Issue 18, part 1, 2025

https://doi.org/10.23939/sisn2025.18.043

УДК 004.9

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ КОРЕКЦІЇ ПОМИЛОК В УКРАЇНОМОВНИХ ТЕКСТАХ З ВИКОРИСТАННЯМ МЕТОДІВ МАШИННОГО НАВЧАННЯ

Ростислав Федчук[1], Вікторія Висоцька[2]

[1, 2] Національний університет «Львівська політехніка»,
кафедра інформаційних систем та мереж, Львів, Україна
[1] E-mail: rostyslav.b.fedchuk@lpnu.ua, ORCID: 0009-0002-6669-0369
[2] E-mail: victoria.a.vysotska@lpnu.ua, ORCID: 0000-0001-6417-3689

Актуальність дослідження зумовлена зростаючою потребою в автоматизації процесів аналізу та корекції текстів, зокрема для україномовного контенту, який відзначається багатством морфологічної і синтаксичної структури. Через широкий спектр помилок, що можуть виникати у текстах, від орфографічних до контекстуальних, існує нагальна потреба у створенні систем, здатних точно ідентифікувати помилки та пропонувати їх коректні виправлення. Специфіка української мови, включаючи складність її граматики та багатогранність, потребує адаптації моделей машинного навчання до локальних особливостей. Метою дослідження є розроблення математичної моделі системи підтримки прийняття рішень для ідентифікації та корекції помилок в україномовних текстах. Завдання включає як формалізацію та математичне описання процесу опрацювання текстів, так і побудову моделі з орієнтацією на задачі класифікації та генерації тексту. Особлива увага приділена ефективному врахуванню специфічних для української мови структурних особливостей із метою підвищення точності та продуктивності системи. Метод дослідження базуються на побудові математичної моделі корекції помилок, яка представлена як задача генерації тексту із врахуванням контексту. У дослідженні було використано статистичні методи та підходи машинного навчання. Особливу увагу присвячено формуванню навчальної вибірки, в якій поєднано тексти з реальними та штучними помилками для забезпечення збалансованого навчального процесу. У модулі корекції включено механізми генерації, що базуються на контекстуальних моделях, здатних передбачати правильне виправлення для помилкових токенів. Математично обґрунтовано підходи до векторизації текстів, враховуючи особливості морфології та синтаксису української мови. Побудована модель є універсальною основою для створення інтелектуальних систем автоматичного редагування україномовного тексту. У результаті проведеного дослідження сформульовано й математично обґрунтовано підходи до побудови моделі корекції помилок в україномовних текстах. Основним результатом стало створення інтегрованої системи, яка використовує контекстуальну інформацію для забезпечення високої точності розпізнавання помилок і їх виправлення. Застосовані математичні методи охоплюють ймовірнісні підходи та векторне представлення токенів, що дозволяє адаптувати систему до особливостей української мови з її високою морфологічною та синтаксичною складністю. Сфор-

мована основа моделі створює можливості для масштабування та подальшого використання у практичних завданнях, таких як автоматичне редагування текстів або підвищення якості контенту в україномовному середовищі.

Ключові слова – корекція помилок, NLP, генерація тексту, машинне навчання, українська мова, обробка тексту.

Постановка проблеми

У сучасну епоху цифрових технологій опрацювання природної мови (NLP) набуває особливого значення, оскільки мільйони текстів щоденно створюються, поширюються й аналізуються автоматизованими системами. Українська мова, попри свою багатогранність, складність її граматики, досі залишається недостатньо охопленою у більшості комерційних та відкритих рішень, спрямованих на автоматичне редагування тексту. Більшість існуючих систем граматичної корекції (GEC) фокусуються переважно на англомовному контенті, залишаючи україномовних користувачів із обмеженим вибором інструментів (Bryant et at., 2023). Це створює реальну потребу у розробленні новітніх методів і моделей, здатних не лише виявляти, але й коригувати широкий спектр помилок у текстах українською мовою – від простих орфографічних до складних синтаксичних і стилістичних (Huang & Fan, 2025). Сучасні досягнення у сфері машинного навчання, зокрема використання трансформерних архітектур і глибокого навчання, відкривають нові можливості для створення високоточних систем корекції (Kholodna & Vysotska, 2023). У цьому контексті актуальним є побудова математично обґрунтованої моделі, здатної ефективно працювати в умовах обмежених ресурсів, характерних для української мови.

Проблема виправлення помилок у тексті належить до категорії задач GEC, які включають дві основні підзадачі ідентифікація та корекція помилок.

Одним із розповсюджених викликів є різноманіття помилок, які можуть виникати у текстах. Орфографічні помилки є відносно прямолінійними для автоматичного опрацювання, тоді як граматичні, пунктуаційні та навіть стилістичні помилки вимагають детального аналізу контексту та взаємодії між словами в межах одного речення або навіть абзацу. Наприклад, у реченні "Мама купив хліб" помилка ("купив" замість "купила") потребує моделювання узгодження між підметом і присудком. Виявлення таких складних залежностей та створення механізмів їх виправлення залишаються відкритим завданням.

Крім того, задача ускладнюється відсутністю великих відкритих корпусів навчальних даних для української мови, які б забезпечили достатню якість навчання моделей машинного навчання. Домінування англомовного контенту в галузі NLP створює нерівномірний розподіл ресурсів і методологій, що потребує адаптації існуючих технологій до специфіки української мови.

В основі задачі GEC лежать два ключові етапи, які виконуються послідовно, ідентифікація і корекція помилок. Етап ідентифікації помилок відповідає за визначення помилкових токенів у тексті та типу помилки, яку вони містять. Процес роботи на цьому етапі включає аналіз кожного токена та оцінку ймовірності наявності помилки, враховуючи контекст оточуючих слів.

Для успішного вирішення задачі GEC необхідна система, здатна інтегрувати процеси ідентифікації та корекції у єдиний модельно-орієнтований підхід і вирішити виклики української мови, такі як контекстна багатозначність, морфологічний розподіл (необхідно враховувати 7 відмінків українських слів, їх число, рід, форму), синтаксична складність, дефіцит навчальних корпусів українською мовою.

Аналіз останніх досліджень та публікацій

На сьогодні автоматизовані інструменти для виправлення текстів відіграють ключову роль у виявленні та усуненні орфографічних, граматичних, пунктуаційних і стилістичних неточностей (Fedchuk & Vysotska, 2024). Серед найбільш ефективних і поширених рішень – програмні застосунки та онлайн-платформи, які часто інтегруються з текстовими редакторами або працюють на базі алгоритмів машинного навчання.

Одним із найвідоміших сервісів у цій галузі є Grammarly, що спеціалізується на автоматичній перевірці текстів (Grammrly Inc, n.d.). Втім, наразі його функціональність обмежується переважно англійською мовою – перевірка граматики, орфографії та стилістики українських текстів поки не підтримується. Попри це, компанія робить кроки в напрямі підтримки української мови в межах комп'ютерної лінгвістики, зокрема, вона створила (Grammarly, n.d.) і надала відкритий доступ до анотованого корпусу для задач граматичної корекції (GEC) української мови (Syvokon & Nahorna, 2021), який налічує близько 34 000 речень. Цей корпус є цінним ресурсом для наукових досліджень, тестування алгоритмів і навчання моделей, орієнтованих на виправлення граматичних помилок в україномовних текстах.

GPTTools.ai (Brovinska, 2024) – це український десктопний застосунок на основі штучного інтелекту GPT-4, створений для ефективної роботи з текстами українською та ще понад 70 мовами. Інструмент надає можливості генерування текстів, редагування, корекції стилістичних, орфографічних та граматичних помилок, а також перекладу. Ключовою особливістю є підтримка індивідуальних промптів – користувачі можуть створювати власні шаблони для автоматизації рутинних завдань. Програма підходить для студентів, науковців, копірайтерів і всіх, хто працює з великими обсягами текстової інформації, забезпечуючи зручність і точність опрацювання даних.

LanguageTool, що підтримує багатомовний формат, включно з українською мовою, на сьогодні є одним із найбільш функціональних інструментів для перевірки текстів (LanguageTool, n.d.). LanguageTool працює на основі правил і на сьогодні налічує 1219 правил (LanguageTool Community, n.d.). Ця платформа доступна як онлайн-інструмент, браузерне розширення або десктопний додаток. Вона здатна працювати з граматичними, орфографічними та контекстуальними помилками, пропонуючи рекомендації для їх виправлення. Можлива інтеграція LanguageTool через LanguageTool API, яка підтримує українську мову і має назву NLP-uk (NLP UK, n.d.). Через цю інтеграцію користувач отримує доступ до нормалізації, токенізації, лематизації, POS-тегування текстів і також засоби для вирішення проблеми двозначностей в українській мові.

Текстові процесори Microsoft Word та Google Docs також мають функцію перевірки текстів, включаючи базову підтримку української мови. Microsoft Word дозволяє виправляти орфографічні помилки шляхом використання словникових баз і лінгвістичних правил. Google Docs пропонує аналогічну базову перевірку, але її функціонал є менш розвиненим. Для комплексного аналізу тексту ці платформи можуть використовувати додаткові інтеграції, такі як LanguageTool.

Формулювання цілі статті

Корекція помилок – це процес автоматичного виявлення та усунення орфографічних, граматичних, пунктуаційних і стилістичних похибок у тексті з урахуванням лінгвістичного контексту. Ця задача належить до напряму корекції граматичних помилок (Grammatical Error Correction, GEC) і охоплює дві ключові підзадачі – ідентифікацію та виправлення помилкових токенів.

Використання методів машинного навчання для корекції помилок дозволяє моделювати складні граматичні та контекстуальні залежності між словами в тексті. Застосування трансформерних архітектур та нейронних мовних моделей забезпечує здатність системи адаптуватися до морфологічної та синтаксичної складності української мови, що є особливо важливим у низькоресурсному мовному середовищі.

Метою дослідження є розроблення математичної моделі системи підтримки прийняття рішень для автоматичної ідентифікації та корекції помилок в україномовних текстах на основі сучасних методів машинного навчання.

Об'єктом дослідження є процеси автоматизованого опрацювання текстової інформації українською мовою, зокрема виявлення та виправлення мовних помилок.

Предметом дослідження є математичні моделі, алгоритми та програмні засоби, що реалізують задачі ідентифікації й корекції помилок у текстах українською мовою із використанням машинного навчання та нейромережевих технологій.

Виклад основного матеріалу

Модуль ідентифікації помилок виконує аналіз тексту (Starko, Rysin & Shvedova, 2021) для виявлення некоректних слів, граматичних, орфографічних і стилістичних помилок. Його основними завданнями є (Lytvyn, Pukach, Vysotska, Vovk & Kholodna, 2023):

- Сегментація тексту – розбиття вхідного тексту на окремі речення.
- Видалення стоп-слів – видалення знаків пунктуації, а також інших мовних конструкцій, які не несуть змістового навантаження і лише будуть заповільнювати ідентифікацію помилок.
- Токенізація тексту – розбиття вхідного тексту на окремі слова або фрази для подальшого аналізу.
- Лематизація, стеммінг та морфологічний аналіз – визначення основної форми слів, їх частини мови та граматичних характеристик.
- Класифікація слів на правильні та потенційно помилкові – застосування методів машинного навчання для оцінки коректності слів у контексті речення.
- Контекстний аналіз – оцінка ймовірності помилки на основі навколишніх слів (n-грамні моделі, трансформери, правила).

На основі цих етапів формується список потенційних помилок, який передається до модуля корекції. Після ідентифікації помилок система повинна запропонувати відповідні виправлення. Основні функції модуля корекції помилок наступні: генерація виправлень, контекстна перевірка, ранжування варіантів виправлень, розрахунок ступеня впевненості моделі щодо виправлення.

Залежно від налаштувань система може або автоматично вносити виправлення, або пропонувати користувачеві вибрати найбільш доречний варіант. У деяких випадках можливий варіант інтерактивного навчання користувачем.

Модуль машинного навчання є критичним для адаптації та покращення системи. Його основні функції включають навчання моделей, оновлення параметрів та збір зворотного зв'язку.

У процесі побудови математичної моделі системи підтримки прийняття рішень (СППР) для ідентифікації та корекції помилок в україномовних текстах важливим етапом є формалізація вхідних і вихідних параметрів. Це дозволяє чітко визначити, які дані система опрацьовує та які результати вона повинна генерувати.

Вхідні дані формалізуються як:

$$X = \{x_1, x_2, \dots, x_n\},$$

де X – вхідний текстовий масив, що складається з n токенів (слів, символів, речень); x_i – окремий токен або слово у тексті.

Результатом роботи системи є виправлений текст та метрики якості виправлення. Формально вихідні дані можна подати у вигляді множини:

$$\hat{X} = \{\widehat{x_1}, \widehat{x_2}, \dots, \widehat{x_n}\},$$

де $\hat{X}$ – виправлений текст; $\widehat{x_i}$ – виправлений варіант слова або токена x_i.

До вихідних параметрів відносяться виправлений текст**,** перелік знайдених помилок та метрики оцінки корекції, що включає точність, повноту, F1-міру та оцінку довіри.

Математична модель корекції помилок в україномовних текстах

Корекція помилок в україномовних текстах може бути сформульована як задача генерації тексту, де модель отримує на вході текст із помилками, аналізує контекст, визначає правильний варіант замість помилкового фрагмента і повертає виправлений текст.

Нехай вхідний текст подається як послідовність токенів

$$X = \{x_1, x_2, \dots, x_N\} \text{ та послідовність міток } Y = \{y_1, y_2, \dots, y_N\},$$

де (x_i) – окремий токен, а $y_i \in C = \{c_0, c_1, \dots, c_K\}$ – тип помилки для відповідного токена (x_i): c_0 – "без помилки", $\{c_0, c_1, \dots, c_K\}$: типи помилок (наприклад, орфографічна, граматична, пунктуаційна).

Завдання корекції полягає у трансформації тексту (X) у текст

$$\hat{X} = \{\widehat{x_1}, \widehat{x_2}, \dots, \widehat{x_N}\},$$

де кожен виправлений токен ($\widehat{x_i}$) є точним представленням без помилок, коректно узгодженим з контекстом, або незмінний токен, якщо помилка відсутня ($y_i = c_0$).

Основна мета моделі – генерація тексту ($\widehat{x_i}$), який відповідає граматичним, стилістичним і лексичним правилам української мови та узгоджений із контекстом збереженого фрагмента тексту.

З математичної точки зору, корекція помилки для кожного токена (x_i) в тексті може бути описана як задача пошуку оптимального кандидата ($\widehat{x_i}$):

$$(\widehat{x_i}) = \arg\max_{x_i' \in \mathcal{V}} P\,(x_i' | x_i, context),$$

де $\mathcal{V}$ – словникова множина всіх допустимих токенів для української мови; $P(x_i'|x_i, context)$ – умовна ймовірність того, що токен x_i' є правильним виправленням x_i, враховуючи контекст. Для кожного токена x_i, модель оцінює ймовірність різних варіантів виправлення x_i', беручи до уваги помилку та контекст навколо токена. Ймовірність моделюється таким чином:

$$P(x_i'|x_i, x_{i-1}, x_{i+1}, \dots, x_{i-m}, x_{i+m}) = f_{cor}(x_i, context),$$

де $context$ – це всі довколишні токени у вікні ширини ($2m$); а f_{cor} – функція корекції, реалізована за допомогою моделі генерації тексту. Корекція помилок вимагає врахування семантичного і граматичного контексту. Семантичний контекст допомагає врахувати узгодженість із сусідніми словами, в той час, як граматичний контекст гарантує, що слово узгоджується з родом, числом, часом, відмінком тощо.

Алгоритм корекції помилок працює на основі вхідних даних, отриманих із модуля ідентифікації, який визначив потенційно помилкові токени та класифікував їх за типами помилок. Корекція помилок полягає у знаходженні оптимальних замін для помилкових токенів, узгоджених із контекстом, граматичними правилами та стилістичними особливостями української мови, і у формуванні виправленого тексту.

Опис алгоритму корекції помилок:

1. Ініціалізація структури даних. Це є початковим етапом, на якому модуль корекції повинен отримати вхідні токени (X) і мітки помилок (Y), а також створити порожній список для виправлених токенів: $\hat{X} = [\,]$.

2. Опрацювання кожного токена. На цьому етапі для кожного (x_i) у тексті (X) виконується перевірка мітки помилки (y_i). Якщо ($y_i = c_0$), тобто токен коректний, тоді ($\widehat{x_1} = x_i$), і токен додається до списку ($\hat{X}$) без змін. Якщо ($y_i \neq c_0$), тобто токен ідентифіковано як помилковий, тоді ініціюється процес виправлення відповідно до типу помилки (y_i).

3. Генерація кандидатів для виправлення. Для кожного помилкового токена (x_i) визначається множина можливих варіантів виправлення ($V = \{v_1, v_2, \dots, v_M\}$) із використанням словника української мови, граматичного аналізу, а також мовної моделі. Генерація правильного тексту може реалізовуватися декількома підходами: Greedy Search, Beam Search та Sampling with Temperature. Greedy Search модель послідовно вибирає найбільш ймовірний варіант для кожного токена, але цей метод не завжди гарантує глобальну оптимальність. Під час Beam Search розглядається кілька найімовірніших варіантів, з яких обирається найкращий на основі глобальної оцінки контексту. Sampling with Temperature використовує генерацію варіантів коригується за параметрами "температури", щоб збільшити варіативність виправлень.

Для задачі генерації правильного тексту цільова функція моделі корекції оптимізує ймовірність правильного виправленого тексту. Для задачі багаторівневого оцінювання використовується крос-ентропійна функція втрат:

$$Loss = -\sum_{i=1}^{N} \log P(\widehat{x_i} \mid x_i, context).$$

Для визначення орфографічних помилок генеруються варіанти, максимально схожі на (x_i) за принципом порівняння відстані Левенштейна. Наприклад, для токена "книгі" множина варіантів може бути: V = { {"книга"}, {"книги"}, {"книзі"} }. Для визначення граматичних помилок з узгодженням із правилами української граматики система використовує морфологічний аналіз. Це обмежує перелік кандидатів лише словами, які відповідають потрібному формату. Мовні моделі, у свою чергу, визначають контекстно-залежні кандидати за допомогою попередньо натренованих моделей, які враховують контекст сусідніх слів під час генерації виправлень.

4. Оцінка кандидатів. Для кожного з кандидатів ($y_i \in V$) система обчислює ймовірність відповідності контексту. Оцінювання здійснюється за допомогою мовних моделей (BERT, GPT) або правил граматики. Моделі обчислюють, яка заміна (y_i) є найкращою у контексті всього речення. У реченні "Я люблю читати книгі", ймовірність P({"книги" | "Я люблю читати"}) буде вищою, ніж для "книзі". Граматичні правила вираховують додаткові обмеження на вибір слова.

5. Вибір найкращого кандидата. Для помилкового токена (x_i) обирається кандидат (v_i^*) із максимальною оцінкою ймовірності:

$$v_i^* = arg \max_{v_j \in V_i} P(v_j \mid context).$$

Для "книгі" значення (v_i^* = {"книги"}), оскільки це найбільш вірогідний варіант у заданому контексті.

6. Корекція пунктуаційних помилок. Якщо тип помилки (y_i) позначає пунктуаційне порушення, то замість токена (x_i) вставляється правильний розділовий знак, визначений за правилами синтаксису. Для тексту "Не знаю як пояснити" система виправить його у "Не знаю, як пояснити".

7. Формування фінального тексту. Після опрацювання кожного токена формується вихідний текст. Якщо токен не змінювався, він додається до фінального списку $\hat{X}$ у вихідному вигляді. Якщо токен був змінений, до списку $\hat{X}$ додається виправлений варіант v_i^*.

8. Перевірка узгодженості. Для усього речення виконується фінальна перевірка узгодженості між токенами. Поводиться перевірка граматичної правильності узгодження між підметом і присудком, іменниками та прикметниками, дієсловами та їхніми аргументами; перевірка семантичної змістовної відповідності всієї конструкції.

9. Повернення виправленого тексту. Система повертає виправлений текст $\hat{X}$, а також може надати список внесених змін із вказівкою типу кожної помилки.

Для візуалізації алгоритму і процесів модуля корекції помилок в україномовних текстах були побудовані діаграми DFD (рис. 1-2).

Приклад роботи алгоритму з вхідним текстом "Я люблю читати книгі й розкажу друзі про книгу.":

Мітки від модуля ідентифікації:

{{«Я»: без помилок}, {«люблю»: без помилок}, {читати: без помилок},
{книгі: орфографічна}, {й: без помилок}, {розкажу: без помилок},
{друзі: граматична}, {про: без помилок},
{"книгу": без помилок} }. Виправлення: "книгі" → "книги", "друзі" → "друзям".

Вихідний текст: "Я люблю читати книги й розкажу друзям про книгу."

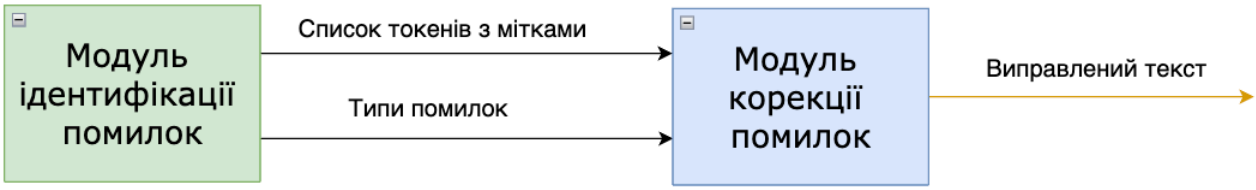

Рис. 1. Діаграма DFD рівень 0 модуля корекції помилок

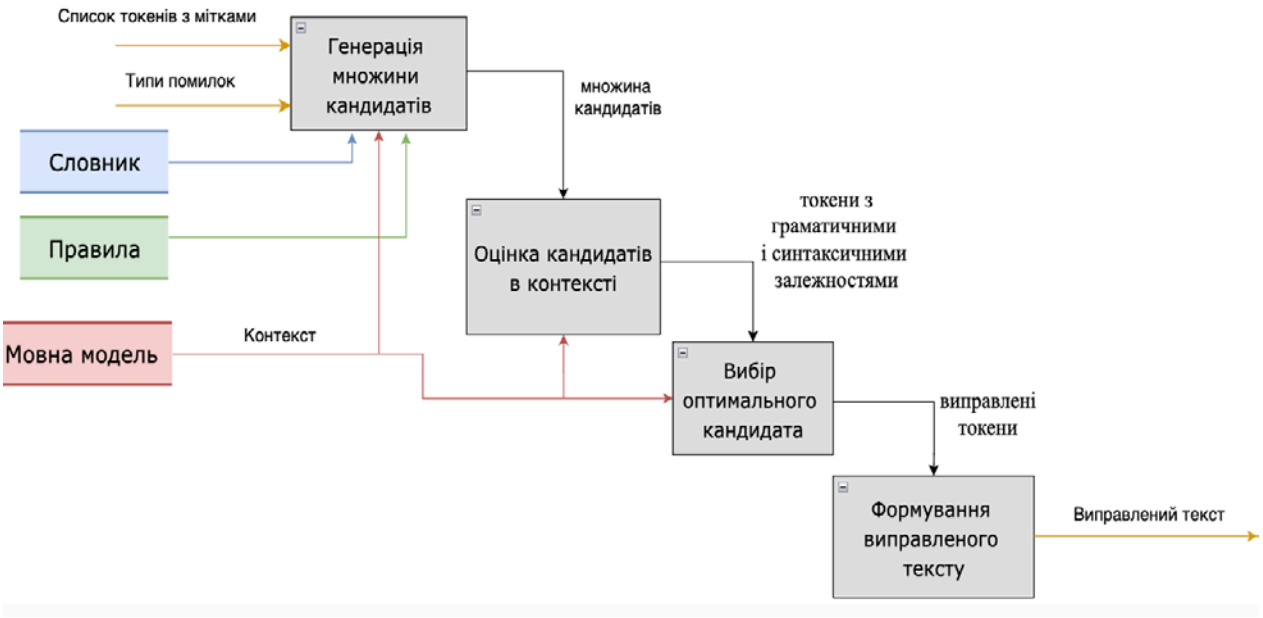

Рис. 2. Діаграма DFD рівня 1 для модуля корекції помилок.

Математична модель машинного навчання вхідних даних

Текст у його вихідній формі не є придатним для опрацювання алгоритмами машинного навчання, які працюють із числовими даними. Тому основне завдання цього етапу – перетворити текст у формат, що дозволяє використовувати його для побудови моделей, здатних ефективно навчатися.

Усі операції представлення тексту пов'язані із збереженням семантичної та синтаксичної інформації. Тому вибір техніки представлення текстів є вирішальним для якості результатів роботи системи машинного навчання.

Перед початком формування векторних подань тексту виконується процес токенізації, яка має враховувати особливості української граматики, наприклад опрацювання складних слів, часток і лапок. Після токенізації кожен токен необхідно перетворити у числовий формат, придатний для машинного опрацювання.

Залежно від завдання та методів, які використовуються для опрацювання тексту, кожен токен може бути представлений у різних форматах. Номер у словнику відповідає конкретному токену (integer encoding), як багатовимірне представлення (One-Hot Encoding), де кожен токен представлений як вектор із розмірністю, рівною числу унікальних токенів у корпусі. Багатовимірний числовий вектор описує характеристики токена (word embeddings). Два перші методи не масштабуються для великих корпусів та не враховують взаємозв'язків між токенами, що робить їх не найкращим варіантом для вибору.

Word embeddings – це багатовимірні вектори, які враховують семантичні та синтаксичні зв'язки між словами й побудовані на основі статистичних моделей. Найпоширеніші методи Word2Vec, FastText та BERT. Word2Vec – статистичний метод побудови векторів через тренування на текстових корпусах. Вектори "схожих" слів ближчі один до одного в багатовимірному просторі. Наприклад: $\{\text{собака}\} \approx [0.2, 0.8, -0.3], \{\text{вовк}\} \approx [0.1, 0.7, -0.4]$. FastText – Розширення Word2Vec, яке враховує всередині токена підсловесні елементи (морфеми). Це важливо для української мови через високу морфологічну варіативність. BERT, в свою чергу, генерує контекстні вектори, які залежать від позиції слова у реченні. Наприклад: "сльози" у реченні "сльози на очах" матиме інший вектор, ніж "сльози природи".

Для коректного представлення текстів необхідно враховувати морфологічну складність через лематизацію та морфологічного аналізу. Для україномовного контенту важливо також адаптувати представлення до неформальних конструкцій, присутніх у розмовній і художній мові (наприклад, "шо" замість "що").

Кожен токен (x_i) з тексту (X) можна представити як вектор (W):

$$W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}, \quad w_i = f_{embed}(x_i \mid context),$$

де (f_{embed}) – функція побудови векторів, яка враховує характеристики токена й контекст.

Навчальна вибірка представляє дані, на яких модель буде вчитися розпізнавати помилки та пропонувати виправлення. Для роботи з україномовними текстами важливо врахувати не лише загальні принципи збору даних, але й специфіку української мови, яка включає багату морфологію, складну синтаксисно-граматичну структуру, а також наявність регіоналізмів, запозичень і розмовної мови.

Головними задачами цього етапу є створення репрезентативного корпусу текстів, що охоплюють різні аспекти мови - формальний, розмовний і художній стилі; введення штучних помилок, що моделюють реальні типи помилок у тексті, або використання корпусів із помилками, позначеними вручну; балансування даних, щоб уникнути домінування одного типу помилки й забезпечити рівномірне навчання моделі; побудова структури даних, яка поєднує оригінальний текст, модифікований текст із помилками, мітки помилок і правильний варіант тексту.

Для забезпечення якісного навчання моделі потрібно створити збалансовану вибірку, де кожен тип помилки повинен мати достатню кількість прикладів і водночас співвідношення некоректних текстів до коректних не повинно бути надмірно великим (40 % помилкових текстів до 60 % коректних). Для створення якісної вибірки тексти мають бути анотовані вручну. Для цього кожна помилка помічається відповідним типом, вказуються точні позиції помилкових токенів, поряд із помилками додається правильний варіант тексту. Результат: кожен текст стане об'єктом формату:

{Я иду чтать книги.} → {[1, {"иду", "граматична помилка", "йду"}],
[2, {чтать, орфографічна помилка, читати}]}.

У підсумку корпус текстів розділяється на три частини: навчальна вибірка для побудови моделі (~70 % даних), валідаційна вибірка для перевірки якості під час навчання (~15 % даних) та тестова вибірка для оцінки точності на нових, невідомих для моделі текстах (~15 % даних).

Процес навчання моделі машинного навчання для задачі ідентифікації та корекції помилок включає налаштування архітектури моделі, визначення цільової функції, оптимізацію параметрів, а також перевірку та валідацію отриманих результатів. Модель вирішує дві основні задачі, серед яких ідентифікація помилок, що є задачею класифікації, де кожен токен тексту має бути класифікований на один із класів помилок або позначений як "без помилки" та корекція помилок – задача генерації тексту, де модель має запропонувати виправлення для помилкових токенів з урахуванням їх контексту. Така модель має максимізувати ймовірності:

$$P(Y \mid X) = \prod_{i=1}^{N} P(y_i \mid x_1, x_2, \dots, x_N),$$

$$P(\hat{X} \mid X, Y) = \prod_{i=1}^{N} P(\hat{x_i} \mid x_1, x_2, \dots, x_N, y_1, y_2, \dots, y_N),$$

де $P(\hat{x_i} \mid X, Y)$ – імовірність правильного виправлення токена; (x_i), $P(y_i \mid X)$ – імовірність того, що токен (x_i) належить до класу (y_i).

Навчання передбачає використання сучасних методів опрацювання текстів на основі нейронних мовних моделей. Для задачі ідентифікації та корекції помилок найбільш ефективними є дві архітектури, такі як рекурентні нейронні мережі (RNN, LSTM, GRU) та Transformer. Рекурентні нейронні мережі використовуються для задачі аналізу послідовностей. Основна їхня перевага полягає у збереженні контексту попередніх токенів. Наприклад, LSTM (Long Short-Term Memory) дозволяє враховувати довготривалі залежності між токенами:

$$h_t = LSTM(x_t, h_{t-1}),$$

де (h_t) – прихований стан на кроці (t), що містить інформацію про поточний токен і попередні.

Transformer-моделі, такі як BERT, XLM-RoBERTa або mT5, є найефективнішими для ідентифікації та корекції помилок у текстах. Завдяки механізму "уваги" враховуються залежності між усіма токенами у тексті, незалежно від їхньої позиційної віддаленості. Для ідентифікації помилок використовується BERT-модель, а для виправлення помилок можна застосовувати mT5 або GPT.

Цільова функція визначає, наскільки добре модель розпізнає помилки і пропонує правильні виправлення. Для задачі ідентифікації помилок функція втрат базується на крос-ентропії:

$$Loss_{id} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K} y_{i,k}\log(\widehat{y_{i,k}}),$$

де $(y_{i,k})$ – мітка токена x_i, яка належить класу c_k; ($\widehat{y_{i,k}}$)– ймовірність, передбачена моделлю для цього класу. Для задачі корекції помилок функція втрат оцінює різницю між виправленим текстом і правильним текстом:

$$Loss_{cor} = -\frac{1}{N}\sum_{i=1}^{\mathrm{T}}\log P\ (\widehat{x_i}, X, Y),$$

де $P\ (\widehat{x_i}, X, Y)$ обчислюється за допомогою трансформерної моделі. Загальна цільова функція моделі виглядає так:

$$Loss\ = \lambda_1 \cdot Loss_{id} + \lambda_2 \cdot Loss_{cor},$$

де (λ_1) і (λ_2) – вагові коефіцієнти для збалансування задач.

Алгоритм навчання можна описати таким чином:

1. Ініціалізація моделі. На цьому етапі вибирається архітектура моделі та налаштовуються гіперпараметри (кількість шарів, кількість нейронів, розмір ембеддингів тощо).
2. Підготовка даних**.** Тексти перетворюються у векторні подання, а навчальна вибірка розділяється на три підмножини: тренувальну, валідаційну та тестову (70 % / 15 % / 15 %).
3. Процес навчання. Тут відбувається подача токенів з вибірки на вхід моделі, розрахунок передбачених класів помилок ($\hat{Y}$) та виправлених текстів ($\hat{X}$). Далі проходить процес обчислення значення втрат ($Loss$). На фініші цього етапу відбувається оптимізація параметрів моделі за допомогою алгоритмів градієнтного спуску (Adam або AdamW).
4. Валідація. Після кожної епохи процес навчання перевіряється на валідаційних даних. Модель оцінюється за такими метриками: Accuracy, Precision, Recall, F1-score, а також BLEU для оцінки якості виправлених текстів.
5. Тестування. Оцінка моделі на залишених тестових даних для отримання фінальних метрик.

Під час навчання налаштовуються такі гіперпараметри як розмір вікна контексту, що визначає, скільки сусідніх токенів враховувати при аналізі; розмір ембеддингів (256 або 512 для відображення токенів векторами); кількість епох, що зазвичай вибирається експериментально, залежно від швидкості досягнення збіжності (10–20 епох). розмір батчу, що представляє собою кількість текстів, що опрацьовуються за один раз (16, 32, 64); темп навчання (0.001), який знижуватиметься зі зростанням числа епох. Після навчання модель стає здатною точно класифікувати токени тексту за типом помилки, пропонувати якісні виправлення, узгоджені з контекстом, а також генерувати виправлений текст із мінімальними стилістичними та граматичними порушеннями.

Експерименти

У рамках даного дослідження для задачі автоматичного виправлення помилок в україномовних текстах було обрано модель *facebook/mbart-large-50*. Вибір цієї трансформерної архітектури зумовлений її мультимовною природою, що дозволяє ефективно працювати з меншоресурсними мовами, зокрема українською. Завдяки попередньому навчанню на великій кількості паралельних текстів, MBART має високий потенціал до генерації граматично й стилістично коректних речень у задачах Seq2Seq. Крім того, модель добре себе показує в інших завданнях трансформації тексту, зокрема перефразуванні та машинному перекладі, що близьке за природою до виправлення помилок.

Для навчання використовувався корпус UA-GEC, який містить приклади речень з типових помилок української мови та їх виправлень. Попереднє опрацювання даних здійснено за допомогою бібліотеки Stanza, зокрема для розділення на source (вхідні, помилкові речення) та target (виправлені речення), що відповідало формату Seq2Seq навчання.

Навчання виконувалося в середовищі Kaggle з доступним GPU та обмеженим обсягом оперативної пам'яті. З врахуванням обмеженості ресурсів, було підібрано компромісний набір гіперпараметрів: batch_size = 4, max_token_length = 64, learning_rate = 5e-5, n_epochs = 6. Навчання проводилось у два етапи. Спочатку відбуваються 3 епохи з базовими налаштуваннями, а згодом додаткове донавчання ще на 3 епохи з меншою швидкістю навчання (таблиця). Спочатку навчання швидкість становила 5e-5, що сприяла швидкому засвоєнню основних закономірностей, тоді як зниження швидкості до 3e-5 на пізніх етапах допомогло уникнути перенавчання та дало змогу точніше налаштувати ваги.

Значення і призначення гіперпараметрів для навчання моделі

Гіперпараметр	Ітерація 1	Ітерація 2	Призначення
Модель	facebook/mbart-large-50	facebook/mbart-large-50	Попередньо натренована мультимовна Seq2Seq модель
Кількість епох (n_epochs)	3	3	Етапи початкового та донавчання
Розмір батчу (batch_size)	4	4	Компроміс між стабільністю градієнтів і пам'яттю GPU
Максимальна довжина токенів (max_token_length)	64	64	Обмеження довжини речення для зменшення навантаження
Швидкість навчання (learning_rate)	5e-5	3e-5	Вища на старті для швидкої адаптації, нижча на пізніх етапах для стабільності

У рамках кількісного аналізу якості моделі для виправлення помилок в україномовних текстах використано низку метрик, які дозволяють об'єктивно оцінити динаміку навчання. Зокрема, були зафіксовані значення текст-рівневих метрик sacreBLEU, BLEU та METEOR, а також втрати (loss) на тренувальній та валідаційній вибірках упродовж трьох епох навчання (рис. 3-5).

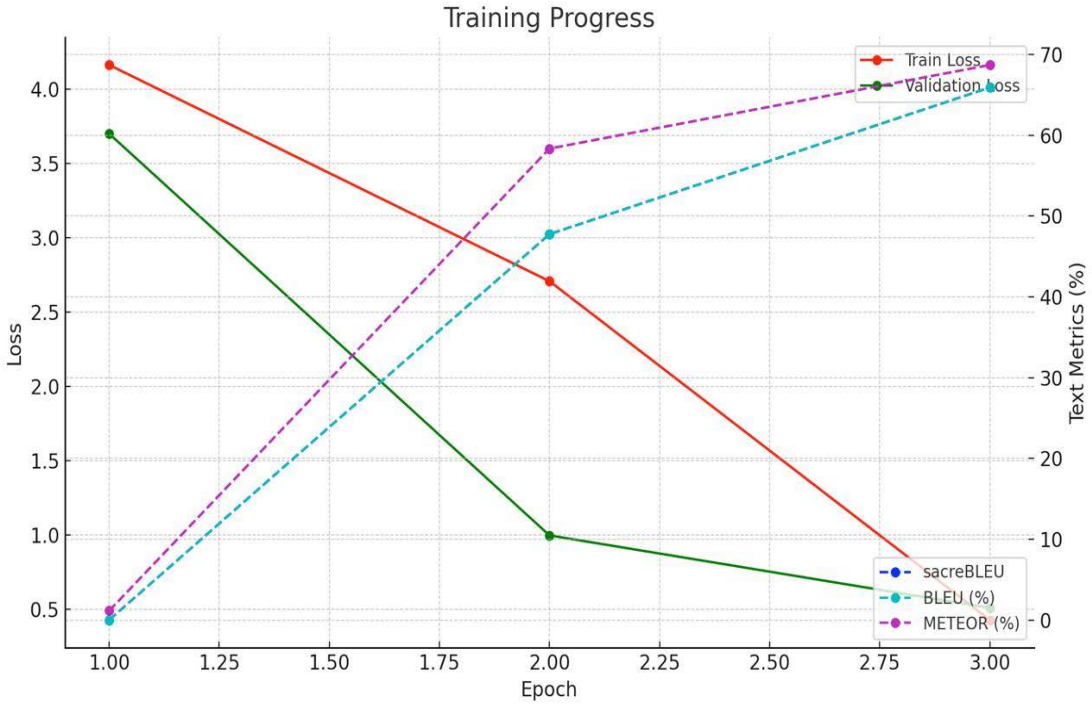

Рис. 3. Графік втрат з врахуванням метрик BLUE та METEOR.

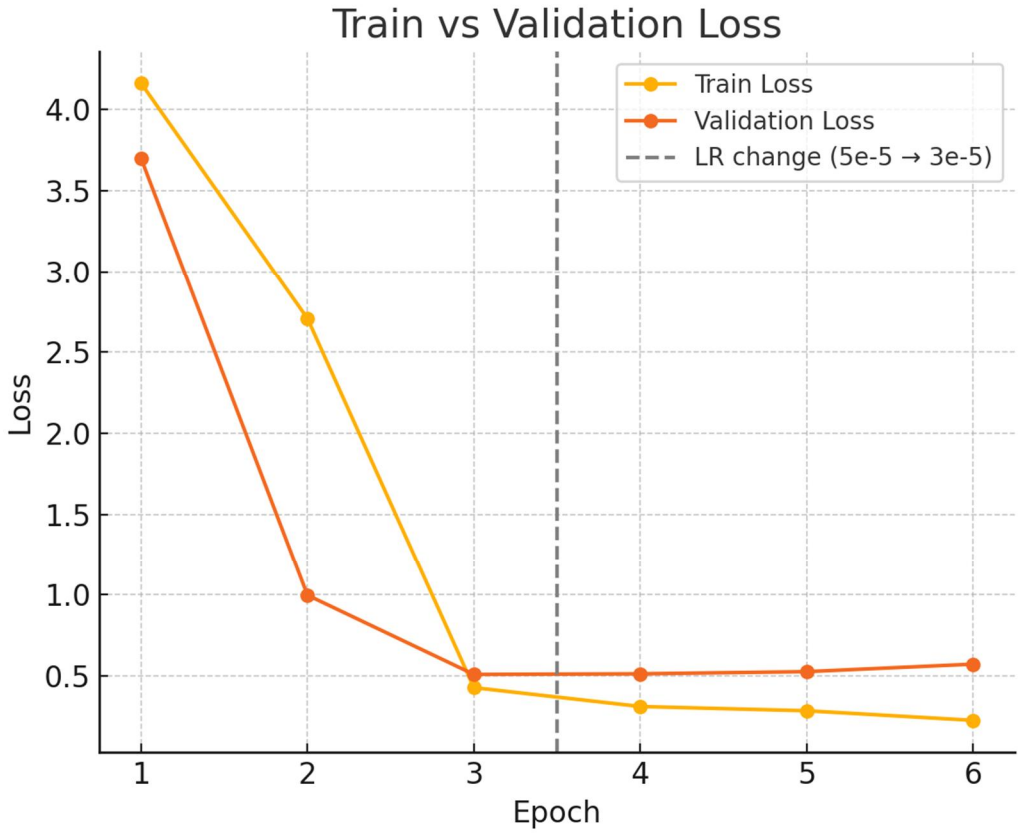

Рис. 4. Графік втрат і графік метрик BLUE

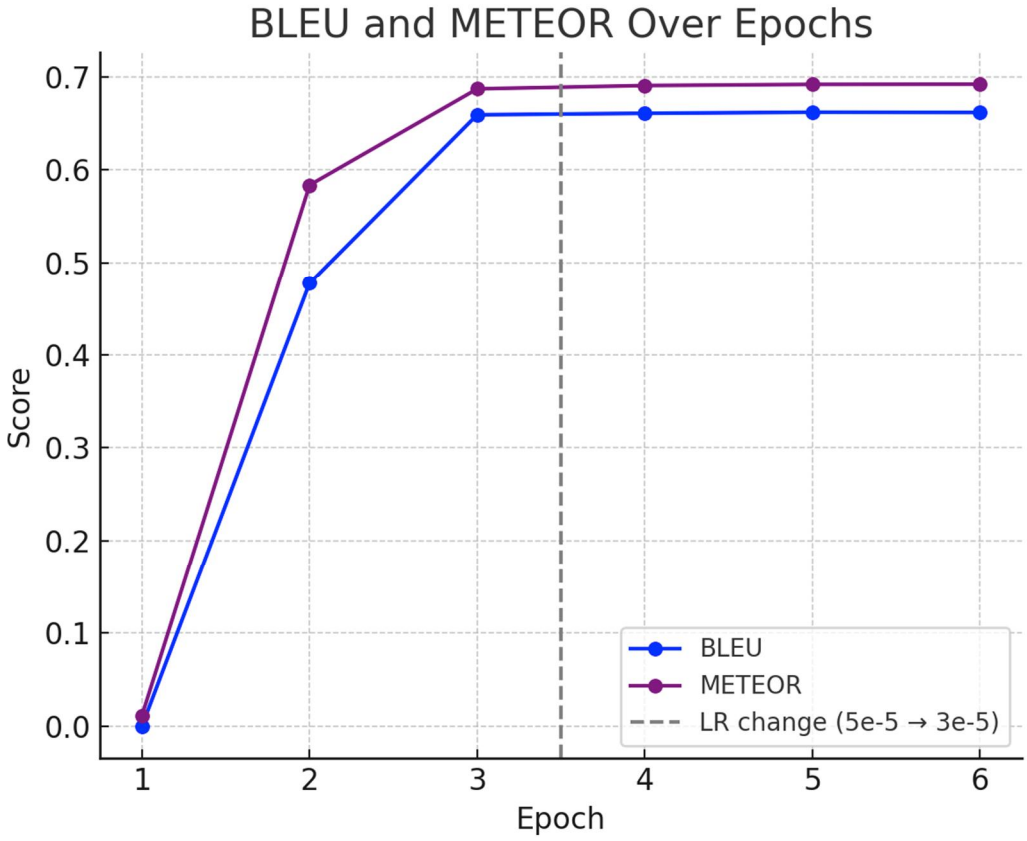

Рис. 5. METEOR після зміни параметру швидкості навчання.

Початкова якість моделі, зафіксована на першій епосі, виявилась очікувано низькою. Значення BLEU-метрики було рівне нулю, sacreBLEU – лише 0.004, а METEOR склала 0.011. Це свідчить про практично повну неспроможність моделі на старті генерувати будь-які виправлення, що хоча б частково збігалися з референтними. Такий результат є типовим для трансформерних моделей до моменту, коли вони починають формувати осмислені відповідності між вхідними і цільовими послідовностями.

Проте вже на другій епосі спостерігається суттєвий прорив. Значення BLEU різко зростає до 0.477, sacreBLEU – до 47.76, а METEOR – до 0.584. Це свідчить про те, що модель почала ефективно засвоювати залежності між помилковими та правильними реченнями, відтворюючи значну частину цільових відповідей або з точним збігом, або з високим ступенем семантичної подібності. Важливо відзначити також падіння значення функції втрат: тренувальний loss зменшився з 4.16 до 2.71, а валідаційний – з 3.70 до 0.99, що є хорошим індикатором того, що навчання йде в правильному напрямі, і модель не лише "запам'ятовує", але й узагальнює закономірності.

На третій епосі зберігається позитивна динаміка. BLEU досягає 0.659, sacreBLEU – 65.92, METEOR – 0.687. При цьому втрати знижуються ще суттєвіше. Тренувальний loss падає до 0.43, валідаційний – до 0.51. Це свідчить про стабільність навчального процесу та відсутність ознак перенавчання, що особливо важливо для задачі корекції тексту, де генеративна здатність моделі має бути гнучкою, а не вузькоспеціалізованою.

Починаючи з четвертої епохи, навчання продовжувалося з меншою швидкістю навчання – learning_rate було зменшено з 5e-5 до 3e-5. Це дало змогу моделі точніше "допиляти" вже сформовані відповідності, проте приріст якості був незначним. BLEU зріс до 0.662, METEOR – до 0.692, а sacreBLEU – до 66.16. Така стабільність метрик свідчить про вихід моделі на плато – вона вже здобула основні закономірності й тепер лише незначно уточнює прогнози. Водночас валідаційна втрата почала повільно зростати (з 0.51 до 0.57), що може свідчити про початок перенавчання. Отже, зменшення learning_rate дозволило уникнути різких коливань, але й не дало суттєвого покращення – тобто найкращі результати були на 3-4 епосі.

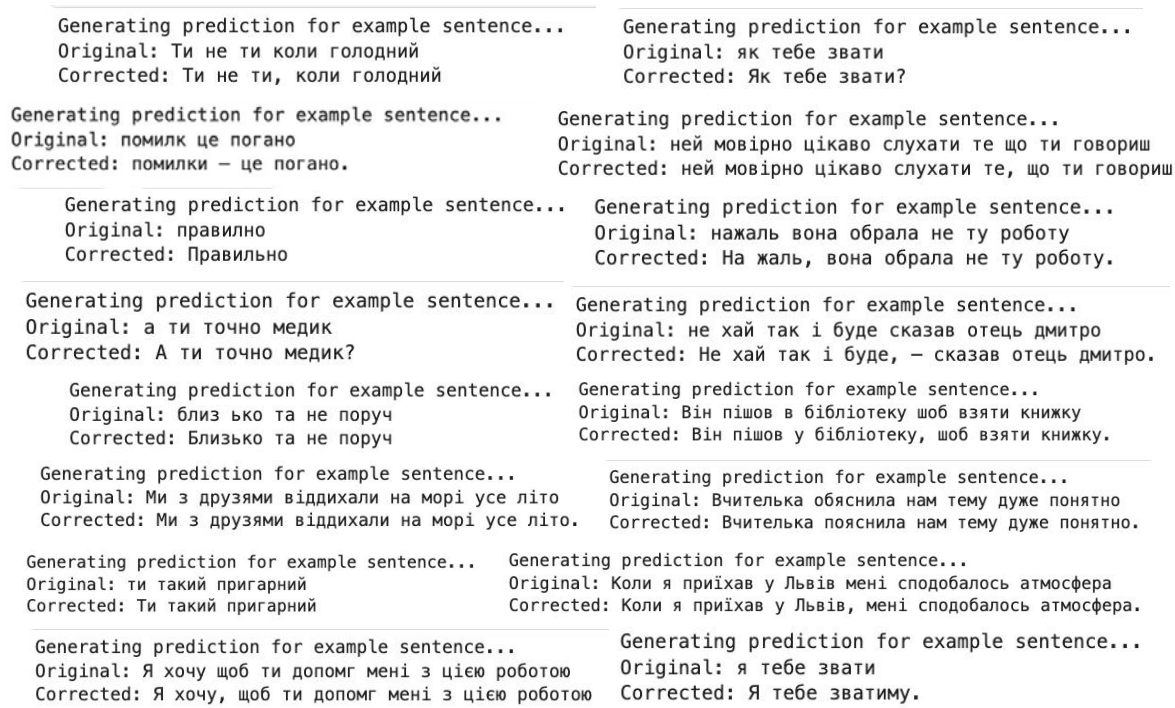

Рис. 6. Результати роботи моделі після навчання.

Узагальнюючи дані графіки, можна зробити висновок, що обраний підхід до навчання моделі є ефективним. Високі показники sacreBLEU та METEOR демонструють не лише лексичну точність, але й стилістичну адекватність згенерованих варіантів. Стрімке зростання метрик між першою і

другою епохами свідчить про те, що навіть за короткий час навчання модель здатна опанувати основи корекції мови. Загалом це дає підстави для впевненого прогнозу щодо потенційного поліпшення якості за умови подальшого тренування, розширення навчального корпусу та збагачення типів мовних похибок у вхідних прикладах.

Оцінка якості роботи моделі здійснювалась шляхом ручного тестування на наборі речень з типово поширеними помилками, що часто трапляються в україномовних текстах. Усього було протестовано десятки речень, що охоплюють різноманітні типи мовних похибок, зокрема пунктуаційні, орфографічні, лексичні та морфологічні. Аналіз результатів дозволяє зробити висновки про поточний рівень сформованих мовних узагальнень у моделі, виявити її сильні сторони, а також позначити сфери, де спостерігається нестача лінгвістичної компетентності.

Найкраще модель показала себе у виправленні пунктуаційних помилок. У більшості випадків вона впевнено розпізнавала та виправляла відсутність ком у складнопідрядних конструкціях. Так, речення «Я бачив як вона грає на піаніно» було успішно перетворене на «Я бачив, як вона грає на піаніно», а в реченні «Коли я приїхав у Львів мені сподобалось атмосфера» була додана кома після підрядної частини, що відповідає нормам українського синтаксису. Аналогічно, в реченні з прямою мовою модель застосувала правильне пунктуаційне оформлення – «не хай так і буде сказав Дмитро» було виправлено до «Не хай так і буде, – сказав Дмитро», що демонструє розуміння моделлю специфіки прямої мови та її інтонаційного виділення на письмі, хоча і лапки модель таки не доставила.

Однак при цьому система лише частково впоралася з орфографічними помилками. Прості випадки, такі як помилкове написання слова з пробілом («Близько») або вживання службового слова з малої літери на початку речення, були успішно виправлені. Але у більш складних ситуаціях, де виправлення потребує глибшого аналізу словникової форми, система не справилася. Наприклад, слово «ней мовірно» залишилося без змін, хоча мало би бути виправлене на «неймовірно». Це свідчить про те, що поточна версія моделі не завжди володіє достатнім рівнем орфографічної чутливості, особливо у випадках, де помилки не мають яскравих контекстуальних підказок і вимагають перевірки на відповідність словниковій нормі.

Аналогічна ситуація спостерігається і з морфологічними помилками. У реченні «Коли я приїхав у Львів, мені сподобалось атмосфера» модель не виявила конфлікту між дієсловом у середньому роді та іменником жіночого роду, залишивши помилкову форму незміненою. Це свідчить про недостатню глибину узгодження між граматичними категоріями в опрацюванні вхідного тексту.

У поодиноких випадках модель демонструє здатність до синтаксичних трансформацій, змінюючи структуру речення з некоректної на граматично правильну. Наприклад, речення «я тебе звати» було трансформовано в «Я тебе зватиму», що вказує на формування базового розуміння предикативних конструкцій та дієслівного узгодження. Проте більшість прикладів цього типу залишаються поза межами успішного виправлення, і трансформаційна здатність моделі наразі виглядає обмеженою.

Висновки

Результатом роботи є розроблена інформаційна технологія на основі математичної моделі корекції помилок в україномовних текстах, орієнтованої на впровадження підходів машинного навчання. У підсумку проведених досліджень сформульовано математичне підґрунтя для вирішення поставлених задач, включаючи формалізацію потоку даних, розміщення компонентів системи та представлення текстів у вигляді, придатному для машинного опрацювання.

Було висвітлено основну математичну структуру системи, яка складається із двох ключових модулів для ідентифікації та корекції помилок. Обидва модулі взаємодіють у межах системи, забезпечуючи коректне опрацювання послідовності тексту. Окремо було побудовано математичну модель задачі ідентифікації помилок, що спирається на ймовірнісний підхід. Основний акцент у моделюванні зроблено на збереженні контексту та врахуванні залежностей між токенами для підвищення точності ідентифікації. Побудовано математичну модель, яка передбачає обчислення

умовної ймовірності вибору найкращого кандидата для виправлення помилкового токена в межах даного контексту тексту. Було розглянуто підхід до підготовки даних через векторизацію текстів, формування навчальної вибірки та організацію процесу навчання моделей на основі великих корпусів текстів. Наголошено на важливості побудови репрезентативного корпусу навчальних даних, що включає тексти з реальними помилками, а також штучно змодельовані приклади помилок із врахуванням їхнього типологічного розподілу.

Було навчено модель на корпусі UA_GEC, що продемонструвала обнадійливі результати у виправленні пунктуації та базових орфографічних помилок, особливо в межах простих і чітко структурованих речень. Водночас вона недостатньо ефективна у виявленні лексичних та морфологічних помилок, а також поки що слабо справляється з глибшими синтаксичними перебудовами. Результати окреслюють чіткі напрями для подальшого вдосконалення системи, а саме розширення навчального корпусу прикладами з морфологічними та лексичними похибками, а також впровадження додаткових механізмів опрацювання, які б компенсували обмежену орфографічну компетентність моделі. У перспективі, покращення цих аспектів дозволить створити більш надійний та комплексний інструмент для автоматичної перевірки україномовних текстів.

У підсумку розроблена математична модель є універсальним підходом до опрацювання українських текстів, що дозволяє вирішувати задачі ідентифікації та корекції помилок у межах єдиної системи. Визначені формальні аспекти взаємодії між компонентами моделі створюють підґрунтя для її ефективного навчання й подальшого впровадження у практичні завдання.

Подяка

Дана стаття підготована завдяки грантової підтримки Національного Фонду Досліджень України, реєстраційний номер проєкту 33/0012 від 3/03/2025 (2023.04/0012) «Розроблення інформаційної системи автоматичного виявлення джерел дезінформації та неавтентичної поведінки користувачів чатів» за конкурсом «Наука для зміцнення обороноздатності України».

СПИСОК ЛІТЕРАТУРИ

Bryant, C., Yuan, Z., Qorib, M. R., Cao, H., Ng, H. T., & Briscoe, T. (2023). Grammatical error correction: A survey of the state of the art. Computational Linguistics, 49(3), 643–701. doi:https://doi.org/10.48550/arXiv.2211.05166.

Brovinska, M. (2024). I waited eight years for Grammarly to support Ukrainian. Dev.ua. Retrieved from https://dev.ua/news/ai-servisy-1706885687

Fedchuk, R., & Vysotska, V. (2024). Current trends in the use of machine learning for error correction in Ukrainian texts. Qeios, Article ID N4VGBJ, 1–18. doi:https://doi.org/10.32388/n4vgbj

Grammarly Inc. (n.d.). About us. Retrieved from https://www.grammarly.com/about

Grammarly. (n.d.). UA-GEC. Retrieved from https://github.com/grammarly/ua-gec

Huang, M., & Fan, R. (2025). Influence of translation errors on information perception in East Slavic languages (Ukrainian–Russian; Russian–Ukrainian). Zeitschrift für Slawistik, 70(1), 141–160. doi:https://doi.org/10.1515/slaw-2025-0006

Kholodna, N., & Vysotska, V. (2023). Technology for grammatical errors correction in Ukrainian text content based on machine learning methods. Radio Electronics, Computer Science, Control, 1, 114. doi:https://doi.org/10.15588/1607-3274-2023-1-12

LanguageTool. (n.d.). We believe that anyone can write beautifully and professionally. Retrieved from https://languagetool.org/about

LanguageTool Community. (n.d.). Error rules for LanguageTool. Retrieved from https://community.languagetool.org/rule/list?lang=uk

Lytvyn, V., Pukach, P., Vysotska, V., Vovk, M., & Kholodna, N. (2023). Identification and correction of grammatical errors in Ukrainian texts based on machine learning technology. Mathematics, 11(4), 904. doi:https://doi.org/10.3390/math11040904

NLP UK. (n.d.). LanguageTool API. GitHub. Retrieved from https://github.com/brown-uk/nlp_uk

Starko, V., Rysin, A., & Shvedova, M. (2021). Ukrainian text preprocessing in GRAC. In 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT) (Vol. 2). IEEE. doi:https://doi.org/10.1109/CSIT52700.2021.9648705

Syvokon, O., & Nahorna, O. (2021). UA-GEC: Grammatical error correction and fluency corpus for the Ukrainian language. arXiv. doi: https://doi.org/10.48550/arXiv.2103.16997

INFORMATION TECHNOLOGIES FOR ERRORS CORRECTION IN UKRAINIAN-LANGUAGE TEXTS BASED ON MACHINE LEARNING

Rostyslav Fedchuk[1], **Victoria Vysotska**[2]

[1, 2] Lviv Polytechnic National University,
Information Systems and Networks Department, Lviv, Ukraine
[1]E-mail: rostyslav.b.fedchuk@lpnu.ua, ORCID: 0009-0002-6669-0369
[2]E-mail: victoria.a.vysotska@lpnu.ua, ORCID: 0000-0001-6417-3689

The relevance of the research is due to the growing need to automate the processes of text analysis and correction, in particular for Ukrainian-language content, which is characterized by a wealth of morphological and syntactic structure. Due to the wide range of errors that can occur in texts, from spelling to contextual, there is an urgent need to create systems that can accurately identify errors and offer their correct corrections. The specificity of the Ukrainian language, including its grammatical complexity and multifacetedness, requires the adaptation of machine learning models to local features. The purpose of the research is to develop a mathematical model of a decision support system for identifying and correcting errors in Ukrainian-language texts. The task includes both the formalization and mathematical description of the text processing process, and the construction of a model with an orientation to the tasks of classification and text generation. Special attention is paid to the effective consideration of structural features specific to the Ukrainian language in order to increase the accuracy and productivity of the system. The research method is based on the construction of a mathematical model of error correction, which is presented as a context-aware text generation problem. The study used statistical methods and machine learning approaches. Special attention is paid to the formation of a training sample, which combines texts with real and artificial errors to ensure a balanced learning process. The correction module includes generation mechanisms based on contextual models capable of predicting the correct correction for erroneous tokens. Approaches to text vectorization are mathematically substantiated, taking into account the peculiarities of the morphology and syntax of the Ukrainian language. The constructed model is a universal basis for creating intelligent systems for automatic editing of Ukrainian-language text. As a result of the research, approaches to building an error correction model in Ukrainian-language texts are formulated and mathematically substantiated. The main result was the creation of an integrated system that uses contextual information to ensure high accuracy of error recognition and correction. The applied mathematical methods include probabilistic approaches and vector representation of tokens, which allows adapting the system to the peculiarities of the Ukrainian language with its high morphological and syntactic complexity. The formed basis of the model creates opportunities for scaling and further use in practical tasks, such as automatic text editing or improving the quality of content in the Ukrainian-speaking environment.

Keywords – error correction, NLP, text generation, machine learning, Ukrainian language, text processing.