NOUN	PROPN
опше	ADV	ADV	VERB	NOUN	ADV	ADV

Табл. 21. – Фрагмент маркованої порівняльної таблиці №16

Слово «гігі» змогла коректно ідентифікувати лише дотренована GPT-3.5, що позначила його як «INTJ» (вигук). Це слово є письмовим виявленням сміху. Решта моделей помилилися, передбачивши теги «NOUN» (іменник), «ADJ» (прикметник) та «PROPN» (власне ім’я).

Слово «сподоб», яке є скороченням від «сподобався», правильно тегувала лише дотренована GPT-3.5, яка позначила його як «VERB» (дієслово). Решта моделей помилково передбачили теги «NOUN» (іменник) та «ADJ» (прикметник).

Слово «опше» яке є варіантом слова «взагалі» коректно ідентифікували всі моделі, окрім SpaCy та Pymorphy3.

У висновку, дотренована модель GPT-3.5 показала найвищі результати точності тегування обсценної лексики та різних видів жаргону незважаючи на те, що таких випадків не було в датасеті, що використовувався для дотренування.

Висновки до Розділу 3

У висновку, дотренована модель GPT-3.5 показала найкращі результати на 4 з 5 категорій, включно з завданням тегування транскрибованих англомовних вкраплень в українському тексті, яке є мультирівневим і потребує певної здібності до дедукції. Тестування дотренованої моделі на завданні граматичної омонімії хоч і дало порівняно високі результати точності (2 місце після LLM RoBERTa), вони все ще є не надто задовільними та є напрямком майбутніх досліджень. Посередня якість виконання тегування моделлю слів з граматичною омонімією є неочікуваною, враховуючи складність решти завдань, які вона виконала з високою точністю. Виходячи з досвіду дотренування моделі для тегування транскрибованих англомовних вкраплень, поведінку можна покращити, додавши порівняно невелику вибірку текстів, спрямованих на розуміння проблеми граматичної омонімії в українській мові до тренувального датасету.

ВИСНОВКИ

У ході цього дослідження була розроблена модель для автоматичного тегування частин мови в українському тексті на основі трансформера GPT-3.5. З метою покращення моделі було здійснено її файн-тюнінг на основі частини датасету Universal Dependencies для української мови, а також доданих тексти з українсько-англійською мовною зміною. Це забезпечило кращу адаптацію моделі до специфічних особливостей сучасного українського мовлення, зокрема до розпізнавання слів, написаних змішаною мовою, зі сленгом, орфографічними помилками та транскрибованими іншомовними словами.

Порівняльний аналіз показав, що дотренована модель GPT-3.5 демонструє високі результати точності в завданнях тегування обсценної лексики, різних видів жаргону та транскрибованих англомовних вкраплень в українському тексті, перевершуючи інші моделі, такі як Flair, RoBERTa, Stanza та SpaCy. Зокрема, у завданні тегування слів з граматичною омонімією, де результативність моделі все ще потребує покращення, GPT-3.5 показала досить високий результат, зайнявши друге місце після RoBERTa. Це свідчить про необхідність подальшого дослідження та вдосконалення моделі у цьому напрямку, зокрема шляхом додавання до тренувального датасету текстів, що відображають різноманітні випадки граматичної омонімії в українській мові.

Загалом, результати дослідження підтверджують ефективність використання великих мовних моделей для завдань POS-тегування, особливо в умовах змішаного мовлення та текстів розмовного мовлення. Висока точність тегування, продемонстрована дотренованою моделлю GPT-3.5, свідчить про її потенціал як потужного інструменту для лінгвістичного аналізу та подальшого розвитку сфери обробки української мови.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Григоренко І. В. Українсько-англійський суржик у новітньому комунікативному дискурсі / І. В. Григоренко // НПУ імені М. П. Драгоманова.
2. Коробов, М. Morphological Analyzer and Generator for Russian and Ukrainian Languages [Електронний ресурс] / М. Коробов // arXiv.org. – Режим доступу: https://arxiv.org/pdf/1503.07283. – Назва з екрану.
3. К. Кучинський, «Дотренування GPT-3 для виконання лінгвістичних завдань української мови», Курсова робота, Київський нацональний університет імені Тараса Шевченка, Київ, 2023.
4. Тараненко О. О. Колоквіалізація, субстандартизація та вульгаризація як характерні явища стилістики сучасної української мови (з кінця 1980-х рр.). Мовознавство. 2002. № 4–5. С. 33-39.
5. Власенко, В. В. Калькування в українській мові: стан і статус / В. В. Власенко // Національна бібліотека України імені В. І. Вернадського. – 2002. – Режим доступу: http://dspace.nbuv.gov.ua/handle/123456789/75569 (дата звернення: 28.05.2024).
6. Adelani, D. I., Doğruöz, A. S., Coneglian, A., Ojha, A. K. Comparing LLM prompting with Cross-lingual transfer performance on Indigenous and Low-resource Brazilian Languages [Електронний ресурс] / D. I. Adelani, A. S. Doğruöz, A. Coneglian, A. K. Ojha // arXiv.org. – 2024. – Режим доступу: https://arxiv.org/abs/2309.11674 (дата звернення: 28.05.2024).
7. Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., Vollgraf, R. FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP [Електронний ресурс] / A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, R. Vollgraf // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations). – 2019. – Режим доступу: https://aclanthology.org/N19-4010.pdf (дата звернення: 28.05.2024).

8. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z. DBpedia: A Nucleus for a Web of Open Data [Електронний ресурс] / S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives // Wiley Interdisciplinary Reviews: Computational Statistics. – Режим доступу: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.195(дата звернення: 28.05.2024).
9. Bose, K., Sarkar, K. Bengali POS Tagging Using Bi-LSTM with Word Embedding and Character-Level Embedding [Електронний ресурс] / K. Bose, K. Sarkar // Proceedings of International Conference on Frontiers in Computing and Systems. Lecture Notes in Networks and Systems, vol 404 / ред. S. Basu, D. K. Kole, A. K. Maji, D. Plewczynski, D. Bhattacharjee. – Springer, Singapore, 2023. – Режим доступу: https://doi.org/10.1007/978-981-19-0105-8_55 (дата звернення: 28.05.2024).
10. Brill, E. A Simple Rule-Based Part of Speech Tagger [Електронний ресурс] / E. Brill // Department of Computer Science, University of Pennsylvania. – Режим доступу: https://aclanthology.org/H92-1022.pdf (дата звернення: 28.05.2024).
11. Can, B. Statistical Models for Unsupervised Learning of Morphology and POS Tagging: PhD Thesis [Електронний ресурс] / B. Can. – University of York, 2011. – Режим доступу: https://etheses.whiterose.ac.uk/2364/ (дата звернення: 28.05.2024).
12. Chaplinsky, D. LT2OpenCorpora [Електронний ресурс] / D. Chaplinsky // GitHub. – Режим доступу: https://github.com/dchaplinsky/LT2OpenCorpora (дата звернення: 28.05.2024).
13. Choudhury, M. Generative AI has a language problem / M. Choudhury // Nature Human Behaviour. – 2023. – Vol. 7. – P. 1802-1803. – Режим доступу: https://www.nature.com/articles/s41562-023-01716-4 (дата звернення: 28.05.2024). – DOI: https://doi.org/10.1038/s41562-023-01716-4.
14. Crochemore, M., Vérin, R. Direct construction of compact directed acyclic word graphs / M. Crochemore, R. Vérin // Combinatorial Pattern Matching. CPM 1997. Lecture Notes in Computer Science, vol 1264 /

ред. A. Apostolico, J. Hein. – Berlin, Heidelberg : Springer, 1997. – Режим доступу: https://doi.org/10.1007/3-540-63220-4_55.
15. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [Електронний ресурс] / J. Devlin, M.-W. Chang, K. Lee, K. Toutanova // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) / ред. J. Burstein, C. Doran, T. Solorio. – Minneapolis, Minnesota : Association for Computational Linguistics, 2019. – С. 4171-4186. – Режим доступу: https://aclanthology.org/N19-1423 (дата звернення: 28.05.2024). – DOI: 10.18653/v1/N19-1423.
16. Dozat, T., Manning, C. D. Deep Biaffine Attention for Neural Dependency Parsing [Електронний ресурс] / T. Dozat, C. D. Manning // Published as a conference paper at ICLR 2017. – Режим доступу: https://nlp.stanford.edu/pubs/dozat2017deep.pdf (дата звернення: 28.05.2024).
17. flair-uk-pos [Електронний ресурс] // Hugging Face. – Режим доступу: https://huggingface.co/lang-uk/flair-uk-pos(дата звернення: 28.05.2024).
18. Ghosh, S., Mishra, B. K. Parts of Speech Tagging in NLP: Utility, Types, and Popular POS Taggers [Електронний ресурс] / S. Ghosh, B. K. Mishra // Taylor & Francis Group. – Режим доступу: https://www.taylorfrancis.com/chapters/edit/10.1201/9780367808495-6/parts-speech-tagging-nlp-utility-types-popular-pos-taggers-soumitra-ghosh-brojo-kishore-mishra (дата звернення: 28.05.2024).
19. Google Scholar [Електронний ресурс] // Google. – Режим доступу: https://scholar.google.com/ (дата звернення: 28.05.2024).
20. Ishida, T., Yamane, I., Sakai, T., Niu, G., Sugiyama, M. Do We Need Zero Training Loss After Achieving Zero Training Error? [Електронний ресурс] / T. Ishida, I. Yamane, T. Sakai, G. Niu, M. Sugiyama // arXiv.org. – 2020. – Режим доступу: https://arxiv.org/abs/2002.08709 (дата звернення: 28.05.2024).
21. Kupiec, J. Robust part-of-speech tagging using a hidden Markov model [Електронний ресурс] / J. Kupiec // Computer Speech &

Language. – 1992. – Vol. 6, Issue 3. – P. 225-242. – Режим доступу: https://doi.org/10.1016/0885-2308(92)90019-Z (дата звернення: 28.05.2024).
22. Kim, J.-K., Kim, Y.-B., Sarikaya, R., Fosler-Lussier, E. Cross-Lingual Transfer Learning for POS Tagging without Cross-Lingual Resources [Електронний ресурс] / J.-K. Kim, Y.-B. Kim, R. Sarikaya, E. Fosler-Lussier // Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. – 2017. – С. 2832-2838. – Режим доступу: https://aclanthology.org/D17-1302 (дата звернення: 28.05.2024).
23. KoichiYasuoka. roberta-base-ukrainian-upos [Електронний ресурс] // Hugging Face. – Режим доступу: https://huggingface.co/KoichiYasuoka/roberta-base-ukrainian-upos (дата звернення: 28.05.2024).
24. LanguageTool [Електронний ресурс] // LanguageTool. – Режим доступу: https://languagetool.org/about (дата звернення: 28.05.2024).
25. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach [Електронний ресурс] / Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov // arXiv.org. – 2019. – Режим доступу: https://arxiv.org/abs/1907.11692 (дата звернення: 28.05.2024).
26. Lyu, C., Du, Z., Xu, J., Duan, Y., Wu, M., Lynn, T., Aji, A. F., Wong, D. F., Wang, L. A Paradigm Shift: The Future of Machine Translation Lies with Large Language Models [Електронний ресурс] / C. Lyu, Z. Du, J. Xu, Y. Duan, M. Wu, T. Lynn, A. F. Aji, D. F. Wong, L. Wang // Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024) / ред. N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue. – Torino, Italia : ELRA and ICCL, 2024. – С. 1339-1352. – Режим доступу: https://aclanthology.org/2024.lrec-main.120 (дата звернення: 28.05.2024).
27. Machado, M., Ruiz, E. Evaluating large language models for the tasks of PoS tagging within the Universal Dependency framework / M.

Machado, E. Ruiz // Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1 / ред. P. Gamallo, D. Claro, A. Teixeira, L. Real, M. Garcia, H. G. Oliveira, R. Amaro. – Santiago de Compostela, Galicia/Spain : Association for Computational Linguistics, 2024. – Режим доступу: https://aclanthology.org/2024.propor-1.46 (дата звернення: 28.05.2024).

28. Mastering spaCy: Transformers and spaCy [Електронний ресурс] // Educative.io. – Режим доступу: https://www.educative.io/courses/mastering-spacy/transformers-and-spacy (дата звернення: 28.05.2024).

29. Mehta, H., Bharti, S. K., Doshi, N. Comparative Analysis of Part of Speech(POS) Tagger for Gujarati Language using Deep Learning and Pre-Trained LLM [Електронний ресурс] / H. Mehta, S. K. Bharti, N. Doshi // Proceedings of the 2024 3rd International Conference for Innovation in Technology (INOCON). – Bangalore, India : IEEE, 2024. – С. 1-3. – Режим доступу: https://ieeexplore.ieee.org/abstract/document/10511678 (дата звернення: 28.05.2024). – DOI: 10.1109/INOCON60754.2024.10511678.

30. Montes-Alcalá, C. Bilingual Texting in the Age of Emoji: Spanish–English Code-Switching in SMS / C. Montes-Alcalá // Languages. – 2024. – Vol. 9, No. 4. – С. 144. – Режим доступу: https://www.mdpi.com/2226-471X/9/4/144(дата звернення: 28.05.2024). – DOI: https://doi.org/10.3390/languages9040144.

31. OpenAI API Documentation [Електронний ресурс] // OpenAI. – Режим доступу: https://platform.openai.com/docs/api-reference/introduction (дата звернення: 28.05.2024).

32. OpenAI Fine-Tuning Guide [Електронний ресурс] // OpenAI. – Режим доступу: https://platform.openai.com/docs/guides/fine-tuning (дата звернення: 28.05.2024).

33. POS Tagger Highlighter [Електронний ресурс] // Streamlit. – Режим доступу: https://pos-tagger-highlighter-kyryl0.streamlit.app/ (дата звернення: 28.05.2024).

34. Pradhan, A., Yajnik, A. Parts-of-speech tagging of Nepali texts with Bidirectional LSTM, Conditional Random Fields and HMM / A.

Pradhan, A. Yajnik // Multimedia Tools and Applications. – 2024. – Vol. 83, No. 4. – C. 9893-9909.
35. pymorphy2 [Електронний ресурс] // GitHub. – Режим доступу: https://github.com/pymorphy2/pymorphy2/blob/master/CHANGES.rst (дата звернення: 28.05.2024).
36. Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C. D. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages [Електронний ресурс] / P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C. D. Manning // arXiv.org. – Режим доступу: https://arxiv.org/abs/2003.07082 (дата звернення: 28.05.2024).
37. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. Language Models are Unsupervised Multitask Learners / A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP). – 2019. – Режим доступу: https://api.semanticscholar.org/CorpusID:160025533 (дата звернення: 28.05.2024).
38. Ratnaparkhi, A. A Maximum Entropy Model for Part-Of-Speech Tagging [Електронний ресурс] / A. Ratnaparkhi // University of Pennsylvania, Dept. of Computer and Information Science. – 1996. – Режим доступу: https://aclanthology.org/W96-0213.pdf (дата звернення: 28.05.2024).
39. Sennrich, X. The Syntax and Morphology of English -ing: Thesis submitted for the degree of Doctor of Philosophy / X. Sennrich. – School of Philosophy, Psychology and Language Sciences, The University of Edinburgh, 2019.
40. Schmid, H. Part-of-Speech Tagging with Neural Networks [Електронний ресурс] / H. Schmid // arXiv.org. – Режим доступу: https://arxiv.org/abs/cmp-lg/9410018 (дата звернення: 28.05.2024).
41. spaCy Models [Електронний ресурс] // spaCy. – Режим доступу: https://spacy.io/usage/models (дата звернення: 28.05.2024).
42. Streamlit Documentation [Електронний ресурс] // Streamlit. – Режим доступу: https://docs.streamlit.io/ (дата звернення: 28.05.2024).
43. Subedi, B., Regmi, S., Bal, B. K., Acharya, P. Exploring the Potential of Large Language Models (LLMs) for Low-resource

Languages: A Study on Named-Entity Recognition (NER) and Part-Of-Speech (POS) Tagging for Nepali Language [Електронний ресурс] / B. Subedi, S. Regmi, B. K. Bal, P. Acharya // Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). – ELRA and ICCL, 2024. – С. 6974-6979. – Режим доступу: https://aclanthology.org/2024.lrec-main.611 (дата звернення: 28.05.2024).

44. TikTok video [Електронний ресурс] // TikTok. – Режим доступу: https://vm.tiktok.com/ZMr1Vb83o/ (дата звернення: 28.05.2024).

45. Universal Dependencies [Електронний ресурс] // Universal Dependencies. – Режим доступу: https://universaldependencies.org/ (дата звернення: 28.05.2024).

46. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I. Attention Is All You Need [Електронний ресурс] / A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin // arXiv.org. – 2017. – Режим доступу: https://arxiv.org/abs/1706.03762 (дата звернення: 28.05.2024).

47. Wang, P., Qian, Y., Soong, F. K., He, L., Zhao, H. Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network [Електронний ресурс] / P. Wang, Y. Qian, F. K. Soong, L. He, H. Zhao // arXiv.org. – Режим доступу: https://doi.org/10.48550/arXiv.1510.06168 (дата звернення: 28.05.2024).

48. Xu, H., Kim, Y. J., Sharaf, A., Awadalla, H. H. A Paradigm Shift in Machine Translation: Boosting Translation Performance of Large Language Models [Електронний ресурс] / H. Xu, Y. J. Kim, A. Sharaf, H. H. Awadalla // arXiv.org. – Режим доступу: https://arxiv.org/abs/2309.11674 (дата звернення: 28.05.2024).

49. Yoo, H., Kang, M., Oh, K. A Semantic Search Model Using Word Embedding, POS Tagging, and Named Entity Recognition / H. Yoo, M. Kang, K. Oh // Proceedings of the 2018 International Conference on Computational Science and Computational Intelligence (CSCI). – Las Vegas, NV, USA : IEEE, 2018. – С. 1204-1209. – Режим доступу:

https://doi.org/10.1109/CSCI46756.2018.00231 (дата звернення: 28.05.2024).
50. Zhang, A., Huang, J., Li, P., Zhang, K. Building Shortcuts between Distant Nodes with Biaffine Mapping for Graph Convolutional Networks / A. Zhang, J. Huang, P. Li, K. Zhang // Association for Computing Machinery. – 2024. – Vol. 18, No. 6. – Режим доступу: https://doi.org/10.1145/3650113 (дата звернення: 28.05.2024).

ДОДАТОК А

1. Повна версія порівняльної таблиці з тегуванням літературних текстів: https://docs.google.com/spreadsheets/d/1sHdyscsk_KKLbJhU0UC1GZqbI9CESsXnydz1kpI6m3g/edit?usp=sharing
2. Повна версія порівняльної таблиці з тегуванням нелітературних текстів, оцінена експертами: https://docs.google.com/spreadsheets/d/1NIKSFu2LNSWD2IReRMI7e9XezKGoejiUdvjHSqQEn70/edit?usp=sharing