УДК 004.415.2
DOI https://doi.org/10.24919/2308-4863/96-1-30

Юліан БАРАНЕЦЬКИЙ,
orcid.org/0009-0008-5545-0865
аспірант кафедри прикладної лінгвістики
Інституту комп'ютерних наук та інформаційних технологій
Національного університету «Львівська політехніка»
(Львів, Україна) yulian.baranetskyy@gmail.com

Наталія КУНАНЕЦЬ,
orcid.org/0000-0003-3007-2462
професор кафедри інформаційних систем та мереж
Інституту комп'ютерних наук та інформаційних технологій
Національного університету «Львівська політехніка»
(Львів, Україна) nek.lviv@gmail.com

ПОСТКОРЕКЦІЯ HTR УКРАЇНСЬКИХ РУКОПИСІВ ЗАСОБАМИ КОРПУСНОЇ ЛІНГВІСТИКИ N-ГРАМНЕ МОДЕЛЮВАННЯ МОВИ

Статтю присвячено проблемі підвищення якості автоматичного розпізнавання рукописних кириличних текстів українською мовою шляхом посткорекції результатів HTR/OCR на основі корпусної лінгвістики. Актуальність теми зумовлена складністю розпізнавання рукописів через варіативність почерків, шумність зображень, помилки сегментації, специфіку кирилиці та наявність історичних орфографічних норм, які часто не підтримуються стандартними сучасними мовними ресурсами. Метою роботи є проаналізувати можливості корпусних методів у задачах HTR/OCR та запропонувати практичну архітектуру посткорекції, що спирається на n-грамне моделювання мови й контекстне ранжування кандидатів виправлення.

У теоретичній частині узагальнено роль корпусів у побудові мовних моделей, окреслено переваги n-грам у знятті неоднозначностей під час розпізнавання, а також розглянуто згладжування, частотний аналіз і колокаційний підхід як інструменти корпусної підтримки. Практична частина реалізує модуль посткорекції за схемою «генерація кандидатів + LM-rescoring» у постановці noisy channel. Для ефективного скорингу використано KenLM; кандидати формуються комбіновано: підстановки символів (sub1/sub2), розклейка злиплих токенів (split) та SymSpell-подібний пошук для вставок/видалень з обмеженням відстані Левенштейна ≤ 2. Для мінімізації доменного зсуву мовну модель і словникові ресурси побудовано на історичному корпусі PluG (Pluperfect GRAC), що краще відповідає орфографії та лексиці досліджуваного матеріалу.

Експеримент проведено на рукописному уривку публіцистичного характеру 1940 року. Після застосування посткорекції система виконала 20 виправлень і суттєво зменшила помилковість: CER знизився з 7,86% до 2,16%, а WER – з 48,61% до 12,50%, тобто більш ніж утричі. Показано, що підхід ефективний для типових OCR-артефактів (плутанини графем, омографи латиниці/кирилиці, злипання слів), однак має обмеження у випадках морфологічної неоднозначності, складної сегментації та власних назв поза словником. Запропоновано напрями вдосконалення: використання n-best/lattice-гіпотез розпізнавача, доменна/жанрово-хронологічна адаптація мовної моделі, морфологічна валідація кандидатів і окремий модуль нормалізації переносів та дефісів.

Ключові слова: *HTR, OCR, посткорекція, корпусна лінгвістика, n-грамна мовна модель, KenLM, SymSpell, відстань Левенштейна, CER, WER, історичні документи.*

Yulian BARANETSKYI,
orcid.org/0009-0008-5545-0865
PhD Student Applied Linguistics
Institute of Computer Science and Information Technologies of Lviv Polytechnic National University
(Lviv, Ukraine) yulian.baranetskyy@gmail.com

Natalia KUNANETS,
orcid.org/0000-0003-3007-2462
Professor at the Department of Information Systems and Networks
Institute of Computer Science and Information Technologies of Lviv Polytechnic National University
(Lviv, Ukraine) nek.lviv@gmail.com

POST-CORRECTION OF UKRAINIAN HANDWRITTEN TEXT RECOGNITION (HTR) USING CORPUS LINGUISTICS: N-GRAM LANGUAGE MODELLING

The article focuses on improving the quality of automatic recognition of handwritten Cyrillic texts in Ukrainian by applying corpus-linguistic post-correction to HTR/OCR output. The relevance of the topic stems from the inherent complexity of handwritten recognition caused by the variability of individual handwriting, image noise, segmentation

errors, Cyrillic-specific confusability, and the presence of historical orthographic norms that are frequently unsupported by standard contemporary language resources. The purpose of the study is to analyse the potential of corpus-based methods for HTR/OCR tasks and to propose a practical post-correction architecture based on n-gram language modelling and context-sensitive candidate ranking.

In the theoretical part, the paper generalises the role of corpora in constructing language models, outlines the advantages of n-grams for resolving ambiguity during recognition, and discusses smoothing, frequency-based evidence, and a collocational approach as instruments of corpus support. The practical part implements a post-correction module following the «candidate generation + LM rescoring» scheme within the noisy-channel framework. KenLM is used for efficient scoring. Candidate corrections are generated in a hybrid manner: character substitutions (sub1/sub2), splitting of erroneously merged tokens (split), and a SymSpell-like search for insertion/deletion edits with a constraint of Levenshtein distance ≤ 2. To minimise domain shift, the language model and lexical resources are built on the historical PluG corpus (Pluperfect GRAC), which better matches the orthography and vocabulary of the analysed material.

An experiment is conducted on a handwritten publicistic excerpt from 1940. After applying post-correction, the system performs 20 edits and significantly reduces error rates: CER decreases from 7.86% to 2.16%, and WER drops from 48.61% to 12.50%, i.e., by more than a factor of three. The results demonstrate that the approach is effective for typical OCR artefacts (grapheme confusions, Latin/Cyrillic homoglyphs, and word concatenations), while it remains limited in cases of morphological ambiguity, complex segmentation, and proper names missing from the lexicon. The paper proposes directions for further improvement, including using n-best/lattice hypotheses from the recogniser, domain/genre and diachronic adaptation of the language model, morphological validation of candidates, and a dedicated module for normalising line-break hyphenation and dash usage.

Key words: *HTR, OCR, post-correction, corpus linguistics, n-gram language model, KenLM, SymSpell, Levenshtein distance, CER, WER, historical documents.*

Постановка проблеми. Сучасний етап розвитку цифрових технологій супроводжується активною інтеграцією систем автоматичного розпізнавання текстів у різні галузі наукового та суспільного життя. Особливо актуальним є використання систем оптичного розпізнавання символів для опрацювання рукописних текстів, оскільки велика кількість важливих документів, історичних джерел, архівних матеріалів, наукових та культурних пам'яток існує саме у формі рукописних матеріалів (Tikhonov, Rabus, 2024). Водночас автоматичне розпізнавання рукописних кириличних текстів залишається одним із найскладніших завдань у галузі комп'ютерного опрацювання природної мови через значні варіації почерку, неоднорідність стилів письма, наявність скорочень, абревіатур, а також різноманітних помилок і неточностей, що ускладнює побудову ефективних автоматичних систем.

Актуальність цієї тематики підсилюється тим, що останніми роками помітно зросла кількість досліджень, спрямованих на вдосконалення алгоритмів розпізнавання тексту за допомогою методів машинного навчання, зокрема нейронних мереж та великих мовних моделей (Carbune et al., 2020). Однак, поряд із розробками в сфері нейромережевих підходів, усе більш актуальними стають корпусні методи, які дозволяють суттєво підвищити точність роботи розпізнавальних систем завдяки застосуванню мовних моделей, побудованих на великих, репрезентативних текстових масивах (корпусах). Таким чином, використання корпусного підходу є перспективним напрямком, який здатний істотно покращити ефективність сучасних OCR-систем та сприяти автоматизації цифровізації культурної та історичної спадщини.

Аналіз досліджень. Автоматичне розпізнавання рукописних текстів є складною задачею через особливості письма, серед яких варіативність форми літер, індивідуальні стилістичні особливості почерку, складність сегментації тексту, присутність помилок та шумів у зображеннях (Brodic et al., 2015). Окрім цього, специфіка кириличних текстів ускладнює завдання OCR порівняно з латинськими через більшу кількість графем, більшу подібність форм символів та меншу кількість доступних тренувальних наборів даних (Баранецький, Кунанець, 2025). Сучасні системи OCR використовують для вирішення цієї проблеми різноманітні алгоритми, починаючи від класичних статистичних моделей до складних глибоких нейромережевих архітектур, проте досягти ідеального рівня точності розпізнавання досі не вдалося.

Використання корпусної лінгвістики для вдосконалення роботи OCR-систем ґрунтується на здатності текстових корпусів надавати об'єктивні статистичні дані щодо мовних закономірностей, лексичних і граматичних структур, а також колокацій і частотності вживання певних слів і словосполучень. Завдяки цим особливостям корпусні методи дозволяють значно розширити можливості систем автоматичного розпізнавання текстів.

Одним із поширених способів використання корпусних даних є побудова статистичних n-грамних моделей. Ці моделі, навчені на великому обсязі текстових корпусів, дають змогу ефективно прогнозувати ймовірність появи слів або символів у певному контексті. Використовуючи ці ймовірності, OCR-системи можуть краще вирішувати неоднозначності, які виникають під час розпізнавання складних чи пошкоджених фрагментів

тексту (Tarride, Kermorvant, 2024) Корпусні дані також можуть застосовуватись для автоматичного виправлення помилок шляхом порівняння розпізнаного тексту із наявними в корпусі типовими граматичними і лексичними конструкціями.

Окрім статистичних моделей, корпусні підходи можуть використовуватись для побудови спеціалізованих словників та лексиконів, адаптованих до конкретних завдань, таких як розпізнавання рукописних текстів певної епохи чи регіону. Це дозволяє суттєво скоротити кількість помилок, пов'язаних із неправильним розпізнаванням специфічних термінів, застарілих слів чи регіональних особливостей письма.

Важливою перевагою корпусних методів є також можливість використання великих репрезентативних текстових масивів для тренування нейронних мереж та інших методів машинного навчання. Завдяки корпусним даним розробники OCR-систем можуть швидко адаптувати моделі до нових текстових джерел, покращуючи тим самим загальну точність розпізнавання. Корпусний підхід допомагає вирішити проблему нестачі даних для тренування нейромережевих моделей, особливо у випадках рідкісних або малоресурсних мовних контекстів.

Таким чином, корпусна лінгвістика пропонує широкий набір інструментів, що здатні суттєво покращити ефективність сучасних OCR-систем, надаючи їм надійні статистичні моделі, спеціалізовані лексикони і якісні тренувальні дані.

Метою статті є аналіз можливостей та особливостей використання методів корпусної лінгвістики для підвищення ефективності автоматичного розпізнавання рукописних кириличних текстів.

Завдання статті включають:

1. Огляд теоретичних засад корпусного підходу і специфіки його застосування в задачах OCR.

2. Детальне дослідження можливостей використання корпусних n-грамних моделей для покращення якості автоматичного розпізнавання рукописних текстів.

3. Представлення конкретних прикладів роботи з кириличними корпусами, зокрема з Генеральним регіонально анотованим корпусом української мови (Шведова та ін., 2025).

4. Оцінку переваг та обмежень корпусного підходу.

Виклад основного матеріалу. Корпусна лінгвістика – це галузь мовознавства, що досліджує мову на основі великих зібрань текстів (корпусів), які дозволяють вивчати мовні закономірності за допомогою кількісних методів.

Текстовий корпус – це структурована електронна колекція текстів, що може бути анотована лінгвістичною інформацією, такою як частини мови, синтаксичні структури, семантичні ролі тощо. Корпуси класифікуються за різними критеріями: за жанром (наукові, художні, публіцистичні), за мовою, за часом створення (сучасні, історичні), за способом анотації (неанотовані, частково або повністю анотовані).

Корпусна лінгвістика надає потужні інструменти для моделювання мови, що є критично важливим у задачах автоматичного розпізнавання рукописного тексту. Серед таких методів особливе місце займають n-грамні моделі, частотний аналіз та дослідження колокацій.

N-грамні моделі є фундаментальним інструментом у статистичному моделюванні мови, особливо в контексті оптичного розпізнавання символів (OCR). Ці моделі дозволяють враховувати ймовірність появи слова або символу на основі попередніх елементів у тексті, що є критично важливим при розпізнаванні рукописного тексту, де часто виникають помилки через неоднозначність написання.

N-грамна модель оцінює ймовірність появи слова або символу на основі попередніх $n-1$ елементів. Наприклад, у біграмній моделі ($n=2$) ймовірність слова залежить від попереднього слова, а в триграмній ($n=3$) – від двох попередніх слів. Ці моделі дозволяють враховувати контекст при розпізнаванні тексту, що особливо корисно при опрацюванні рукописних документів з неоднозначним написанням.

У задачах OCR, особливо при розпізнаванні рукописного тексту, N-грамні моделі використовуються для покращення точності розпізнавання. Вони дозволяють враховувати контекст при розпізнаванні символів, що зменшує ймовірність помилок. Наприклад, якщо система розпізнає слово як «дім», але в контексті попередніх слів більш ймовірним є «день», N-грамна модель може допомогти виправити цю помилку.

Однією з проблем при використанні N-грамних моделей є обмеженість корпусу, що може призводити до нульових ймовірностей для деяких послідовностей. Для вирішення цієї проблеми застосовуються методи згладжування.

Метод Гуда-Тьюринга оцінює ймовірності для невідомих N-грам на основі кількості N-грам, які спостерігались один раз. Основна ідея полягає в тому, що ймовірність для невідомих N-грам пропорційна кількості N-грам, які з'явились один раз у тренувальних даних. Цей метод ефективно працює для великих корпусів, де є достатньо

інформації про рідкісні події. Такий підхід було застосовано в системі рукописного розпізнавання арабського тексту в роботі Ахмеда Фаріс Раїд аль-Масуді та Хішама Салам Рафід аль-Убейді, де автори порівнювали різні техніки згладжування та продемонстрували значне зниження Word Error Rate (WER) при використанні Гуда–Тьюринга порівняно з базовими підходами (Al-Masoudi, Al-Obeidi, 2015).

Згладжування Кнесера-Нея – метод який вважається одним із найефективніших для згладжування в N-грамних моделях. Він базується на абсолютному зменшенні частот спостережень та враховує кількість унікальних контекстів, у яких з'являється слово. Формула для біграмної моделі:

$$P(w_i \mid w_{i-1}) = \max(C(w_{i-1}, w_i) - D, 0) / C(w_{i-1}) + \lambda(w_{i-1}) * P\_continuation(w_i)$$

де:

D – параметр зменшення;

$\lambda(w_{i-1})$ – коефіцієнт нормалізації;

$P\_continuation(w_i)$ – ймовірність слова w_i з'явитися в новому контексті.

У дослідженні Фішера (Fischer, 2020), проведеному в рамках проєкту HisDoc із розпізнавання історичних рукописів, було показано, що застосування згладжування Кнесера–Нея до триграмної моделі дозволило покращити якість розпізнавання на понад 8% порівняно з моделлю без згладжування. Особливо корисним цей метод виявився при роботі з короткими словами та функціональними словами, які часто не мають стабільного контексту. У задачах розпізнавання рукописного тексту, особливо для мов з кириличною абеткою, згладжування в N-грамних моделях є критично важливим. Рукописний текст часто містить варіації в написанні, скорочення та інші особливості, які можуть призводити до появи невідомих N-грам. Застосування методів згладжування дозволяє системам OCR більш точно розпізнавати текст, зменшуючи кількість помилок і покращуючи загальну якість розпізнавання.

Частотний аналіз є одним із найдавніших і водночас найефективніших методів опрацювання текстових даних. Його суть полягає у підрахунку кількості появ окремих мовних одиниць (словоформ, лем, морфем, символів) у тексті або корпусі текстів. Цей підхід дозволяє виявити закономірності вживання слів, визначити ключові терміни, а також дослідити стилістичні та жанрові особливості текстів.

На відміну від N-грамних моделей, які аналізують послідовності слів, частотний аналіз фокусується на окремих одиницях без урахування їхнього порядку. Це робить його особливо корисним для задач, де важлива саме частота вживання, а не контекст.

Частотний аналіз базується на припущенні, що мовні одиниці в текстах розподілені нерівномірно: деякі слова з'являються дуже часто, тоді як інші – рідко. Це явище описується законом Ципфа, згідно з яким частота слова обернено пропорційна його рангу в частотному списку. Такий розподіл дозволяє ефективно ідентифікувати ключові слова та терміни в текстах.

Частотний аналіз знаходить широке застосування в розпізнаванні рукописного тексту:

– Покращення точності розпізнавання. В умовах, коли система HTR має справу з пошкодженими, неповними або шумними зображеннями тексту, розпізнавання на рівні символів або слів може бути неточним. У таких випадках частотні словники відіграють ключову роль у процесі post-correction – автоматичного виправлення результатів OCR/HTR. Система може зіставити розпізнане слово з частотним словником мови або певного тематичного домену. Якщо слово не входить до переліку високочастотних одиниць, воно вважається ймовірно помилковим і пропонується заміна на найбільш схоже (за формою та ймовірністю) слово з лексикону. У цьому контексті важливу роль відіграє підхід, запропонований у Юнгом (Jung et al., 2024), де частотність використовується як фактор довіри до слова. У тестових експериментах частотний словник покращив результати розпізнавання в умовах поганої якості зображення, підвищивши точність декодування більш ніж на 3%. Це особливо помітно при роботі з реальними документами, де можуть бути плями, лінії, або інші дефекти, які частково спотворюють символи.

– Ідентифікація спеціалізованої лексики. Частотний аналіз ефективно застосовується для виявлення та підсилення розпізнавання термінів, що не є загальновживаними, але часто трапляються в специфічному контексті. У тому самому дослідженні Юнга, проведеному на корпусах академічних лекцій, розроблено метод оцінки відносної частотності слів у розпізнаних текстах слайдів. Ці частотності потім використовувалися як вхідні ознаки для мовної моделі, яка опрацьовувала аудіо (ASR), але метод прямо релевантний для HTR, оскільки лінгвістичні моделі однакові. Наприклад, якщо у лекції з анатомії часто з'являються слова «синапс», «нейрон», гіпокамп», вони отримують вищу вагу в частотному словнику. Під час розпізнавання системі легше припустити, що незрозуміле слово є, наприклад, «гіпокамп», а не «гірокам», виходячи з розповсюдженості терміну в контексті. Автори зафіксували зростання точ-

ності на 3,2%, коли цей метод використовувався для покращення лінгвістичної моделі.

– Формування частотних профілів почерку. Інший напрям застосування частотного аналізу в HTR стосується графемного рівня, тобто форми й частоти написання символів. У звіті, опублікованому Національним інститутом юстиції США у 2019 році (Johnson et al., 2019), було проведено масштабне дослідження графічних характеристик рукопису на основі аналізу частоти появи окремих елементів почерку: типових рисок, нахилів, зв'язків між буквами тощо. Мета дослідження – створити частотні профілі, які дозволяють не лише покращити точність розпізнавання через адаптацію до індивідуального стилю написання, а й ідентифікувати автора або перевірити автентичність документа. У межах HTR така інформація може бути використана як додаткове джерело лінгвістичної або візуальної пріоризації: якщо певна форма літери «г» зустрічається частіше за іншу, система може віддати перевагу саме цій формі при амбівалентному розпізнаванні.

Колокації – це стійкі поєднання слів, які часто зустрічаються разом у мові. На відміну від N-грам, які розглядають послідовності слів без урахування їхньої лексичної сполучуваності, колокації фокусуються на семантичній та синтаксичній взаємодії між словами. Це дозволяє системам HTR краще розпізнавати слова в контексті, зменшуючи кількість помилок, пов'язаних із неоднозначністю рукописного тексту. У цьому контексті колокаційний аналіз виступає як ефективний інструмент для покращення точності розпізнавання, особливо у випадках, коли інші методи, такі як частотний або N-грамний аналіз, можуть бути менш ефективними через спарсність даних.

У дослідженні Роуз Т. Г. Та Еветт Л. Дж. (Rose, Evett, 1993) було продемонстровано, що використання колокаційного аналізу дозволяє значно покращити точність розпізнавання рукописного тексту. Зокрема, було показано, що врахування колокаційних зв'язків між словами дозволяє зменшити кількість помилок, пов'язаних із неоднозначністю рукописного тексту.

В рамках даної статті, було вирішено використати n-грамну мовну модель як статистичну основу посткорекції результатів HTR/OCR: замість того, щоб «вгадувати» правильну форму за ізольованим словом, модель оцінює мовність (правдоподібність) кожного кандидата в контексті сусідніх токенів, і вибір робиться за принципом максимальної імовірності/найвищої LM-оцінки.

У практичній частині роботи n-грамна модель реалізована на базі KenLM (Heafield, 2011) використовується як компонент ранжування для множини кандидатів виправлення, згенерованих на рівні токенів. Такий підхід відповідає класичній постановці «noisy channel»: базовий розпізнавач генерує текст із помилками як результат дії «шумного каналу», а мовна модель слугує апріорною оцінкою того, наскільки природною є послідовність слів після заміни помилкового токена на кандидат (Brill, Moore, 2000).

KenLM застосовується як ефективна реалізація n-грамних мовних моделей, що підтримує:

– навчання n-грамних моделей із текстових корпусів;

– побудову компактних бінарних моделей для швидкого скорингу;

– обчислення лог-правдоподібностей фраз/контекстів, що критично для швидкого порівняння багатьох кандидатів у локальному вікні.

У моделі використовується ймовірність послідовності токенів $w_1, w_2, \ldots, w_T$, яка в n-грамному наближенні факторизується як:

$$P\left(w_1^T\right) \approx \prod_{i=1}^{T} P\left(w_i \# w_{i-n+1}^{i-1}\right)$$

де w_{i-n+1}^{i-1} – контекст із попередніх $n-1$ слів (Jurafsky, Martin, 2023).

У задачі посткорекції це означає: для кожного проблемного токена формується набір альтернатив $c \in \mathcal{C}$, і вибирається та, що максимізує LM-оцінку в локальному контексті:

$$\hat{c} = \underset{c \in \mathcal{C}}{\operatorname{argmax}} \log P(\text{контекст із } c)$$

Практично це реалізується як порівняння приросту LM-скору (gain) відносно базового варіанта без виправлення; також вводиться поріг, який зменшує ризик надкорекції (застосовувати лише зміни з достатньо позитивним gain).

Важливо підкреслити, що LM сама по собі не «знає», як саме виправити слово: вона лише обирає найкращий варіант серед запропонованих. Тому якість посткорекції суттєво залежить від генератора кандидатів (Garbe, 2012).

У роботі використовується відстань Левенштейна у двох взаємопов'язаних ролях:

1. як механізм контролю «близькості» кандидата до помилкового токена (під час генерації/верифікації кандидатів) (Navarro, 2001);

2. як математична основа метрик CER/WER, за якими оцінюється якість системи (розділ із метриками буде подано в практичній частині) (Manning, Schütze, 1999).

Відстань Левенштейна визначається як мінімальна кількість елементарних операцій редагування (вставка I, видалення D, заміна S), необхідних для перетворення одного рядка на інший.

Формально це класична динамічна рекурентна схема:

$$\mathrm{lev}(a,b) = \min\begin{cases}\mathrm{lev}\left(a_{1..i-1}, b_{1..j}\right)+1\\ \mathrm{lev}\left(a_{1..i}, b_{1..j-1}\right)+1\\ \mathrm{lev}\left(a_{1..i-1}, b_{1..j-1}\right)+\left[a_i \neq b_j\right]\end{cases}$$

де $\left[a_i \neq b_j\right]$ дорівнює 0, якщо символи рівні, і 1 – якщо різні.

Для посткорекції HTR/OCR це важливо з практичної точки зору: типові помилки рукописного розпізнавання включають пропуски символів, зайві символи, помилки підстановки, тобто саме ті трансформації, які модель редагування описує природним способом. Саме тому обмеження кандидатів на відстань редагування (наприклад, edit distance ≤ 2) дозволяє отримувати «реалістичні» альтернативи без перебору всього словника (Norvig, 2007).

Окремо варто зазначити: використання edit distance в генерації кандидатів не означає автоматичного вибору «найближчого» слова – остаточне рішення приймає мовна модель, яка враховує контекст. Таким чином edit distance виступає фільтром/механізмом скорочення пошуку, а KenLM – контекстним критерієм оптимальності.

Оскільки мовна модель виконує ранжування, необхідно сформувати множину правдоподібних кандидатів. У цій роботі використано комбіновану генерацію кандидатів, орієнтовану на типові дефекти HTR/OCR для кирилиці.

Перший клас кандидатів, Substitution-кандидати (sub1, sub2) – підстановки символів у межах слова. Логіка узгоджується з типовими плутанинами рукописного розпізнавання (візуальна схожість графем, нестабільність штрихів тощо), коли помилка проявляється як заміна однієї літери іншою.

– sub1: усі слова, що відрізняються від помилкового рівно однією заміною, після чого виконується відсів за словником (залишаються лише словникові форми).

– sub2: дві заміни застосовуються лише як «fallback» для довших OOV-токенів і лише якщо кандидат присутній у словнику (щоб обмежити розмір кандидатного простору і ризик надкорекції).

У літературі подібні моделі помилок розглядаються як спрощений варіант noisy-channel spelling correction, де генератор кандидатів відповідає компоненту «channel model», а мовна модель – апріорному розподілу (Ahmad, Kondrak, 2005).

Другий клас кандидатів, Split-кандидати для склеєних токенів, спрямований на помилку відсутності пробілу (token glue): розпізнавач може повернути два слова як один токен. Тоді для довших OOV слів перевіряються всі позиції розрізу, і кандидат типу «left\right» формується лише тоді, коли обидві частини присутні у словнику.

Цей тип помилки не завжди добре «захоплюється» edit distance, бо проблема не лише у символах, а в межі токенізації. Тому split-кандидати виділяються в окрему гілку генератора (Palmer, 2010).

Третій клас кандидатів, SymSpell-підхід для вставок/видалень (edit distance ≤ 2), охоплює помилки, які substitution-генератор принципово не покриває: вставка або видалення літери. Для цього застосовано SymSpell-подібну схему (symmetric delete):

– для словника наперед будується індекс «delete-форм» (усі рядки, що отримуються видаленням 1–2 символів зі словникових слів);

– для OOV токена також генеруються delete-форми;

– перетин ключів дає компактний список кандидатів, який потім верифікується точним edit distance (Левенштейн ≤ 2) (Boytsov, 2012).

Перевага цього підходу – масштабованість: замість порівняння OOV токена з усіма словами словника, пошук проводиться через індекс за delete-ключами, що радикально зменшує кількість перевірок. У практичній системі SymSpell використовується як безпечний «fallback» (коли немає або недостатньо якісні non-split кандидати), щоб не роздувати кандидатний простір на кожному слові (Karch et al., 2010).

У цьому дослідженні корпуси розглядаються як джерело емпіричних мовних закономірностей і доменно релевантних текстів для побудови мовної моделі, що виконує посткорекцію HTR/OCR. Було обрано n-грамну мовну модель на поверхневих словоформах (surface LM), для якої критичною є відповідність навчальних даних реальному стилю, орфографії та жанру розпізнаваного матеріалу. Саме тому як базу для тренування LM і побудови словникових ресурсів доцільно залучати два типи корпусів: (1) широкий репрезентативний корпус сучасної української мови для контролю та порівняння; (2) спеціалізований історичний/діахронний корпус, максимально близький до досліджуваних рукописних/друкованих джерел за правописом і лексикою.

ГРАК – великий анотований корпус української мови з корпусним менеджером і розвиненим пошуком, який використовують для частотних та контекстних досліджень, лексикографії й навчальних застосувань. Утім, для посткорекції

історичних текстів ГРАК у чистому вигляді менш придатний як основна база, адже сучасний корпус не гарантує покриття дореформених і архаїчних написань та форм, критичних для словникового фільтра системи.

PluG (також відомий як Pluperfect GRAC) (Shvedova, Lukashevskyi, 2026) – корпус старих/історичних українських текстів, доступний як відкритий ресурс, який може бути використаний для побудови словника та навчання мовних моделей під історичний матеріал. Важливо, що PluG надає саме той тип лінгвістичного «покриття», який потрібен у посткорекції:

– лексичне покриття історичних словоформ (зменшує кількість OOV та підвищує корисність корпусного фільтра кандидатів);

– контекстну типовість (n-грамна модель, навчена на індоменно близьких текстах, стабільніше ранжує виправлення в реальних синтагматичних умовах);

– керованість і відтворюваність експерименту (корпус можна зафіксувати як версіонований ресурс, а словник/частотні списки – як артефакти).

У розроблюваній системі PluG використано як єдине корпусне джерело експерименту: з нього сформовано словник (vocab) і похідні структури для генерації/фільтрації кандидатів; на його ж основі було навчено n-грамну мовну модель, яка здійснює контекстне переоцінювання (rescoring). Такий вибір мінімізує «підмішування» сучасної норми й підвищує валідність висновків: покращення якості виправлень можна інтерпретувати як ефект саме індоменної корпусної підтримки, а не як результат нормалізації під сучасний стандарт.

Якщо розпізнаваний фрагмент належить до старішого правопису/типографіки, містить архаїзми або нестандартизовані написання, то LM, навчена лише на сучасних корпусах, має тенденцію «притягувати» текст до сучасної норми, тобто породжувати надмірні або хибні заміни.

Доцільність PluG як корпусної бази підтверджується тим, що він уже використовується як ресурс у суміжних українських NLP-задачах, де потрібні тексти з нетиповою для сучасної мови орфографією або стилем. Наприклад, PluG фігурує як складник або джерело навчальних даних у роботах із багатомовного моделювання та корпусного бенчмаркінгу (Jumelet et al., 2025), де важливо мати «природні» (не синтетичні) тексти різних жанрів та епох. Це підкреслює ключову перевагу PluG саме для HTR-посткорекції: корпус задає допустимий простір словоформ і n-грамних контекстів у тій же орфографічній реальності, що й джерело розпізнавання, отже LM-rescoring «підкріплює» виправлення не правилами, а статистично правдоподібними контекстами.

У практичній частині реалізовано модуль посткорекції, який працює поверх вихідного результату базового HTR/OCR. Архітектурно система відповідає типовій схемі «candidate generation + language model rescoring»: для кожного потенційно помилкового токена генерується обмежений набір альтернатив, після чого альтернативи ранжуються за оцінкою n-грамної мовної моделі в локальному контексті.

Мовна модель навчається на корпусі текстів українською мовою який попередньо приводиться до вигляду «плоского» тренувального тексту з токенізацією на рівні слів. Практично це означає (а) нормалізацію регістру; (б) очищення від зайвих розділових знаків на краях токенів; (в) фільтрацію токенів за допустимим алфавітом (кириличні літери та апостроф), що зменшує шум у словникових структурах (Bird et al., 2009).

На цьому етапі формуються два ресурси:

1. n-грамна мовна модель у форматі, придатному для швидкого скорингу (KenLM). З технічної точки зору KenLM є стандартним інструментом для отримання швидких LM-оцінок для фрагментів тексту, що робить можливим перебір кандидатів на кожному проблемному токені.

2. Словникові множини (vocab та core-vocab), що будуються з тренувального корпусу: повний словник використовується як фільтр допустимих кандидатів, а «core» (за частотою) – як компактніша база для індексації SymSpell-подібного пошуку (Ramirez-Orta et al., 2022).

На рисунку 1, зображено пайплайн підготовки ресурсів для роботи системи. Ці ресурси є локальними компонентами та будуються одноразово для конкретного корпусу з можливістю перебудиови при зміні конфігурації системи.

Щоб зменшити ризик надкорекції та обмежити простір пошуку, модуль кандидатогенерації працює за принципом OOV-only: кандидати генеруються лише для токенів, відсутніх у словнику (за умови, що токен відповідає формальним критеріям «слово»: кирилиця/апостроф, мінімальна довжина, відсутність цифр). Такий підхід є поширеним практичним запобіжником у задачах виправлення помилок, оскільки дозволяє не «переписувати» вже коректні словоформи (Kolak, Resnik, 2002).

Окремо реалізовано SymSpell-подібний індекс для пошуку кандидатів з edit distance ≤ 2 через symmetric deletes. На практиці індекс будується на «core-vocab», щоб контролювати споживання пам'яті: короткі й надто рідкісні форми непро-

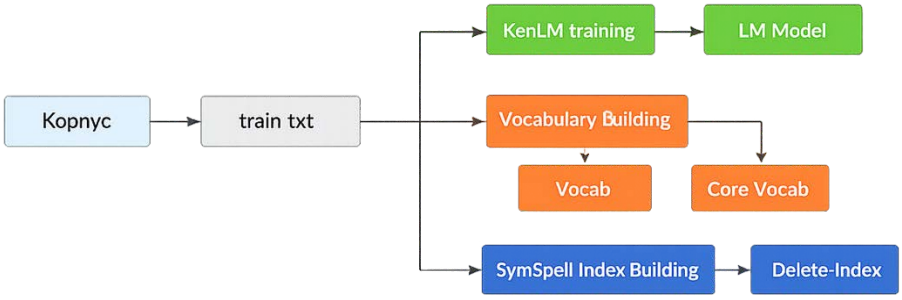

Рис. 1. Пайплайн підготовки ресурсів

порційно роздувають кількість delete-ключів та bucket-списків. Після отримання первинного списку кандидатів з індексу кожен кандидат додатково верифікується точною відстанню Левенштейна, що гарантує дотримання обмеження на кількість редагувань (Kukich, 1992).

Вхідним для посткорекції є текст, отриманий базовим HTR/OCR-модулем. Перед генерацією кандидатів застосовується технічна нормалізація, потрібна саме для підвищення стабільності обчислень:

– нормалізація омографів (латинські символи, що візуально схожі на кириличні, наприклад і→і, с→с), оскільки такі заміни є типовими артефактами OCR/HTR при змішаних шрифтах/артефактах;

– уніфікація апострофа та дефісів (різні Unicode-варіанти апострофа/тире зводяться до одного символу);

– склеювання переносів на межі рядка (патерн «слово – слово» з продовженням з малої літери), що зменшує кількість псевдо-OOV токенів, спричинених лише форматуванням (Madarász et al., 2024).

Генератор кандидатів комбінує кілька механізмів, кожен з яких адресує типові класи помилок HTR/OCR:

1. sub1/sub2 (підстановки): генерація кандидатів із заміною 1 символа (та, за потреби, 2 символів) із подальшим словниковим фільтром.

2. split (розклейка): для довших OOV токенів перевіряються можливі позиції розрізу; кандидат приймається лише якщо обидві частини існують у словнику.

3. SymSpell fallback (вставки/видалення): пошук кандидатів edit distance ≤ 2 через delete-index, з подальшою верифікацією Левенштейном.

Принциповим елементом є керованість кандидатного простору. Генерація обмежена порогами (максимальна кількість кандидатів, мінімальна довжина токена для SymSpell, верхня межа для в delete-index), щоб уникати вибуху кількості альтернатив. Контроль кандидатного простору робить можливим стабільне LM-reranking в обчислювально обмежених умовах (Kumar, 2019).

Далі для кожного токена обчислюється локальний контекстний скор мовної моделі. На рівні імплементації оцінка виконується не для всього документа, а в «вікні» навколо позиції токена (наприклад, ±6 токенів), що (а) зменшує обчислювальне навантаження; (б) фокусує рішення на локальній синтаксично-колокаційній узгодженості; (в) робить алгоритм придатним для покрокового застосування. Кандидат приймається лише тоді, коли його приріст скору перевищує заданий поріг, що зменшує ризик надкорекції та дозволяє інтерпретувати виправлення як статистично «обґрунтоване» з позиції моделі.

На рисунку 2 зображено концептуальну схему корекції за описаним алгоритмом.

Після запуску система формує два вихідні об’єкти:

1. виправлений текст (цільовий результат для подальшого аналізу/використання);

2. табличний звіт виправлень (індекс позиції токена, оригінал, запропонований кандидат, приріст LM-скору, джерело кандидата). Наявність такого звіту важлива для відтворюваності й якісного аналізу, оскільки дозволяє відокремити «що саме змінилося» від «чому модель вирішила, що це краще».

Далі розроблена система була застосування у посткорекції фрагмента рукописного тексту 1940 року. Метою експерименту є оцінка ефективності алгоритму у виправленні помилок базового HTR/OCR-розпізнавання та аналіз характеру як успішних, так і проблемних випадків.

Для експерименту було обрано уривок публіцистичного характеру, датований 1940 роком (див. рисунок 3). Текст містить орфографічні особливості довоєнного періоду (зокрема написання пролетаріят, Европа, цеї), що є додатковим викликом для автоматичної корекції.

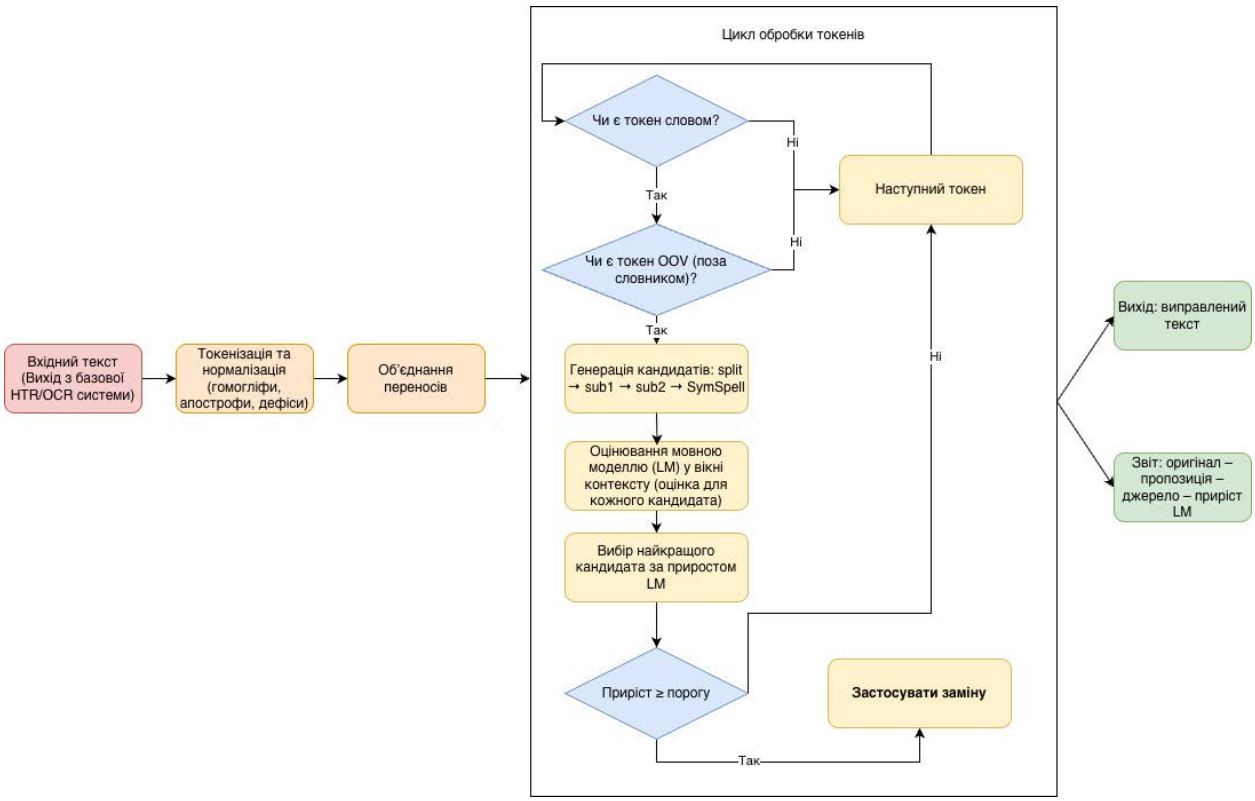

Рис. 2. Концептуальна схема алгоритму корекції

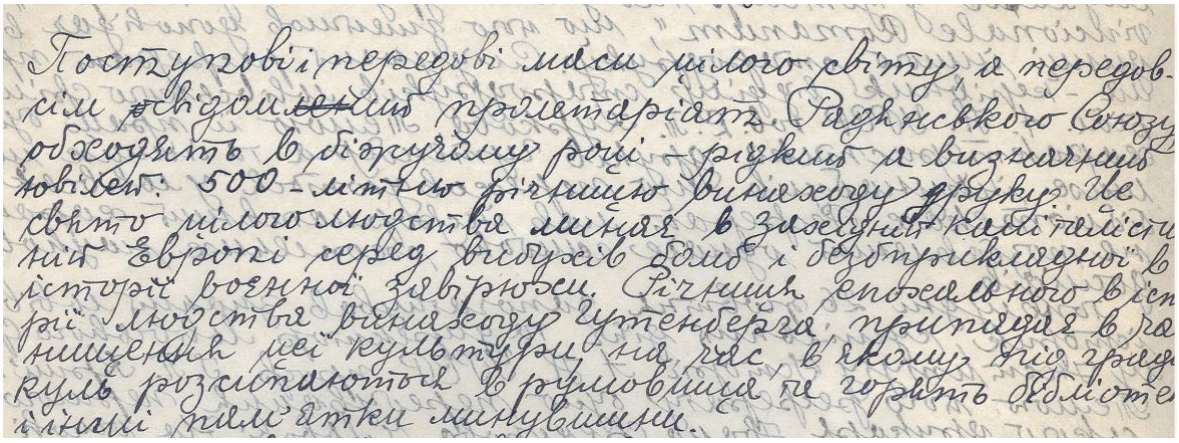

Рис. 3. Зображення рукопису для розпізнавання

Нижче наведено еталонний варіант тексту розпізнаний вручну: *«Поступові і передові маси цілого світу, а передовсім освідомлений пролетаріят Радянського Союзу обходять в біжучому році – рідкий і визначний ювілей: 500-літню річницю винаходу друку. Це свято цілого людства минає в західній капіталістичній Європі серед вибухів бомб і безприкладної в історії воєнної завірюхи. Річниця епохального в історії людства винаходу Гутенберга припадає в час нищення цеї культури, на час, в якому під градом куль розсипаються в румовища та горять бібліотеки і інші пам'ятки минувшини.»*

Далі, наведено текст, отриманий без застосування мовної моделі: *«Поступові і тередиві маси цілогс світу, а передов- сім рсвідом-лений пролетаріят Гадпнського Союзу рбходять в біжучому році – рідкий ивизничний ювілсй: 500-літню річницю винахооду дфуку. Це святоцілого людства миная в зихідніи капіталісти- ній Європі серед вибухів бомб і безприкладної в історії военної завірюхи. Річниця рпохального в історії людства винаходу Гутенберча припадая в ча нищення цеї культури, на час, в якому під гпадом куль розсипаються в румовища та чорять бібліоте і інші пам'ятки минувшини.»*

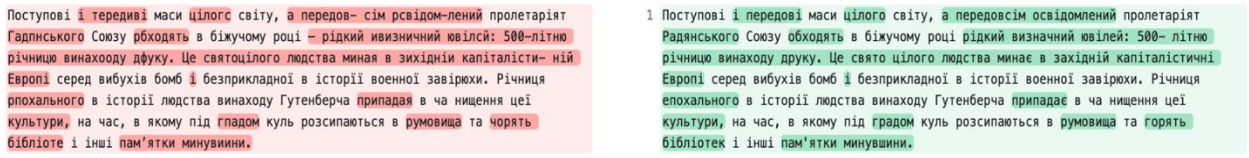

Рис. 4. Порівняння текстів до і після застосування алгоритму посткорекції

Характерні типи помилок:

– заміна графем (ювілсй, дфуку, рбходять);

– латинські омографи (і, а, Е);

– склеювання слів (святоцілого);

– дефісні розриви (передов- сім);

– пропуски/вставки літер (винахооду, минувиини);

– помилки у власних назвах (Гадпнського, Гутенберча).

Після застосування запропонованої системи (n-грамна LM + обмежена генерація кандидатів + SymSpell-індекс) отримано такий результат: *«Поступові і передові маси цілого світу, а передовсім освідомлений пролетаріят Радянського Союзу обходять в біжучому році рідкий визначний ювілей: 500- літню річницю винаходу друку. Це свято цілого людства минає в західній капіталістичні Європі серед вибухів бомб і безприкладної в історії военної завірюхи. Річниця епохального в історії людства винаходу Гутенберча припадає в ча нищення цеї культури, на час, в якому під градом куль розсипаються в румовища та горять бібліотек і інші пам'ятки минувшини.»*

На рисунку 4 зображено порівняння текстів після базового OCR та результату посткорекції.

Алгоритм здійснив 20 виправлень, що підтверджено звітом LM-reranking. Приклади виправлень наведено у таблиці 1.

Система продемонструвала високу ефективність у випадках:

– локальних орфографічних спотворень (edit distance ≤ 2);

– характерних OCR-помилок (гпадом → градом);

– латинсько-кириличних омографів;

– склеювання слів (святоцілого → свято цілого).

Частина помилок у досліджуваному фрагменті не була виправлена повністю або була скоригована лише частково. В таблиці 2 наведено приклади частково виправлених помилок і проблемних випадків.

Детальніший аналіз цих випадків дозволяє виокремити кілька системних причин.

По-перше, спостерігається морфологічна неоднозначність, коли мовна модель обирає граматично правдоподібну, але не тотожну еталону форму. Наприклад, у випадку «капіталістиній» система запропонувала форму «капіталістичні», тоді як еталон містить форму «капіталістичній». Обидва варіанти є лексично коректними, однак відрізняються граматичною формою. Оскільки n-грамна модель оцінює локальний контекст, вона оптимізує ймовірність послідовності токенів, а не їх точну відповідність еталонному тексту. У подібних випадках система демонструє лінгвістичну

Таблиця 1

Приклади успішних виправлень

Base OCR	Після алгоритму	Еталон	LM-gain
тередиві	передові	передові	2.4626
цілогс	цілого	цілого	7.2238
рсвідомлений	освідомлений	освідомлений	1.0761
Гадпнського	радянського	Радянського	5.6184
дфуку	друку	друку	2.1633
гпадом	градом	градом	9.1848
минувиини	минувшини	минувшини	1.7501

Таблиця 2

Випадки з частковою або некоректною корекцією

OCR	Після алгоритму	Еталон	Коментар
капіталістиній	капіталістичні	капіталістичній	морфологічна невідповідність
історії	історії	історії	складна графема
военної	военної	воєнної	історична орфографія
Гутенберча	гутенберг	Гутенберга	власна назва поза vocab
бібліоте	бібліотек	бібліотеки	помилка відмінка
в ча	в ча	в час	фрагментація токена

правдоподібність, але не гарантує повної морфологічної точності.

По-друге, вплив має історична орфографія тексту 1940 року. У фрагменті збережено форми на кшталт «пролетаріят», «Європі», «цеї», що відображають довоєнні правописні норми. Якщо такі варіанти неповністю представлені у корпусному словнику або core-vocab, система може або залишати їх без змін, або пропонувати сучасні відповідники. Це створює додатковий рівень складності, оскільки з формальної точки зору ці слова не є помилками, але відрізняються від сучасної норми, на якій переважно базується корпус.

По-третє, частина помилок пов'язана зі складними сегментаційними викривленнями, які фактично потребують багатотокенного аналізу. Наприклад, випадки типу «в ча» або «бібліотек» (замість «бібліотеки») виникають унаслідок поєднання пропуску літери та попереднього дефекту сегментації. Поточна архітектура системи працює на рівні окремого токена з обмеженим контекстним вікном і не виконує повноцінної синтаксичної реконструкції речення. Відповідно, багатокомпонентні помилки можуть залишатися частково невиправленими.

Нарешті, окрему категорію становлять власні назви, відсутні в core-vocab. Прикладом є варіант «Гутенберча», для якого модель пропонує форму «гутенберг» із нульовим приростом LM-оцінки. Через відсутність повного набору словоформ цієї власної назви у словнику система не має достатньо надійної кандидатної альтернативи та, відповідно, не здійснює впевненої корекції. Це демонструє залежність посткорекції від покриття словникових ресурсів і підкреслює необхідність спеціалізованих списків історичних і власних назв для підвищення точності системи.

Для кількісної оцінки результатів розпізнавання, було застосовано CER (Character Error Rate) і WER (Word Error Rate) – стандартні метрики якості розпізнавання тексту, що вимірюють мінімальну кількість редагувань, потрібних для перетворення результату системи на еталон. Обидві метрики базуються на відстані Левенштейна, де дозволені операції S (substitution – заміна), D (deletion – видалення), I (insertion – вставка). CER оцінює помилки на рівні символів (чутлива до пунктуації, пробілів, дефісів тощо), тоді як WER працює на рівні токенів-слів і краще відображає «читабельність» та семантичну коректність, але може «не бачити» дрібні орфографічні відхилення всередині слова.

У цьому експерименті CER/WER обчислено для двох системних виходів (базовий OCR та посткорекція) відносно еталону (Rice, 1996). Перед підрахунками виконано мінімальну технічну нормалізацію: уніфікація апострофа, зведення множинних пробілів до одного та обрізання крайових пробілів. Для WER токенізація виконувалася як виділення «слів» (послідовностей літер/цифр з можливими внутрішніми дефісами типу 500-літню).

CER:

Еталон має N = 509 символів.

– Базовий OCR:

Levenshtein_edits = 40

CER_OCR = 40 / 509 = 0.0786 ≈ 7.86%

– Результат алгоритму:

Levenshtein_edits = 11

CER_ALG = 11 / 509 = 0.0216 ≈ 2.16%

Результат розрахунку CER показує, посткорекція зменшила символьну помилковість приблизно у 3.6 раза (7.86% → 2.16%), тобто більшість дрібних OCR-артефактів (помилки літер, пропуски/зайві символи, частина злипань) була успішно усунута.

WER:

Еталон має N_w = 72 слів.

– Базовий OCR:

Word-level edits = 35

WER_OCR = 35 / 72 = 0.4861 ≈ 48.61%

– Результат алгоритму:

Word-level edits = 9

WER_ALG = 9 / 72 = 0.1250 = 12.50%

Результат розрахунку WER показує, що після посткорекції частка «помилкових слів» знизилась приблизно у 3.9 раза (48.61% → 12.50%). Це корелює з якісним аналізом: система відновила значну кількість ключових лексем (передові, радянського, обходять, ювілей, винаходу, друку, західній тощо), але залишилися помилки сегментації/пунктуації та окремі морфологічні/лексичні відхилення (капіталістичні Європі, історії, Гутенберча, бібліотек замість бібліотеки, втрата тире/дефісної структури в одному місці). Порівняння CER і WER метрик наведено в таблиці 3.

Отримані результати демонструють ефективність n-грамної мовної моделі як механізму контекстного ранжування кандидатів, проте подальше підвищення якості можливе шляхом переходу від ручної кандидатогенерації до підходів типу confusion-aware. Найбільш перспективним є використання n-best гіпотез, lattice-структур або символьних альтернатив, які надає сам розпізнавач. У такому випадку мовна модель не створює кандидати штучно, а здійснює повторне оцінювання вже ймовірнісно обґрунтованих варіантів, що суттєво зменшує ризик некоректних замін і знижує залежність від евристик.

Таблиця 3

Порівняння метрик CER і WER

	ED_char	CER (%)	ED_word	WER (%)
Базовий OCR / еталон	40	7.86	35	47.95
Посткорекція / еталон	11	2.16	10	13.70

Другим напрямом удосконалення є доменно-специфічна адаптація мовної моделі. У дослідженні використовувався корпус широкого покриття, однак для історичного публіцистичного тексту 1940-х років доцільною є або побудова LM на жанрово й хронологічно близькому підкорпусі, або застосування методів адаптації мовної моделі. Це дозволило б точніше враховувати характерну лексику, орфографічні особливості та стилістичні патерни періоду.

Перспективним також є залучення морфологічної інформації як додаткового обмеження при виборі кандидата. Морфологічна валідація могла б зменшити випадки надкорекції, коли система обирає граматично правдоподібну, але контекстуально неточну форму. Водночас така інтеграція потребує обережності, щоб не нівелювати історичні або варіантні словоформи, характерні для досліджуваного матеріалу.

Окремо варто виділити технічну нормалізацію переносів і дефісів. Частина помилок виникає не через мовні закономірності, а через особливості верстки та розбиття рядків у джерелі. Тому доцільно реалізувати ізольований модуль опрацювання переносів, який працює на рівні форматування, не втручаючись у мовну модель. Це дозволить зменшити кількість псевдо-OOV токенів без ризику лінгвістичної надкорекції.

Висновки. Проведений експеримент на одному історичному рукописному уривку продемонстрував, що корпусна n-грамна мовна модель може суттєво покращити результати базового OCR без зміни самого розпізнавача. Зниження CER і WER більш ніж утричі свідчить про ефективність контекстного reranking у виправленні типових символьних і лексичних помилок.

Експеримент підтвердив, що навіть класична статистична мовна модель залишається практично цінним інструментом для посткорекції, особливо у випадках, коли недоступні внутрішні ймовірнісні структури розпізнавача або нейронні моделі великого масштабу. Корпус виступає не лише джерелом словника, а насамперед джерелом розподілу контекстних ймовірностей, що дозволяє системі приймати узгоджені рішення.

Водночас результати окреслюють межі застосування підходу. Модель менш ефективна при складних сегментаційних помилках, відсутності власних назв у словнику, а також у випадках історичних орфографічних варіантів, недостатньо представлених у корпусі. Це свідчить про необхідність комбінування мовної моделі з більш глибокими лінгвістичними або ймовірнісними механізмами.

Таким чином, дослідження показує, що корпусний підхід у поєднанні з n-грамною мовною моделлю є дієвим і відносно простим для реалізації інструментом підвищення якості HTR/OCR. Його практична цінність полягає у можливості покращення результатів розпізнавання без повторного навчання розпізнавальної моделі, що особливо актуально для історичних текстів та обмежених обчислювальних ресурсів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Tikhonov A., Rabus A. Handwritten Text Recognition of Ukrainian Manuscripts in the 21st Century: Possibilities, Challenges, and the Future of the First Generic AI-based Model. *Kyiv-Mohyla Humanities Journal*. 2024. № 11. P. 226–247. DOI: 10.18523/2313-4895.11.2024.226-247.

2. Carbune V., Gonnet P., Deselaers T., Rowley H. A., Daryin A., Calvo M., Wang L.-L., Keysers D., Feuz S., Gervais P. Fast Multi-language LSTM-based Online Handwriting Recognition. *International Journal on Document Analysis and Recognition (IJDAR)*. 2020. Vol. 23, № 1. P. 1–15. DOI: 10.1007/s10032-020-00350-4.

3. Brodic D., Amelio A., Milivojevic Z. N. An Approach to the Analysis of the South Slavic Medieval Labels Using Image Texture. *arXiv preprint*. 2015. DOI: 10.48550/arXiv.1509.01978.

4. Баранецький Ю. Р., Кунанець Н. Е. Розпізнавання рукописного тексту: сучасні підходи та виклики. *Актуальні питання гуманітарних наук : міжвузівський збірник наукових праць молодих вчених Дрогобицького державного педагогічного університету імені Івана Франка*. 2025. Вип. 83, т. 1. С. 190–198. DOI: 10.24919/2308-4863/83-1-29.

5. Tarride S., Kermorvant C. Revisiting N-Gram Models: Their Impact in Modern Neural Networks for Handwritten Text Recognition. *arXiv preprint*. 2024. DOI: 10.48550/arXiv.2404.19317.

6. Шведова М., фон Вальденфельс Р., Старко В., Рисін А. Генеральний регіонально анотований корпус української мови (ГРАК) : вебресурс. URL: https://uacorpus.org/ (дата звернення: 18.12.2025).

7. Al-Masoudi A. F. R., Al-Obeidi H. S. R. Smoothing techniques evaluation of N-gram language model for Arabic OCR postprocessing. *Journal of Theoretical and Applied Information Technology*. 2015. Vol. 82, № 3. P. 432–439.

8. Fischer A. *Automatic Handwriting Recognition for Historical Documents : HisDoc Project Report*. HES-SO, 2020. 32 p. DOI: 10.1142/9789811203244-0005

9. Jung K., Kim N.-J., Ryu H. G., Lee H.-J. Enhancing ASR Performance through OCR Word Frequency Analysis: Theoretical Foundations. *arXiv preprint*. 2024. DOI: 10.48550/arXiv.2405.02995.

10. Johnson M. E., Vastrick T. W., Boulanger M., Schuetzner E. *Measuring the Frequency Occurrence of Handwriting and Hand-Printing Characteristics*. National Institute of Justice, Grant No. 250539. 2019. URL: https://www.ojp.gov/pdffiles1/nij/grants/250539.pdf (дата звернення: 28.12.2025).

11. Rose T. G., Evett L. J. Text Recognition using Collocations and Domain Codes. *Proceedings of the Workshop on Very Large Corpora*. 1993. P. 65–73.

12. Heafield K. KenLM: Faster and Smaller Language Model Queries. *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland : Association for Computational Linguistics, 2011. P. 187–197.

13. Brill E., Moore R. An Improved Error Model for Noisy Channel Spelling Correction. *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*. Hong Kong : ACL, 2000. P. 286–293. DOI: 10.3115/1075218.1075255.

14. Jurafsky D., Martin J. H. *Speech and Language Processing : an introduction to natural language processing, computational linguistics, and speech recognition*. 3rd ed. draft (Jan. 7, 2023). Stanford University, 2023. URL: https://web.stanford.edu/~jurafsky/slp3/old_jan23/ed3book_jan72023.pdf (дата звернення: 01.02.2026).

15. Garbe W. *1000x Faster Spelling Correction algorithm*. 2012. URL: https://seekstorm.com/blog/1000x-spelling-correction/ (дата звернення: 03.02.2026).

16. Navarro G. A guided tour to approximate string matching. *ACM Computing Surveys*. 2001. Vol. 33, № 1. P. 31–88. DOI: 10.1145/375360.375365.

17. Manning C. D., Schütze H. *Foundations of Statistical Natural Language Processing*. Cambridge, MA : MIT Press, 1999. 680 p. DOI: 10.1017/S1351324902212851

18. Norvig P. *How to Write a Spelling Corrector*. 2007. URL: https://norvig.com/spell-correct.html (дата звернення: 05.02.2026).

19. Ahmad F., Kondrak G. Learning a Spelling Error Model from Search Query Logs. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. 2005. P. 955–962. DOI: 10.3115/1220575.1220695.

20. Palmer D. D. Text Preprocessing. *Handbook of Natural Language Processing* / ed. N. Indurkhya, F. J. Damerau. 2nd ed. Boca Raton, FL : CRC Press, 2010. P. 9–30. DOI: 10.1201/9781420085938-10.

21. Boytsov L. Super-Linear Indices for Approximate Dictionary Searching. *Similarity Search and Applications (SISAP 2012)*. *Lecture Notes in Computer Science*. 2012. Vol. 7404. P. 162–176. DOI: 10.1007/978-3-642-32153-5_12.

22. Karch D., Luxen D., Sanders P. Improved Fast Similarity Search in Dictionaries. *String Processing and Information Retrieval (SPIRE 2010)*. *Lecture Notes in Computer Science*. 2010. Vol. 6393. P. 173–178. DOI: 10.1007/978-3-642-16321-0_16.

23. Shvedova M., Lukashevskyi A. *PluG (Pluperfect GRAC): Corpus of Old Ukrainian Texts*. URL: https://github.com/Dandelliony/pluperfect_grac (дата звернення: 12.02.2026).

24. Jumelet J., Haga T., Fourtassi A., et al. BabyBabelLM: A Multilingual Benchmark of Developmentally Plausible Training Data. *arXiv preprint*. 2025. DOI: 10.48550/arXiv.2510.10159.

25. Bird S., Klein E., Loper E. *Natural Language Processing with Python*. Sebastopol : O'Reilly Media, 2009. 504 p.

26. Ramirez-Orta J. A., Xamena E., Maguitman A. G., Milios E., Soto A. J. Post-OCR Document Correction with Large Ensembles of Character Sequence-to-Sequence Models. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2022. Vol. 36, № 10. P. 11192–11199. DOI: 10.1609/aaai.v36i10.21369.

27. Kolak O., Resnik P. OCR Error Correction Using a Noisy Channel Model. *Proceedings of the Human Language Technology Conference (HLT 2002)*. 2002. P. 122–178. DOI: 10.3115/1289189.1289208.

28. Kukich K. Techniques for automatically correcting words in text. *ACM Computing Surveys*. 1992. Vol. 24, № 4. P. 377–439. DOI: 10.1145/146370.146380.

29. Madarász G., Ligeti-Nagy N., Holl A., Váradi T. OCR Cleaning of Scientific Texts with LLMs. *Natural Scientific Language Processing and Research Knowledge Graphs (NSLP 2024)*. *Lecture Notes in Computer Science (LNAI)*. 2024. Vol. 14770. P. 49–58. DOI: 10.1007/978-3-031-65794-8_4.

30. Kumar R. H. *Spelling Correction to Improve Classification of Technical Error Reports : degree project*. Stockholm : KTH Royal Institute of Technology, 2019. 56 p.

31. Rice S. V. *Measuring the accuracy of page-reading systems*. Las Vegas : Information Science Research Institute, 1996. 25 p. DOI: 10.25669/hfa8-0cqv

REFERENCES

1. Tikhonov A., Rabus A. Handwritten Text Recognition of Ukrainian Manuscripts in the 21st Century: Possibilities, Challenges, and the Future of the First Generic AI-based Model. Kyiv-Mohyla Humanities Journal. 2024. No. 11. 226–247. DOI: 10.18523/2313-4895.11.2024.226-247.

2. Carbune V., Gonnet P., Deselaers T., Rowley H. A., Daryin A., Calvo M., Wang L.-L., Keysers D., Feuz S., Gervais P. Fast Multi-language LSTM-based Online Handwriting Recognition. *International Journal on Document Analysis and Recognition (IJDAR)*. 2020. 23(1). 1–15. DOI: 10.1007/s10032-020-00350-4.

3. Brodic D., Amelio A., Milivojevic Z. N. An Approach to the Analysis of the South Slavic Medieval Labels Using Image Texture. *arXiv preprint*. 2015. DOI: 10.48550/arXiv.1509.01978.

4. Baranetskyi Yu. R., Kunanets N. E. Rozpiznavannia rukopysnoho tekstu: suchasni pidkhody ta vyklyky [Handwritten text recognition: modern approaches and challenges]. *Aktualni pytannia humanitarnykh nauk: mizhvuzivskyi zbirnyk naukovykh prats molodykh vchenykh Drohobytskoho derzhavnoho pedahohichnoho universytetu imeni Ivana Franka*. 2025. 83(1). 190–198. DOI: 10.24919/2308-4863/83-1-29. [in Ukrainian].

5. Tarride S., Kermorvant C. Revisiting N-Gram Models: Their Impact in Modern Neural Networks for Handwritten Text Recognition. *arXiv preprint*. 2024. DOI: 10.48550/arXiv.2404.19317.

6. Shvedova M., fon Valdenfels R., Starko V., Rysin A. Heneralnyi rehionalno anotovanyi korpus ukrainskoi movy (HRAK) [General Regionally Annotated Corpus of Ukrainian (GRAC)]. Web resource. [in Ukrainian].
7. Al-Masoudi A. F. R., Al-Obeidi H. S. R. Smoothing techniques evaluation of N-gram language model for Arabic OCR postprocessing. *Journal of Theoretical and Applied Information Technology*. 2015. 82(3). 432–439.
8. Fischer A. *Automatic Handwriting Recognition for Historical Documents: HisDoc Project Report*. HES-SO, 2020. 32 p. DOI: 10.1142/9789811203244-0005.
9. Jung K., Kim N.-J., Ryu H. G., Lee H.-J. Enhancing ASR Performance through OCR Word Frequency Analysis: Theoretical Foundations. *arXiv preprint*. 2024. DOI: 10.48550/arXiv.2405.02995.
10. Johnson M. E., Vastrick T. W., Boulanger M., Schuetzner E. *Measuring the Frequency Occurrence of Handwriting and Hand-Printing Characteristics*. National Institute of Justice, Grant No. 250539. 2019.
11. Rose T. G., Evett L. J. Text Recognition using Collocations and Domain Codes. *Proceedings of the Workshop on Very Large Corpora*. 1993. 65–73.
12. Heafield K. KenLM: Faster and Smaller Language Model Queries. *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, 2011. 187–197.
13. Brill E., Moore R. An Improved Error Model for Noisy Channel Spelling Correction. *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*. Hong Kong: ACL, 2000. 286–293. DOI: 10.3115/1075218.1075255.
14. Jurafsky D., Martin J. H. *Speech and Language Processing: an introduction to natural language processing, computational linguistics, and speech recognition*. 3rd ed. draft (Jan. 7, 2023). Stanford University, 2023.
15. Garbe W. 1000x Faster Spelling Correction algorithm. 2012.
16. Navarro G. A guided tour to approximate string matching. *ACM Computing Surveys*. 2001. 33(1). 31–88. DOI: 10.1145/375360.375365.
17. Manning C. D., Schütze H. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999. 680 p. DOI: 10.1017/S1351324902212851.
18. Norvig P. How to Write a Spelling Corrector. 2007.
19. Ahmad F., Kondrak G. Learning a Spelling Error Model from Search Query Logs. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. 2005. 955–962. DOI: 10.3115/1220575.1220695.
20. Palmer D. D. Text Preprocessing. In: Indurkhya N., Damerau F. J. (eds.). *Handbook of Natural Language Processing*. 2nd ed. Boca Raton, FL: CRC Press, 2010. 9–30. DOI: 10.1201/9781420085938-10.
21. Boytsov L. Super-Linear Indices for Approximate Dictionary Searching. In: *Similarity Search and Applications (SISAP 2012)*. *Lecture Notes in Computer Science*. 2012. 7404. 162–176. DOI: 10.1007/978-3-642-32153-5_12.
22. Karch D., Luxen D., Sanders P. Improved Fast Similarity Search in Dictionaries. In: *String Processing and Information Retrieval (SPIRE 2010)*. *Lecture Notes in Computer Science*. 2010. 6393. 173–178. DOI: 10.1007/978-3-642-16321-0_16.
23. Shvedova M., Lukashevskyi A. PluG (Pluperfect GRAC): Corpus of Old Ukrainian Texts.
24. Jumelet J., Haga T., Fourtassi A., et al. BabyBabelLM: A Multilingual Benchmark of Developmentally Plausible Training Data. *arXiv preprint*. 2025. DOI: 10.48550/arXiv.2510.10159.
25. Bird S., Klein E., Loper E. *Natural Language Processing with Python*. Sebastopol: O'Reilly Media, 2009. 504 p.
26. Ramirez-Orta J. A., Xamena E., Maguitman A. G., Milios E., Soto A. J. Post-OCR Document Correction with Large Ensembles of Character Sequence-to-Sequence Models. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2022. 36(10). 11192–11199. DOI: 10.1609/aaai.v36i10.21369.
27. Kolak O., Resnik P. OCR Error Correction Using a Noisy Channel Model. *Proceedings of the Human Language Technology Conference (HLT 2002)*. 2002. 122–178. DOI: 10.3115/1289189.1289208.
28. Kukich K. Techniques for automatically correcting words in text. *ACM Computing Surveys*. 1992. 24(4). 377–439. DOI: 10.1145/146370.146380.
29. Madarász G., Ligeti-Nagy N., Holl A., Váradi T. OCR Cleaning of Scientific Texts with LLMs. In: *Natural Scientific Language Processing and Research Knowledge Graphs (NSLP 2024)*. *Lecture Notes in Computer Science (LNAI)*. 2024. 14770. 49–58. DOI: 10.1007/978-3-031-65794-8-4.
30. Kumar R. H. *Spelling Correction to Improve Classification of Technical Error Reports: degree project*. Stockholm: KTH Royal Institute of Technology, 2019. 56 p.
31. Rice S. V. *Measuring the accuracy of page-reading systems*. Las Vegas: Information Science Research Institute, 1996. 25 p. DOI: 10.25669/hfa8-0cqv.

Дата першого надходження статті до видання: 25.02.2026
Дата прийняття статті до друку після рецензування: 30.03.2026
Дата публікації (оприлюднення) статті: 22.04.2026