Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів керованості машинного перекладу

(тема)

Виконав:
здобувач другого року навчання,
групи СШІм-22-2
Максим Крутіхін
(власне ім'я, прізвище)
Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва освітньої програми)
Керівник доц. Олексій Турута
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Олег ЗОЛОТУХІН
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Крутіхіну Максиму Леонідовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів керованості машинного перекладу

затверджена наказом університету від 21 квітня 20 25 р. № 295Ст

2. Термін подання студентом роботи до екзаменаційної комісії 10 червня 20 25 р.

3. Вихідні дані до роботи Документація для роботи з нейронними мережами, аналіз тексту, аналіз роботи машинного перекладу, навчання, аналіз датасетів, документація для роботи з різними перекладачами

4. Перелік питань, що потрібно опрацювати в роботі
1) Огляд технологій машинного перекладу та керованості
2) Постановка задачі дослідження
3) Аналіз датасетів та даних
4) Розробка системи керованого машинного перекладу
5) Експериментальне дослідження та аналіз результатів

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	21.04.2025	виконано
2	Постановка задачі та узгодження з керівником	03.05.2025 – 05.05.2025	виконано
3	Передпроєктне обстеження	07.05.2025 – 12.05.2025	виконано
4	Опис та аналіз предметної галузі	07.05.2025 – 12.05.2025	виконано
5	Постановка завдання	09.05.2025	виконано
6	Аналіз існуючих аналогів	09.05.2025	виконано
7	Вибір інструментів для програмної реалізації	10.05.2025 – 15.05.2025	виконано
8	Реалізація машинного перекладу	13.05.2025 – 23.05.2025	виконано
9	Оформлення пояснювальної записки	30.05.2025 – 07.06.2025	виконано
10	Захист перед ЕК	10.06.2025	виконано

Дата видачі завдання 21 квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Олексій Турута
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 91 с., 4 рис., 7 табл., 1 дод., 46 джерел.

КЕРОВАНІСТЬ, МАШИННИЙ ПЕРЕКЛАД, BERTSCORE, BLEU, FASTTEXT, OPENNMT, PYTHON, SEQUENCE CONTROLS, TRANSFORMER.

Об’єктом дослідження в даній роботі є процес машинного перекладу з можливістю керуванням стилем, термінологією, і іншими аргументами перекладу.

Предметом дослідження є функціональна система керованого машинного перекладу для англійської-української мови.

Метою роботи є розробка інноваційний методів керованості в машинному перекладі, які дозволяють адаптувати результати перекладу до специфічних вимог користувача, та реалізація яка демонструє підходи на практиці.

Методи дослідження включають аналіз архітектур машинного перекладу, формалізацію задачі, використання sequence controls, навчання моделі з векторним представленням, та застосування метрик для оцінювання результатів.

У ході роботи було створено систему керованого перекладу, з можливістю задати стиль, формальність, та термінологію перекладу. Експерименти засвідчили підвищення точності й адаптивності перекладу у порівнянні з некерованими підходами.

ABSTRACT

Master’s thesis contains: 91 pp., 4 fig., 7 tabl., 1 ann., 46 references.

BERTSCORE, BLEU, CONTROLLABILITY, FASTTEXT, MACHINE TRANSLATION, OPENNMT, PYTHON, SEQUENCE CONTROLS, TRANSFORMER.

The object of research in this thesis is the process of machine translation with controllability over style, terminology, and other translation arguments.

The subject of research is a functional controllable machine translation system for the Eng-Ukr language.

The aim of the work is to develop innovative methods for controllable machine translation that allow adapting the output to user-specific requirements, and to implement an architecture demonstrating these approaches in practice.

The research methods include analysis of machine translation architectures, formalization of the task with control variables, use of sequence controls, training the model with FastText word embeddings, and evaluation with BLEU, METEOR, and BERTScore metrics.

As a result, a controllable machine translation system was developed, enabling customization of translation style, formality, and terminology.

Experiments showed improved translation accuracy and adaptability compared to non-controllable approaches, The proposed solution has practical value for academic, technical, and professional translation tasks.

ЗМІСТ

ВСТУП

Актуальність теми: сучасний світ характеризується інтенсивною глобалізацією та зростаючою потребою у якісному міжмовному спілкуванні, що робить машинний переклад одним із найбільш затребуваних напрямків розвитку технологій штучного інтелекту. Особливої актуальності набуває проблема керованості машинного перекладу, коли користувачі потребують не просто автоматичного перетворення тексту з однієї мови на іншу, але й можливості впливати на стиль, тональність, термінологію та інші характеристики перекладу відповідно до специфічних потреб та контексту використання. Для англо-української мовної пари ця проблема є особливо гострою через відносну обмеженість якісних перекладацьких ресурсів, складність морфологічної структури української мови та зростаючі потреби україномовної спільноти у доступі до міжнародного контенту та ефективних засобах міжкультурної комунікації. Розвиток методів керованості машинного перекладу відкриває нові можливості для створення адаптивних систем, здатних задовольняти різноманітні користувацькі потреби від технічного перекладу до творчої адаптації текстів, що робить дане дослідження надзвичайно актуальним як з наукової, так і з практичної точки зору.

Мета і завдання дослідження: метою дослідження є розробка та дослідження ефективних методів керованості машинного перекладу для англо-української мовної пари з практичною реалізацією функціональної системи, що демонструє переваги запропонованих підходів. Для досягнення поставленої мети необхідно вирішити наступні завдання: проаналізувати сучасні підходи до керованого машинного перекладу та виявити їх сильні та слабкі сторони; розробити методи інтеграції sequence controls з векторними представленнями слів у Transformer архітектурі; створити комплексну систему підготовки та обробки паралельних корпусів для навчання керованих моделей перекладу; імплементувати функціональну систему

машинного перекладу з веб-інтерфейсом та механізмами керованості; провести експериментальне дослідження ефективності розроблених методів на різних типах текстів; порівняти результати з існуючими комерційними та академічними рішеннями; сформулювати практичні рекомендації щодо застосування розроблених методів у реальних умовах.

Об'єкт дослідження: процес машинного перекладу між англійською та українською мовами з можливістю керування характеристиками генерованого перекладу через різні типи контрольних сигналів та параметрів.

Предмет дослідження: методи керованості машинного перекладу, включаючи sequence controls, інтеграцію попередньо навчених векторних представлень, архітектурні модифікації нейронних мереж та їх вплив на якість і контрольованість процесу автоматичного перекладу.

Практичне значення роботи: полягає у створенні функціональної системи керованого машинного перекладу, що може бути використана для розв'язання реальних задач автоматичного перекладу в академічних, корпоративних та державних установах. Розроблені методи керованості дозволяють адаптувати систему до специфічних потреб різних доменів, включаючи технічну документацію, академічні публікації, юридичні тексти та освітні матеріали. Створена система може служити основою для розробки спеціалізованих перекладацьких рішень для україномовної спільноти, сприяючи підвищенню доступності міжнародного контенту та покращенню якості міжкультурної комунікації. Модульна архітектура системи забезпечує можливість легкого розширення функціональності та адаптації до нових вимог, що робить її цінним інструментом як для практичного використання, так і для подальших наукових досліджень у галузі обробки природної мови.

Методи дослідження : у роботі використано комплекс теоретичних та експериментальних методів дослідження, що включає: аналіз літературних джерел для систематизації існуючих підходів до керованого машинного

перекладу; методи обчислювальної лінгвістики для формалізації задач обробки природної мови; алгоритми глибокого навчання, зокрема Transformer архітектури та FastText для створення векторних представлень слів; статистичні методи для оцінювання якості машинного перекладу, включаючи BLEU, METEOR та інші автоматичні метрики; експериментальні методи для валідації ефективності розроблених підходів на стандартизованих датасетах; порівняльний аналіз для позиціонування розробленої системи відносно існуючих рішень; методи програмної інженерії для створення модульної архітектури системи та веб-інтерфейсу; методи обробки та аналізу великих обсягів текстових даних для підготовки тренувальних корпусів.

1 ОГЛЯД ТЕХНОЛОГІЙ МАШИННОГО ПЕРЕКЛАДУ ТА КЕРОВАНОСТІ

1.1 Опис принципів машинного перекладу

Машинний переклад представляє собою автоматизований процес перетворення тексту з однієї природної мови на іншу за допомогою комп'ютерних алгоритмів та програмних систем. Ця галузь обчислювальної лінгвістики об'єднує методи штучного інтелекту, статистичного аналізу та глибокого навчання для розв'язання однієї з найскладніших задач обробки природної мови. Основною метою машинного перекладу є створення систем, здатних відтворювати не лише лексичне значення слів та фраз, але й граматичну структуру, семантичний зміст та культурні особливості тексту оригіналу. Протягом десятиліть розвитку цієї сфери було запропоновано численні підходи, починаючи від простих словникових замін та закінчуючи складними нейронними архітектурами, що дозволяють досягати якості перекладу, порівнянної з людською [1].

Перші спроби автоматизації перекладу базувались на статистичних методах машинного перекладу (Statistical Machine Translation, SMT), які фундаментально змінили підхід до розуміння міжмовних відповідностей. Статистичний машинний переклад ґрунтується на ймовірнісних моделях, що навчаються на великих паралельних корпусах текстів для виявлення закономірностей перекладу між мовними парами. Центральною концепцією SMT є модель перекладу, яка оцінює ймовірність того, що певна послідовність слів у цільовій мові є правильним перекладом заданої послідовності в мові-джерелі. Цей підхід використовує статистичні методи для вирівнювання слів, фраз та речень між паралельними текстами, створюючи таблиці перекладних еквівалентів та моделі мовного порядку, що дозволяють генерувати граматично правильні переклади. Незважаючи на певні обмеження у роботі з довгими залежностями та контекстом,

статистичні методи заклали міцний фундамент для подальшого розвитку більш досконалих технологій [2].

Революційним етапом у розвитку машинного перекладу стало впровадження нейронного машинного перекладу (Neural Machine Translation, NMT), який кардинально змінив парадигму обробки мовної інформації. Нейронний машинний переклад базується на глибоких нейронних мережах, зокрема архітектурах encoder-decoder, які здатні вивчати складні нелінійні залежності між елементами вхідного та вихідного тексту. На відміну від статистичних методів, що оперують дискретними компонентами мови, нейронні моделі представляють слова та фрази у вигляді багатовимірних векторів у неперервному семантичному просторі, що дозволяє краще відображати смислові відношення та контекстуальні зв'язки. Encoder у такій архітектурі перетворює вхідну послідовність слів на компактне векторне представлення, яке містить усю суттєву інформацію про вміст та структуру оригінального тексту, тоді як decoder генерує переклад, використовуючи це представлення та механізми уваги для фокусування на релевантних частинах вхідного тексту [3].

Подальшим проривом у галузі машинного перекладу стала розробка архітектури Transformer, яка повністю переосмислила підходи до моделювання послідовностей та обробки природної мови. Transformer представляє інноваційну архітектуру, що базується виключно на механізмах самоуваги (self-attention) та не використовує рекурентні або згорткові компоненти, що дозволяє ефективно обробляти паралельно всі елементи послідовності одночасно. Основою цієї архітектури є механізм багатоголової уваги (multi-head attention), який дозволяє моделі одночасно фокусуватися на різних аспектах вхідної інформації та виявляти складні залежності між словами незалежно від їх позиційної відстані у тексті. Transformer складається з стеку encoder та decoder блоків, кожен з яких містить шари самоуваги та нейронні мережі прямого поширення з залишковими з'єднаннями та нормалізацією, що забезпечує стабільне

навчання глибоких моделей. Ця архітектура не лише значно покращила якість перекладу, але й стала основою для розробки великих мовних моделей наступного покоління [4].

Сучасний етап розвитку машинного перекладу характеризується появою великих мовних моделей (Large Language Models, LLM), які демонструють безпрецедентні можливості у розумінні та генерації природної мови. Ці моделі, такі як GPT серії, T5, PaLM та інші, навчаються на величезних обсягах текстових даних з використанням методів самонавчання (self-supervised learning) та здатні виконувати широкий спектр мовних задач, включаючи переклад, без спеціалізованого навчання для кожної конкретної задачі. LLM використовують архітектуру Transformer у якості основи, але масштабуються до мільярдів параметрів, що дозволяє їм вивчати складні мовні патерни та міжмовні відповідності з неструктурованих текстових корпусів. Особливістю сучасних LLM є їх здатність до навчання в контексті (in-context learning), коли модель може адаптуватися до нових задач перекладу просто через надання прикладів у промпті, без додаткового фінтюнінгу. Це відкриває нові можливості для керованого машинного перекладу, де користувачі можуть впливати на стиль, регістр та специфічні особливості перекладу через спеціально сформульовані інструкції та контекстуальні підказки [5].

1.2 Поняття керованості у машинному перекладі

Керованість у машинному перекладі визначається як здатність системи адаптувати процес генерації перекладу відповідно до заданих користувачем параметрів, контекстуальних вимог або специфічних характеристик цільового тексту. Цей концептуальний підхід передбачає можливість впливати на різні аспекти перекладацького процесу, включаючи стилістичні особливості, термінологічні вподобання, граматичні конструкції та семантичні нюанси вихідного тексту. Керованість

фундаментально відрізняється від традиційного автоматичного перекладу тим, що вона надає користувачеві або системі інструменти для активного втручання у процес генерації, замість пасивного отримання єдино можливого варіанту перекладу. Ця парадигма особливо затребувана у професійних середовищах, де перекладацькі рішення повинні відповідати специфічним галузевим стандартам, корпоративним стилістичним керівництвам або культурним особливостям цільової аудиторії. Розвиток керованих систем машинного перекладу відкриває нові можливості для персоналізації мовних технологій та створення більш гнучких інструментів для міжкультурної комунікації [6, с. 16].

У контексті нейронних мережевих архітектур керованість реалізується через різноманітні механізми додаткового вводу та структурні модифікації моделей, що дозволяють інтегрувати зовнішню інформацію безпосередньо у процес навчання та інференції. Одним з найпоширеніших підходів є використання додаткових токенів керування (control tokens), які вставляються на початок або в середину вхідної послідовності та несуть інформацію про бажані характеристики перекладу, такі як формальність регістру, галузева належність тексту або цільова демографічна група. Архітектурні модифікації можуть включати додаткові шари ембедингів для кодування керуючих сигналів, спеціалізовані attention механізми, що дозволяють моделі селективно фокусуватися на релевантних аспектах керуючої інформації, та гібридні decoder архітектури, які одночасно генерують переклад та враховують обмеження, задані керуючими параметрами. Fine-tuning представляє ще один потужний метод досягнення керованості, коли попередньо навчена модель додатково тренується на спеціалізованих датасетах, що містять приклади перекладів з бажаними характеристиками, дозволяючи системі адаптуватися до специфічних доменів або стилістичних вимог [7, с. 11].

Sequence controls являють собою спеціалізований механізм керованості, що інтегрується безпосередньо в архітектуру encoder-decoder

моделей та дозволяє точно контролювати процес генерації послідовностей на рівні структурних елементів. Цей підхід передбачає використання спеціальних маркерів початку та кінця послідовності (start та end tokens), які не лише позначають межі генерованого тексту, але й можуть нести додаткову семантичну інформацію про характер очікуваного перекладу. Sequence controls можуть включати інформацію про довжину цільової послідовності, що дозволяє контролювати стисліть або детальність перекладу, про синтаксичну структуру, яка повинна бути збережена або модифікована, та про специфічні лінгвістичні феномени, такі як використання пасивного стану, модальність або темпоральні характеристики. Інтеграція sequence controls у навчальний процес вимагає ретельної розробки стратегій анотування тренувальних даних та модифікації функцій втрат, що враховують не лише якість перекладу, але й відповідність згенерованого тексту заданим керуючим параметрам [8].

Революційний підхід до керованості машинного перекладу з'явився з розвитком великих мовних моделей, які дозволили реалізувати керування через промпт-інжиніринг та контекстуальне навчання без необхідності модифікації архітектури або додаткового навчання. Промпт-інжиніринг представляє мистецтво формулювання інструкцій та контекстуальних підказок, які направляють поведінку мовної моделі у бажаному напрямку через природномовний інтерфейс. Цей метод дозволяє користувачам специфікувати стиль перекладу, цільову аудиторію, галузеву термінологію та навіть емоційне забарвлення через детальні текстові описи у промпті, що робить керування перекладом інтуїтивно зрозумілим для нетехнічних користувачів. In-context learning, як природне розширення промпт-інжинірингу, дозволяє моделям адаптуватися до специфічних перекладацьких задач через надання прикладів бажаного перекладу безпосередньо у вхідному контексті, що створює ефект "навчання на льоту" без зміни параметрів моделі. Цей підхід особливо ефективний для роботи з рідкісними мовними парами, специфічною термінологією або унікальними

стилістичними вимогами, де традиційні методи fine-tuning можуть бути непрактичними через обмеженість тренувальних даних [9, с. 10].

Гібридні підходи до керованості представляють найсучасніші методи, що поєднують переваги різних технік для досягнення максимальної гнучкості та точності у контролі процесу перекладу. Ці системи інтегрують структурні модифікації нейронних архітектур з можливостями промпт-інжинірингу, створюючи багаторівневі механізми керування, що працюють на різних етапах обробки мовної інформації. Наприклад, комбінація спеціалізованих attention механізмів з контекстуальними інструкціями дозволяє досягти як тонкого архітектурного контролю над процесом генерації, так і гнучкого користувацького інтерфейсу для специфікації вимог до перекладу.

Адаптивні системи керованості використовують методи метанавчання та трансферного навчання для автоматичного визначення оптимальних стратегій керування на основі характеристик вхідного тексту та історії взаємодії з користувачем. Персоналізовані моделі керованості навчаються на індивідуальних перекладацьких преференціях користувачів, створюючи унікальні профілі стилю та термінологічних вподобань, що дозволяє автоматично адаптувати процес перекладу до специфічних потреб кожного користувача без явного задання керуючих параметрів [10, с. 7].

Таким чином, сучасна керованість у машинному перекладі представляє собою багатоаспектну парадигму, що охоплює широкий спектр технологічних рішень від архітектурних модифікацій нейронних мереж до інтелектуальних промпт-стратегій для великих мовних моделей. Еволюція цієї галузі демонструє поступовий перехід від жорстких, заздалегідь визначених параметрів керування до динамічних, адаптивних систем, здатних самостійно оптимізувати стратегії перекладу відповідно до контексту та вимог користувача (рисунок 1.1).

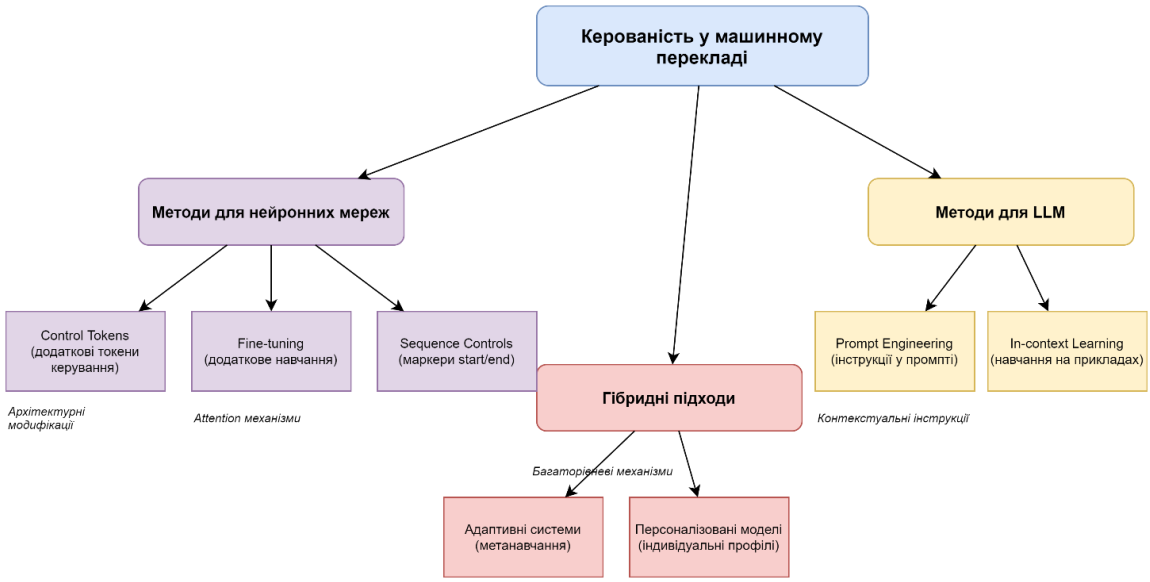

Рисунок 1.1 – Класифікація методів керованості у машинному перекладі

Розвиток гібридних підходів відкриває нові можливості для створення персоналізованих перекладацьких систем, що можуть одночасно забезпечувати високу якість автоматичного перекладу та гнучкість у адаптації до специфічних потреб різних галузей, культурних контекстів та індивідуальних користувацьких преференцій. Цей багатовимірний підхід до керованості формує основу для подальших досліджень у галузі адаптивного машинного перекладу та розробки більш інтелектуальних мовних технологій [11, с. 132].

1.3 Аналіз існуючих продуктів та рішень

Сучасний ландшафт комерційних та академічних рішень машинного перекладу характеризується значною різноманітністю підходів, архітектур та спеціалізацій, що відображає зростаючі потреби різних секторів економіки та наукових спільнот у високоякісних мовних технологіях. Аналіз існуючих продуктів дозволяє виявити основні тенденції розвитку галузі, оцінити ефективність різних технологічних рішень та визначити

прогалини, які потребують подальшого дослідження та розробки. Комерційні системи машинного перекладу, такі як Google Translate, Microsoft Translator, Amazon Translate та DeepL, демонструють різні підходи до балансування між універсальністю та спеціалізацією, швидкістю обробки та якістю результатів, простотою використання та гнучкістю налаштувань. Академічні фреймворки, включаючи OpenNMT, FairSeq, Marian NMT та Tensor2Tensor, надають дослідникам та розробникам інструменти для експериментування з новими архітектурами та методами, сприяючи постійному вдосконаленню технологій машинного перекладу. Інституційні рішення, призначені для специфічних організаційних потреб, демонструють важливість адаптації загальних технологій до конкретних доменів та вимог безпеки [12].

Європейська платформа eTranslation, розроблена Європейською комісією, представляє унікальний приклад інституційного підходу до машинного перекладу, що поєднує передові технологічні рішення з специфічними вимогами міжнародної організації щодо багатомовності та конфіденційності. Ця система забезпечує переклад між усіма 24 офіційними мовами Європейського Союзу, а також деякими додатковими мовами, використовуючи нейронні технології перекладу з додатковими механізмами адаптації до європейської адміністративної термінології та стилістики. eTranslation інтегрує спеціалізовані словники та глосарії, розроблені європейськими інституціями, що дозволяє забезпечити термінологічну консистентність та відповідність офіційним стандартам документообігу. Система також впроваджує механізми керованості, що дозволяють користувачам специфікувати тип документа, цільову аудиторію та рівень формальності, адаптуючи процес перекладу до конкретних комунікативних потреб. Особливістю eTranslation є її інтеграція з екосистемою європейських цифрових сервісів, що забезпечує безшовну роботу з різними адміністративними системами та гарантує дотримання суворих стандартів захисту персональних даних згідно з GDPR [13, с. 134].

Grammarly представляє інноваційний приклад використання технологій машинного перекладу в контексті граматичної корекції та покращення якості тексту, демонструючи потенціал інтеграції різних аспектів обробки природної мови для створення комплексних мовних помічників. Система використовує підходи, засновані на Grammatical Error Correction (GEC), які можна розглядати як спеціалізовану форму машинного перекладу, де вхідний текст з помилками «перекладається» у граматично правильний варіант тієї ж мови. Grammarly інтегрує множинні моделі нейронного навчання для виявлення та корекції різних типів мовних помилок, включаючи граматичні неточності, стилістичні недоліки, пунктуаційні порушення та лексичні неточності.

Система демонструє високий рівень керованості через можливість налаштування на різні стилі письма, цільові аудиторії та типи документів, від академічних робіт до ділової кореспонденції. Алгоритми Grammarly також включають механізми контекстуального аналізу, що дозволяють системі розуміти семантичні інтенції автора та пропонувати не лише граматично правильні, але й стилістично доречні варіанти редагування, ефективно функціонуючи як інтелектуальний редактор з елементами внутрішньомовного перекладу [14].

OpenNMT представляє найвпливовіший серед академічних фреймворків машинного перекладу, що забезпечує дослідників та практиків потужними інструментами для розробки, навчання та розгортання нейронних систем перекладу з високим рівнем кастомізації та експериментальної гнучкості. Цей open-source фреймворк підтримує широкий спектр архітектур, від класичних RNN-based encoder-decoder моделей до сучасних Transformer архітектур, дозволяючи дослідникам експериментувати з різними підходами та порівнювати їх ефективність на конкретних датасетах та задачах. OpenNMT надає комплексні можливості для препроцесингу даних, включаючи токенізацію, субворд сегментацію, створення словників та обробку паралельних корпусів, а також розвинені

механізми постпроцесингу для оптимізації якості генерованих перекладів. Фреймворк інтегрує різноманітні методи керованості, включаючи підтримку додаткових features, control tokens, domain adaptation та multi-domain навчання, що робить його особливо цінним для дослідження саме тих аспектів машинного перекладу, які є предметом даного дослідження (таблиця 1.1). Екосистема OpenNMT також включає спеціалізовані інструменти для аналізу якості перекладу, візуалізації attention patterns та debugging моделей, що значно спрощує процес розробки та оптимізації нових підходів до керованого машинного перекладу [15, с. 90].

Таблиця 1.1 – Порівняльна характеристика основних платформ машинного перекладу

Платформа	Тип	Основна архітектура	Підтримувані мови	Методи керованості	Цільова аудиторія
Google Translate	Комерційна	Transformer + LLM	100+	Контекстні підказки, domain hints	Масовий користувач
DeepL	Комерційна	Transformer	30+	Формальність, глосарії	Професійний переклад
eTranslation	Інституційна	Neural MT + Custom	24 EU + додаткові	Тип документа, термінологія	Європейські інституції
Grammarly	Комерційна	GEC + ML	Англійська	Стиль письма, аудиторія	Редагування тексту
OpenNMT	Open-source	Configurable	Будь-які	Control tokens, domain adaptation	Дослідники, розробники
Microsoft Translator	Комерційна	Transformer	70+	Custom models, глосарії	Корпоративні клієнти

Аналіз представлених рішень демонструє кілька фундаментальних тенденцій у розвитку сучасних систем машинного перекладу та їх адаптації до різноманітних користувацьких потреб та технологічних вимог. По-перше, спостерігається чітка сегментація ринку між універсальними

рішеннями, орієнтованими на широку аудиторію, та спеціалізованими системами, розробленими для конкретних доменів чи організаційних потреб. По-друге, всі сучасні платформи демонструють певний рівень керованості, хоча методи та ступінь цієї керованості значно різняться від простих підказок контексту до складних систем адаптації домену. По-третє, очевидною є тенденція до інтеграції машинного перекладу з іншими технологіями обробки природної мови, що створює комплексні мовні екосистеми замість ізольованих перекладацьких інструментів. Водночас, аналіз виявляє значні прогалини у сфері персоналізованої керованості перекладу, зокрема у можливостях динамічної адаптації до індивідуальних стилістичних преференцій користувачів та автоматичної оптимізації параметрів перекладу на основі зворотного зв'язку, що обґрунтовує актуальність подальших досліджень у цьому напрямку [16, с. 469].

1.4 Невирішені проблеми та мета дослідження

Незважаючи на значні досягнення у галузі машинного перекладу та розвиток різноманітних методів керованості, існує низка фундаментальних проблем, що обмежують ефективність сучасних систем та їх здатність адаптуватися до специфічних потреб користувачів і контекстуальних вимог. Однією з найсуттєвіших проблем є обмежена гранулярність керування процесом перекладу, коли більшість існуючих систем надають лише поверхневі можливості впливу на результат, такі як вибір рівня формальності або загального домену, але не дозволяють тонко налаштовувати специфічні аспекти перекладу, включаючи синтаксичні структури, лексичний вибір або культурні адаптації. Проблема семантичної консистентності у керованому перекладі полягає в тому, що зміна одного аспекту перекладу (наприклад, стилю) часто непередбачувано впливає на інші характеристики тексту, порушуючи цілісність та адекватність перекладу. Динамічна адаптація до контексту залишається значною

технічною проблемою, оскільки більшість систем не здатні автоматично коригувати свою поведінку на основі накопиченого досвіду взаємодії з конкретним користувачем або специфікою перекладної задачі [14].

Технологічні обмеження сучасних архітектур створюють додаткові перешкоди для розвитку ефективних методів керованості, особливо у контексті інтеграції різнотипних керуючих сигналів та забезпечення їх сумісності з існуючими нейронними архітектурами. Проблема мультимодальної керованості виникає через складність одночасного врахування різних типів керуючої інформації, таких як структурні обмеження, термінологічні вимоги, стилістичні преференції та контекстуальні підказки, без створення конфліктів між цими параметрами. Ефективність навчання керованих моделей значно знижується через необхідність балансування між загальною якістю перекладу та точністю виконання керуючих інструкцій, що часто призводить до субоптимальних результатів в одному або обох аспектах. Проблема оцінювання якості керованого перекладу ускладнюється відсутністю стандартизованих метрик, здатних одночасно враховувати як лінгвістичну точність перекладу, так і ступінь відповідності заданим керуючим параметрам. Масштабованість керованих систем також представляє серйозний виклик, оскільки додавання нових типів керування або розширення кількості підтримуваних мовних пар експоненційно збільшує складність архітектури та обчислювальні вимоги.

Методологічні прогалини у дослідженні керованості машинного перекладу виявляються у недостатній систематизації підходів та відсутності комплексного теоретичного фреймворку, який би дозволив порівнювати та об'єднувати різні методи керування. Проблема персоналізації полягає у складності створення індивідуальних профілів користувачів, що автоматично адаптують поведінку системи до специфічних перекладацьких преференцій без явного задання параметрів для кожного випадку використання. Інтерпретованість керованих моделей залишається

серйозною проблемою, оскільки користувачі часто не розуміють, як саме їхні інструкції впливають на процес перекладу, що знижує довіру до системи та ускладнює налагодження неочікуваних результатів.

Проблема трансферу знань між різними мовними парами у контексті керованості означає, що навички керування, набуті моделлю для однієї пари мов, часто не переносяться ефективно на інші мовні комбінації, вимагаючи окремого навчання для кожної мовної пари. Адаптивність до нових типів керування представляє ще один виклик, коли системи не здатні швидко інкорпорувати нові види керуючих сигналів без повного перенавчання або значних архітектурних модифікацій [16].

Розрив між академічними дослідженнями та практичними потребами промисловості створює додаткові перешкоди для впровадження передових методів керованості у реальних системах машинного перекладу. Більшість академічних робіт фокусуються на досягненні високих показників на стандартних бенчмарках, але не враховують практичні обмеження, такі як латентність обробки, споживання ресурсів, простота інтеграції з існуючими системами та можливість масштабування для обслуговування великої кількості користувачів одночасно.

Проблема стандартизації інтерфейсів керування означає, що різні системи використовують несумісні підходи до специфікації керуючих параметрів, що ускладнює міграцію між платформами та створення універсальних інструментів розробки. Економічна ефективність розробки та підтримки керованих систем також викликає занепокоєння, оскільки складність таких систем значно збільшує витрати на розробку, тестування та експлуатацію порівняно з традиційними неконтрольованими моделями перекладу. Етичні та соціальні аспекти керованості, включаючи можливість упередженості у керуючих параметрах та потенційне посилення мовних нерівностей через нерівномірну підтримку різних мов і культур, потребують окремого дослідження та врегулювання [17, с. 37].

Мета даного дослідження полягає у розробці та дослідженні інноваційних методів керованості машинного перекладу, що адресують виявлені проблеми через створення більш гнучких, ефективних та практично застосовних підходів до контролю процесу перекладу. Основні завдання дослідження включають розробку нових архітектурних рішень для інтеграції різнотипних керуючих сигналів у нейронні моделі перекладу, створення методів динамічної адаптації системи до змінних контекстуальних вимог та розробку ефективних алгоритмів персоналізації перекладу на основі користувацьких преференцій та історії взаємодії.

Дослідження також спрямоване на розробку комплексних метрик оцінювання якості керованого перекладу, що враховують як лінгвістичну точність, так і відповідність керуючим параметрам, а також на створення практичних інструментів та фреймворків, що дозволять розробникам ефективно імплементувати механізми керованості у власних системах. Експериментальна частина роботи передбачає валідацію запропонованих методів на реальних перекладацьких задачах з використанням різноманітних мовних пар та типів текстів, демонструючи практичну цінність розроблених підходів та їх переваги над існуючими рішеннями. Очікувані результати дослідження мають внести суттєвий вклад як у теоретичні основи керованого машинного перекладу, так і у практичне вдосконалення мовних технологій, сприяючи створенню більш адаптивних та користувацько-орієнтованих систем автоматичного перекладу.

2 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

2.1 Формальний опис задачі машинного перекладу

Машинний переклад як формальна задача обчислювальної лінгвістики визначається як процес автоматичного перетворення послідовності символів або токенів з мови-джерела $S = \{s_1, s_2, \ldots, s_\mathrm{m}\}$ у відповідну послідовність в цільовій мові $T = \{t_1, t_2, \ldots, t_\mathrm{n}\}$, що максимізує ймовірність семантичної та синтаксичної еквівалентності між вхідним та вихідним текстами. Математично, задача машинного перекладу формулюється як знаходження оптимального відображення f: S → T, де функція f прагне максимізувати умовну ймовірність P(T|S), що представляє ймовірність генерації послідовності T при заданій вхідній послідовності S. Ця ймовірнісна постановка дозволяє врахувати невизначеність, властиву природним мовам, та множинність можливих правильних перекладів для одного вхідного тексту. Формальний підхід до машинного перекладу також включає визначення простору можливих перекладів Ω, обмежень на довжину та структуру вихідних послідовностей, а також функції оцінювання якості перекладу, що дозволяє порівнювати різні варіанти та вибирати оптимальний результат. Сучасні підходи до формалізації завдання машинного перекладу враховують не лише лексичні та граматичні аспекти перетворення, але й прагматичні, стилістичні та культурні особливості, що робить математичний опис значно складнішим за класичні статистичні моделі [18].

У контексті нейронного машинного перекладу формальна постановка задачі набуває додаткових вимірів через використання неперервних векторних представлень та глибоких нейронних архітектур, що дозволяють моделювати складні нелінійні залежності між елементами вхідного та вихідного тексту. Нейронна модель перекладу визначається як параметрична функція:

$$f\theta(S) = T, \quad (2.1)$$

де θ представляє множину навчальних параметрів моделі, що оптимізуються через процедуру градієнтного спуску на великих паралельних корпусах.

Encoder-decoder архітектура формалізує процес перекладу як двоетапну процедуру: спочатку encoder перетворює вхідну послідовність S у компактне векторне представлення контексту c = encoder(S), а потім decoder генерує вихідну послідовність:

$$T = decoder(c, T < t), \quad (2.2)$$

де T<t позначає частину цільової послідовності, згенеровану на попередніх кроках.

Ця формалізація дозволяє природно інкорпорувати механізми уваги (attention), які модифікують контекстне представлення для кожного кроку генерації: $c\_t = attention(encoder\_states, decoder\_state\_t)$, забезпечуючи більш точне та контекстуально релевантне моделювання процесу перекладу. Функція втрат для нейронного машинного перекладу зазвичай визначається як cross-entropy між передбаченими та справжніми розподілами ймовірностей над словником: $L(\theta) = -\sum_i log\ P(t_i|T < i, S;\ \theta)$, що дозволяє ефективно навчати моделі на великих датасетах.

Керованість у машинному перекладі вимагає розширення базової формальної постановки задачі через введення додаткових керуючих змінних та модифікацію цільової функції для врахування специфічних вимог до характеристик перекладу. Формально, керований машинний переклад можна визначити як задачу знаходження функції:

$$g: (S, C) \rightarrow T, \quad (2.3)$$

де C представляє множину керуючих параметрів, що специфікують бажані характеристики перекладу.

Такі як стиль, формальність, домен або структурні особливості.

Керуючі параметри можуть мати різні форми представлення: дискретні категоричні змінні (наприклад, formal/informal), неперервні числові значення (рівень складності тексту) або структуровані об'єкти (термінологічні словники, синтаксичні шаблони). Цільова функція для керованого перекладу модифікується для одночасної оптимізації якості перекладу та відповідності керуючим параметрам:

$$L\_total = L\_translation + \lambda \cdot L\_control, \quad (2.4)$$

де L_control вимірює ступінь дотримання заданих керуючих обмежень;

λ є гіперпараметром, що балансує між цими двома цілями.

Sequence controls як спеціалізований тип керування формалізуються через додаткові токени в послідовності, що несуть метаінформацію про бажані характеристики генерації: S' = [control_tokens] ⊕ S, де ⊕ позначає операцію конкатенації послідовностей [19, с. 9].

Таблиця 2.1 – Формальні компоненти задачі машинного перекладу

Компонент	Математичне позначення	Опис	Приклад
Вхідна послідовність	$S = \{s_1, s_2, ..., s_m\}$	Токенізований текст мови-джерела	["Hello", "world", "!"]
Цільова послідовність	$T = \{t_1, t_2, ..., t_n\}$	Бажаний переклад у цільовій мові	["Привіт", "світ", "!"]
Функція перекладу	$f: S \rightarrow T$	Відображення між мовами	Нейронна модель
Ймовірнісна модель	$P(T\|S)$	Умовна ймовірність перекладу	Softmax розподіл
Керуючі параметри	$C = \{c_1, c_2, ..., c_k\}$	Додаткові обмеження на переклад	[formal, technical]
Керована функція	$g: (S, C) \rightarrow T$	Переклад з урахуванням керування	Модифікована архітектура
Функція втрат	$L(\theta) = -\sum \log P(t_i\|T<i, S; \theta)$	Критерій оптимізації параметрів	Cross-entropy loss
Sequence controls	S' = [tokens] ⊕ S	Послідовність з керуючими токенами	["<formal>", "Hello", "world"]

Практична реалізація формальної постановки задачі машинного перекладу вимагає конкретизації архітектурних рішень, алгоритмів навчання та методів оцінювання, що дозволяють перетворити теоретичні концепції у функціональні системи перекладу. Transformer архітектура, яка є основою більшості сучасних систем, формалізує процес перекладу через систему самоуваги та cross-attention механізмів:

$$Attention(Q, K, V) = softmax(QK\hat{}T/\sqrt{d\_k})V, \quad (2.5)$$

де Q, K, V представляють матриці запитів, ключів та значень відповідно.

Multi-head attention розширює цю формалізацію через паралельне обчислення кількох attention heads:

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W\hat{}O, \quad (2.6)$$

де кожен head_i = Attention(QW_i^Q, KW_i^K, VW_i^V).

Позиційне кодування вводиться для збереження інформації про порядок токенів у послідовності: PE(pos, 2i) = sin(pos/10000^(2i/d_model)), PE(pos, 2i+1) = cos(pos/10000^(2i/d_model)). Процедура навчання формалізується як ітеративна оптимізація параметрів θ через градієнтний спуск:

$$\theta\_\{t+1\} = \theta\_t - \eta\nabla\_\theta\, L(\theta\_t), \quad (2.7)$$

де η представляє швидкість навчання.

Інференція у навчених моделях може здійснюватися через різні стратегії декодування, включаючи greedy search: $t\_i = argmax\, P(t|T < i, S;\, \theta)$, beam search для збереження кількох найймовірніших гіпотез, або sampling методи для генерації більш різноманітних перекладів. Ці формальні компоненти разом утворюють повну математичну специфікацію

задачі машинного перекладу, що дозволяє систематично досліджувати та порівнювати різні підходи до її розв'язання [20, с. 170].

Інтеграція керованості у формальну постановку задачі машинного перекладу вимагає додаткових теоретичних розробок для забезпечення коректної взаємодії між базовими механізмами перекладу та керуючими сигналами без погіршення загальної якості системи. Формалізація керуючих механізмів включає визначення простору керуючих параметрів C та способів їх інкорпорації у процес обчислення ймовірностей перекладу: $P(T|S,C) = \prod_i P(t_i|T < i, S, C;\ \theta)$. Адаптивні керуючі механізми можуть бути формалізовані через динамічну модифікацію архітектури або параметрів моделі на основі поточного контексту: $\theta\_adaptive = f\_adaptation(\theta\_base, C, context)$, що дозволяє системі автоматично налаштовуватися до специфічних вимог перекладу. Метрики оцінювання керованого перекладу розширюють традиційні показники якості через додаткові компоненти, що вимірюють відповідність керуючим параметрам:

$$Score\_total = \alpha \cdot BLEU(T, T\_ref) + \beta \cdot Control\_adherence(T, C), \quad (2.8)$$

де α та β є ваговими коефіцієнтами, що визначають відносну значущість якості перекладу та дотримання керуючих інструкцій.

Така комплексна формалізація забезпечує теоретичну основу для розробки, навчання та оцінювання керованих систем машинного перекладу, що здатні забезпечувати високу якість перекладу при одночасному дотриманні специфічних користувацьких вимог та контекстуальних обмежень [21].

Представлена формалізація створює цілісну теоретичну основу для розуміння та подальшого розвитку методів машинного перекладу, особливо в контексті керованих систем, що здатні адаптуватися до різноманітних користувацьких потреб та контекстуальних вимог (рисунок 2.1).

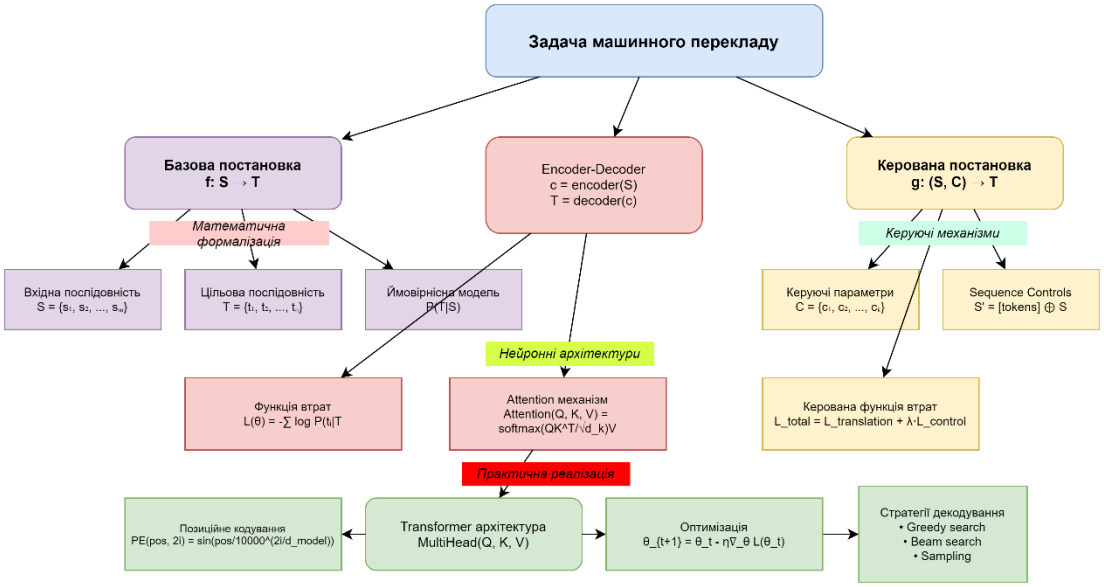

Рисунок 2.1 – Структурна схема формального опису задачі машинного перекладу

Багаторівнева структура формального опису, що охоплює від базових математичних принципів до конкретних архітектурних рішень, дозволяє систематично досліджувати взаємозв'язки між різними компонентами системи перекладу та оптимізувати їх взаємодію для досягнення максимальної ефективності. Ця формалізація також забезпечує методологічну основу для емпіричного дослідження та порівняння різних підходів до керованого машинного перекладу, що є предметом подальших розділів даної роботи.

2.2 Метрики оцінювання якості перекладу

Оцінювання якості машинного перекладу представляє собою багатоаспектну задачу, що вимагає комплексного підходу до вимірювання відповідності згенерованого тексту еталонному перекладу з урахуванням лексичної точності, граматичної правильності, семантичної адекватності та прагматичної доречності. Автоматичні метрики оцінювання розроблялися

як альтернатива трудомістким та суб'єктивним методам людського оцінювання, прагнучи забезпечити швидке, об'єктивне та відтворюване вимірювання якості перекладу для великих обсягів тексту.

Фундаментальним принципом більшості автоматичних метрик є припущення, що якість машинного перекладу можна оцінити через порівняння з одним або кількома еталонними перекладами, виконаними професійними перекладачами, що дозволяє кількісно виразити ступінь близькості автоматично згенерованого тексту до людського стандарту. Однак це припущення має певні обмеження, оскільки для одного вихідного тексту може існувати множина рівноцінних перекладів, що відрізняються лексичним вибором, синтаксичною структурою або стилістичними особливостями, але зберігають семантичну еквівалентність оригіналу.

Сучасні дослідження у галузі автоматичного оцінювання спрямовані на розробку більш досконалих метрик, здатних врахувати цю варіативність та забезпечити більш точну кореляцію з людськими судженнями про якість перекладу [22].

Метрика BLEU (Bilingual Evaluation Understudy) стала першою широко прийнятою автоматичною мірою якості машинного перекладу та залишається одним з найвпливовіших стандартів у галузі, незважаючи на певні концептуальні обмеження та критику з боку дослідницької спільноти. BLEU базується на обчисленні точності n-грам між кандидатом перекладу та еталонними варіантами, де n-грама визначається як послідовність n послідовних слів у тексті, що дозволяє оцінити збіг на різних рівнях гранулярності від окремих слів до фраз та речень. Формально, BLEU обчислюється як геометричне середнє точностей для n-грам різної довжини (зазвичай від 1 до 4), помножене на штрафний коефіцієнт за надмірну стислість:

$$BLEU = BP \times exp(\sum_{i=1}^{N} w_i \log p_i), \qquad (2.8)$$

де p_i представляє точність для і-грам;

w_i є ваговими коефіцієнтами (зазвичай 1/N);

BP (brevity penalty) карає переклади, що є суттєво коротшими за еталонні [23, с. 108].

Незважаючи на свою популярність, BLEU має суттєві недоліки, включаючи надмірну залежність від поверхневого лексичного збігу, нездатність враховувати синоніми та парафрази, відсутність урахування порядку слів на дальніх відстанях та схильність до завищених оцінок для коротких речень. Ці обмеження особливо проблематичні для оцінювання перекладів між морфологічно багатими мовами або при роботі з творчими та художніми текстами, де лексична варіативність є природною та бажаною характеристикою [24, с. 75].

METEOR (Metric for Evaluation of Translation with Explicit ORdering) була розроблена як відповідь на обмеження BLEU та впроваджує більш софістиковані методи вирівнювання слів між кандидатом та еталонним перекладом, включаючи врахування синонімів, стемінгу та парафраз для більш точного вимірювання семантичної подібності. Основна інновація METEOR полягає у використанні WordNet та інших лексичних ресурсів для ідентифікації семантично еквівалентних слів, що не співпадають на поверхневому рівні, дозволяючи метриці розпізнавати правильні переклади, які використовують синоніми замість точних лексичних відповідників з еталону. Процедура обчислення METEOR включає кілька етапів: спочатку створюється вирівнювання між словами кандидата та еталону на основі точних збігів, потім додаються вирівнювання на основі словникових синонімів, після чого враховуються стемінговані форми слів для роботи з морфологічними варіаціями. Фінальна оцінка METEOR обчислюється як гармонічне середнє точності та повноти з додатковим штрафом за фрагментацію, що враховує кількість суміжних блоків вирівняних слів:

$$METEOR = \frac{(1-Penalty) \times (Precision \times Recall)}{(\alpha \times Precision + (1-\alpha) \times Recall)}, \quad (2.9)$$

де Penalty = γ × (Chunks/Matches)^β відображає ступінь порушення порядку слів.

Ця метрика демонструє кращу кореляцію з людськими оцінками порівняно з BLEU, особливо для мов з багатою морфологією та у випадках, коли переклади використовують лексичні варіації, що семантично еквівалентні еталонним формулюванням [25].

Розвиток нейронних методів обробки природної мови призвів до появи семантичних метрик оцінювання, таких як BERTScore, що використовують попередньо навчені мовні моделі для обчислення семантичної подібності між перекладами на рівні контекстуалізованих векторних представлень.

BERTScore революціонізувала підхід до автоматичного оцінювання через відмову від поверхневого лексичного порівняння на користь глибокого семантичного аналізу, використовуючи BERT або інші трансформерні моделі для генерації контекстуалізованих ембедингів кожного токена у кандидаті та еталонному перекладі. Метрика обчислює косинусну подібність між відповідними токенами у векторному просторі BERT, дозволяючи ідентифікувати семантичну еквівалентність навіть при значних лексичних відмінностях:

$$BERTScore\_F = \frac{2 \times (BERTScore\_P \times BERTScore\_R)}{(BERTScore\_P + BERTScore\_R)}, \quad (2.9)$$

де precision та recall обчислюються як максимальна подібність між токенами кандидата та еталону відповідно.

ChrF (Character n-gram F-score) представляє альтернативний підхід, що оперує на рівні символів замість слів, що робить її особливо ефективною для морфологічно складних мов та мов з агглютинативною структурою, де словоформи можуть значно варіюватися при збереженні семантичного змісту. ROUGE (Recall-Oriented Understudy for Gisting Evaluation), хоча первинно розроблена для оцінювання автоматичного реферування, також

знаходить застосування у машинному перекладі, особливо для оцінювання повноти та покриття змісту оригінального тексту у перекладі [26].

Специфічні виклики оцінювання керованого машинного перекладу вимагають розробки нових метрик та методологій, здатних одночасно вимірювати якість перекладу та ступінь відповідності заданим керуючим параметрам, що створює багатокритеріальну задачу оптимізації з потенційними конфліктами між різними цілями.

Комплексні метрики для керованого перекладу повинні враховувати не лише традиційні аспекти якості, такі як адекватність та плавність, але й специфічні характеристики, задані керуючими параметрами, включаючи стилістичну відповідність, термінологічну консистентність, структурні особливості та прагматичну доречність результату.

Розробка таких метрик ускладнюється необхідністю кількісного вимірювання якісних характеристик тексту, таких як формальність, емоційне забарвлення або культурна адаптація, що традиційно вважалися суб'єктивними та важко формалізованими аспектами мовної продукції. Автоматичне оцінювання дотримання керуючих параметрів може включати використання окремих класифікаторів стилю, аналізаторів тональності, засобів виявлення домену або спеціалізованих нейронних моделей, навчених розпізнавати специфічні характеристики тексту, що дозволяє об'єктивно вимірювати успішність керування процесом перекладу.

Інтегральні оцінки якості керованого перекладу часто формулюються як зважена комбінація традиційних метрик якості та показників відповідності керуючим параметрам:

$$Score\_total = \alpha \times Quality\_score + \beta \times Control\_adherence + \gamma \times Consistency\_penalty, \quad (2.10)$$

де вагові коефіцієнти визначають відносну значущість різних аспектів оцінювання залежно від специфіки перекладацької задачі [27, с. 249].

2.3 Архітектури та методи машинного перекладу

Архітектурні підходи до машинного перекладу пройшли значну еволюцію від простих правил-орієнтованих систем до складних нейронних архітектур, що здатні моделювати глибокі семантичні та синтаксичні залежності між мовами з безпрецедентною точністю та гнучкістю. Encoder-decoder архітектура стала фундаментальною парадигмою для сучасного нейронного машинного перекладу, представляючи елегантне розв'язання проблеми перетворення послідовностей змінної довжини через розділення процесу на два концептуально різні етапи: кодування вхідної інформації у компактне внутрішнє представлення та декодування цього представлення у цільову мову.

Encoder у цій архітектурі функціонує як система стиснення з втратами, що перетворює вхідну послідовність токенів у векторне представлення фіксованої розмірності, яке теоретично містить всю суттєву інформацію, необхідну для генерації правильного перекладу, включаючи лексичні значення, граматичні відношення, семантичні ролі та прагматичні нюанси. Decoder, у свою чергу, реалізує процес автогресивної генерації, де кожен наступний токен у цільовій послідовності передбачається на основі закодованого контексту та всіх попередньо згенерованих токенів, створюючи послідовний та контекстуально узгоджений переклад. Ця архітектурна парадигма дозволила значно покращити якість машинного перекладу порівняно з попередніми статистичними методами, особливо для довгих речень та складних синтаксичних конструкцій [28, с. 31].

Революційне впровадження механізмів уваги (attention mechanisms) кардинально змінило ландшафт нейронного машинного перекладу, вирішивши фундаментальну проблему "вузького горлечка" у традиційних encoder-decoder моделях, де вся інформація про вхідну послідовність мала бути стиснута у єдиний контекстний вектор фіксованої розмірності. Attention механізм дозволяє decoder динамічно фокусуватися на різних

частинах вхідної послідовності на кожному кроці генерації, створюючи адаптивне контекстуальне представлення, що враховує специфічні інформаційні потреби для генерації поточного токена. Математично, attention обчислюється як зважена сума всіх encoder станів, де ваги визначаються через функцію схожості між поточним decoder станом та кожним encoder станом:

$$\alpha\_\{ij\} = exp(e\_\{ij\})/\sum\_\{k = 1\}\^\{T\_x\} exp(e\_\{ik\}), \quad (2.11)$$

де e_{ij} представляє оцінку релевантності j-го encoder стану для i-го кроку декодування.

Цей механізм не лише покращив якість перекладу, особливо для довгих речень, але й забезпечив інтерпретованість моделей через візуалізацію attention weights, що дозволяє аналізувати, на які частини вхідного тексту модель "звертає увагу" при генерації кожного слова перекладу. Різні варіанти attention механізмів, включаючи additive attention, multiplicative attention та self-attention, пропонують альтернативні способи обчислення схожості та агрегації інформації, кожен з власними обчислювальними характеристиками та областями застосування [29, с. 70].

Архітектура Transformer представляє кульмінацію розвитку attention-based підходів, радикально переосмисливши принципи побудови нейронних мереж для обробки послідовностей через повну відмову від рекурентних та згорткових компонентів на користь виключно attention механізмів. Фундаментальною інновацією Transformer є концепція self-attention, що дозволяє кожному елементу послідовності безпосередньо взаємодіяти з усіма іншими елементами, незалежно від їх позиційної відстані, створюючи повнозв'язну мережу інформаційних потоків всередині послідовності. Multi-head attention розширює цю концепцію через паралельне обчислення кількох attention функцій з різними параметрами:

$$MultiHead(Q,K,V) = Concat(head_1,\ldots,head\_h)W\hat{}O, \quad (2.12)$$

де кожен head_i = Attention(QW_i^Q, KW_i^K, VW_i^V), дозволяючи моделі одночасно фокусуватися на різних типах відношень між словами, таких як синтаксичні залежності, семантичні асоціації та дискурсивні зв'язки.

Позиційне кодування у Transformer архітектурі компенсує відсутність природної чутливості до порядку слів через додавання спеціальних векторів до input embeddings:

$$PE(pos,2i) = sin(pos/10000\hat{}\{2i/d\_\{model\}\}), PE(pos,2i+1) = cos(pos/10000\hat{}\{2i/d\_\{model\}\}). \quad (2.13)$$

Що дозволяє моделі розрізняти слова на основі їх позиції у послідовності (рисунок 2.2). Архітектура також включає резидуальні з'єднання та layer normalization для стабілізації навчання глибоких мереж, а також feed-forward підмережі для нелінійного перетворення представлень на кожному рівні [23].

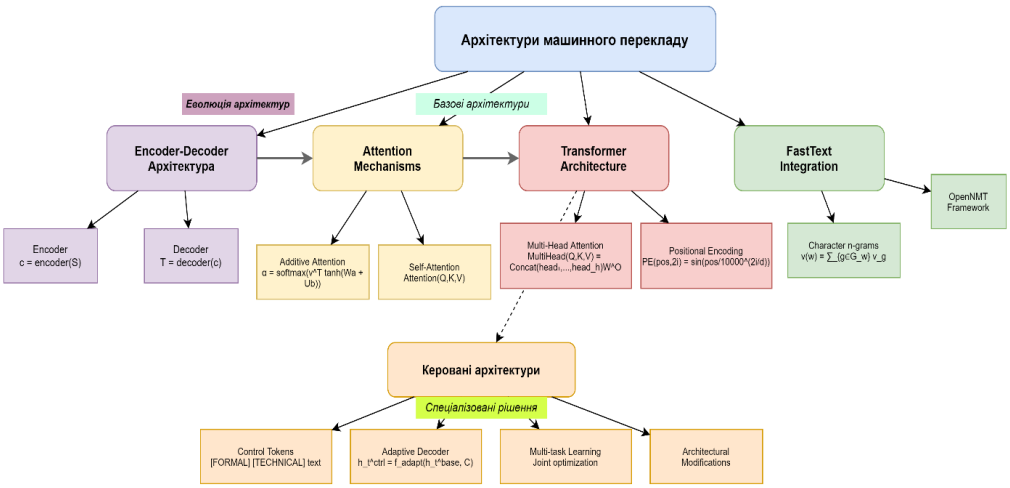

Рисунок 2.2 – Еволюція архітектур машинного перекладу та їх компонентів

Інтеграція попередньо навчених векторних представлень, особливо FastText embeddings, у архітектури машинного перекладу представляє потужний метод покращення якості моделей через використання зовнішніх лінгвістичних знань, отриманих з великих монолінгвальних корпусів. FastText розширює традиційні word embeddings через врахування внутрішньої структури слів на рівні символьних n-грам, що робить ці представлення особливо ефективними для морфологічно багатих мов та слів, що не зустрічалися під час навчання (out-of-vocabulary words). Математично, FastText представляє кожне слово як суму векторів його символьних n-грам:

$$v(w) = \sum\_\{g \in G\_w\} v\_g, \quad (2.14)$$

де G_w є множиною всіх n-грам слова w, що дозволяє генерувати представлення для невідомих слів на основі їх морфологічної структури.

Інтеграція таких embeddings у encoder-decoder архітектури може здійснюватися через ініціалізацію embedding шарів попередньо навченими векторами з подальшим fine-tuning на паралельних корпусах, або через фіксацію цих представлень для збереження зовнішніх лінгвістичних знань. OpenNMT фреймворк надає гнучкі інструменти для такої інтеграції, дозволяючи дослідникам експериментувати з різними стратегіями використання попередньо навчених embeddings, включаючи multi-lingual embeddings для cross-lingual transfer learning та domain-specific embeddings для адаптації до спеціалізованих текстів. Ця інтеграція особливо ефективна для low-resource мовних пар, де обмежений обсяг паралельних даних може бути компенсований багатими монолінгвальними представленнями [26].

Спеціалізовані архітектурні модифікації для керованого машинного перекладу вимагають ретельного проектування механізмів інкорпорації керуючих сигналів без порушення базової функціональності системи перекладу та зі збереженням можливості ефективного навчання на великих

датасетах. Контрольовані токени (control tokens) представляють найпростіший підхід до керованості, коли спеціальні маркери додаються на початок вхідної послідовності для специфікації бажаних характеристик перекладу: [FORMAL] [TECHNICAL] source_sentence, що дозволяє моделі навчитися асоціювати ці токени з відповідними стилістичними або доменними особливостями у цільовому тексті. Більш досконалі підходи включають архітектурні модифікації encoder або decoder компонентів через додаткові attention механізми, що спеціально призначені для обробки керуючої інформації, або через інтеграцію додаткових embedding шарів для кодування різних типів метаданих про бажаний переклад.

Адаптивні decoder архітектури можуть динамічно модифікувати свої параметри або attention patterns на основі керуючих сигналів:

$$h\_t\^\{controlled\} = f\_\{adaptation\}(h\_t\^\{base\}, control\_context). \quad (2.15)$$

Створюючи контекстно-залежні механізми генерації. Multi-task learning підходи дозволяють одночасно навчати модель на основній задачі перекладу та допоміжних задачах класифікації стилю або домену, створюючи багатофункціональні системи, здатні не лише перекладати, але й контролювати характеристики генерованого тексту через shared representations та joint optimization objectives [28].

3 АНАЛІЗ ДАТАСЕТІВ ТА ДАНИХ

3.1 Огляд використаних датасетів

Вибір та характеристики датасетів для навчання системи машинного перекладу відіграють фундаментальну роль у визначенні якості, робастності та загальної ефективності кінцевої моделі, оскільки якість та різноманітність тренувальних даних безпосередньо впливають на здатність системи генерувати адекватні переклади для широкого спектру текстових жанрів та доменів. Паралельні корпуси, що складаються з текстів, професійно перекладених між мовами-джерелом та цільовими мовами, представляють основний тип даних для supervised навчання систем нейронного машинного перекладу, забезпечуючи моделі конкретні приклади правильних відповідностей між мовними конструкціями. Якість паралельних корпусів визначається не лише точністю перекладів, але й такими характеристиками, як рівень вирівнювання на рівні речень, консистентність термінології, стилістична однорідність та представленість різних текстових жанрів від новинних статей до технічної документації.

Доменна різноманітність у тренувальних даних є суттєвим фактором для створення універсальних систем перекладу, здатних ефективно працювати з текстами різної тематики та стилістики, тоді як доменна спеціалізація може бути бажаною для створення систем, оптимізованих для конкретних галузей або типів контенту. Сучасні дослідження також підкреслюють значення збалансованості датасетів щодо довжини речень, складності синтаксичних конструкцій та частоти зустрічальності різних граматичних феноменів для забезпечення рівномірного навчання моделі на всіх аспектах мовної структури [30].

WikiMatrix представляє один з найбільших та найрізноманітніших паралельних корпусів, створений через автоматичне вирівнювання статей Вікіпедії між різними мовними версіями з використанням передових

алгоритмів виявлення паралельних речень та подальшою ретельною фільтрацією для забезпечення високої якості вирівнювання. Цей корпус характеризується винятковою доменною різноманітністю, охоплюючи широкий спектр тем від історії та географії до науки та культури, що робить його особливо цінним для навчання універсальних систем перекладу, здатних ефективно працювати з енциклопедичними та довідковими текстами. Методологія створення WikiMatrix включає використання багатомовних sentence embeddings для ідентифікації семантично еквівалентних речень між мовними версіями статей, з подальшою фільтрацією на основі мовних детекторів, показників подібності та эвристичних правил для усунення некоректних вирівнювань та дублікатів.

Англо-українська частина WikiMatrix містить мільйони паралельних речень, що покривають широкий діапазон лексики та граматичних конструкцій, характерних для енциклопедичного стилю, включаючи складні іменникові групи, пасивні конструкції, наукову термінологію та формальний регістр мови. Однак варто відзначити, що автоматичне походження корпусу може призводити до певних артефактів у вирівнюванні та появи невеликої кількості некоректних паралельних пар, що вимагає додаткової постобробки та валідації даних перед використанням у навчанні моделей [22].

XLEnt корпус представляє спеціалізований датасет, орієнтований на покращення якості машинного перекладу для текстів з високою інформаційною щільністю та складними синтаксичними структурами, що часто зустрічаються у академічній літературі, технічній документації та професійних публікаціях. Цей корпус було створено через ретельну курацію та професійний переклад відібраних текстів, що забезпечує винятково високу якість паралельних даних з мінімальною кількістю помилок вирівнювання та переклад, що робить його особливо цінним для тонкого налаштування моделей та валідації їх ефективності на складних текстах. XLEnt характеризується збалансованим представленням різних

синтаксичних конструкцій, включаючи підрядні речення, складні предикати, еліптичні конструкції та різноманітні типи анафоричних відношень, що дозволяє моделям навчатися ефективно обробляти структурну складність природної мови. Англо-українська секція XLEnt включає тексти з галузей права, медицини, техніки та гуманітарних наук, забезпечуючи експозицію до спеціалізованої термінології та дискурсивних патернів, характерних для професійного та академічного спілкування. Додатковою перевагою XLEnt є наявність детальних метаданих для кожної паралельної пари, включаючи інформацію про домен, складність тексту, тип дискурсу та якість перекладу, що дозволяє проводити більш детальний аналіз ефективності моделей на різних типах текстів [31].

Таблиця 3.1 – Характеристики використаних датасетів для машинного перекладу

Датасет	Розмір (пари речень)	Домени	Метод створення	Особливості
WikiMatrix	~2.5M EN-UK	Енциклопедичні статті	Автоматичне вирівнювання Вікіпедії	Висока доменна різноманітність, формальний стиль
XLEnt	~800K EN-UK	Академічні, технічні тексти	Професійний переклад	Висока якість, складні конструкції
QED	~220K EN-UK	Освітні лекції, виступи	Переклад субтитрів	Розмовний стиль, усна мова
Tatoeba	~15K EN-UK	Загальна лексика	Краудсорсинг	Короткі речення, базова лексика

QED (QCRI Educational Domain) корпус займає унікальну нішу серед паралельних датасетів завдяки своєму фокусу на освітньому контенті, зокрема перекладах субтитрів до освітніх відеолекцій та презентацій, що робить його безцінним ресурсом для навчання моделей роботі з розмовною мовою та неформальними дискурсивними стилями. Цей датасет відображає особливості усного мовлення, перенесеного у письмову форму через субтитри, включаючи неповні речення, хезитації, повтори, розмовні маркери та інші характеристики спонтанного дискурсу, що значно

відрізняється від формального письмового тексту, типового для більшості інших корпусів. Методологія створення QED базується на професійному перекладі субтитрів освітніх відео з платформ онлайн-навчання, де перекладачі мали завдання не лише передати семантичний зміст, але й адаптувати текст до обмежень субтитрування, включаючи синхронізацію з відео та читабельність для глядачів.

Англо-українська частина QED містить переклади лекцій з широкого спектру дисциплін, від точних наук та інженерії до гуманітарних предметів та соціальних наук, забезпечуючи експозицію до різноманітної наукової та освітньої лексики у контексті пояснень та дидактичного дискурсу. Специфічною характеристикою QED є наявність часових міток та інформації про сегментацію, що дозволяє аналізувати, як структура усного мовлення впливає на стратегії перекладу та які адаптації необхідні для ефективної передачі освітнього контенту між мовами.

Tatoeba корпус, хоча і значно менший за обсягом порівняно з іншими датасетами, представляє цінний ресурс для навчання та тестування базових перекладацьких навичок завдяки своїй фокусу на короткі, граматично прості речення, що покривають фундаментальну лексику та основні синтаксичні патерни обох мов. Цей краудсорсинговий проект залучає волонтерів з усього світу для створення та перевірки перекладів, що забезпечує високий рівень якості та природності мовних пар, особливо для базових комунікативних ситуацій та повсякденної лексики.

Tatoeba характеризується ретельною модерацією та багаторівневою перевіркою якості, коли кожен переклад проходить валідацію носіями мови та експертами з лінгвістики, що мінімізує ймовірність граматичних помилок або неприродних формулювань у фінальному датасеті. Англо-українська секція Tatoeba включає речення, що представляють різні комунікативні функції, від простих повідомлень та питань до вираження емоцій та думок, забезпечуючи збалансоване покриття базової граматики та лексики обох мов. Незважаючи на свій відносно невеликий розмір, Tatoeba відіграє

суттєву роль у комплексній стратегії навчання, особливо для початкових етапів тренування моделі, де простота та якість даних можуть бути більш цінними за їх кількість, а також для тестування здатності моделі обробляти фундаментальні мовні конструкції без ускладнень, пов'язаних з доменною специфікою або стилістичними особливостями більш складних корпусів [32].

3.2 Підготовка та обробка даних

Процес підготовки та обробки паралельних корпусів для навчання систем машинного перекладу представляє багатоетапну процедуру, що включає токенізацію, нормалізацію, очищення та структурування даних з метою оптимізації якості тренувального матеріалу та забезпечення ефективного навчання нейронних моделей. Токенізація як фундаментальний крок препроцесингу передбачає розбиття неструктурованого тексту на дискретні одиниці обробки, зазвичай слова або субслова, що дозволяє моделі оперувати з конкретними лінгвістичними елементами замість неперервних символьних послідовностей.

OpenNMT tokenizer, використаний у даному дослідженні, реалізує агресивну стратегію токенізації, що включає розділення пунктуації, нормалізацію регістру символів, сегментацію чисел та обробку алфавітних переходів для забезпечення консистентного представлення тексту незалежно від його первинного форматування.

Конфігураційні параметри токенізатора, включаючи режим агресивної обробки (aggressive mode), анотування роз'єднувачів (joiner_annotate), сегментацію числових послідовностей (segment_numbers) та обробку змін алфавіту (segment_alphabet_change), визначають специфічні аспекти токенізації та впливають на кінцеву гранулярність представлення тексту. Ця детальна токенізація особливо суттєва для морфологічно багатих мов, таких

як українська, де словоформи можуть значно варіюватися через флективні зміни, а правильна сегментація дозволяє моделі краще вивчати морфологічні закономірності та граматичні відношення між елементами речення [33].

Очищення та фільтрація паралельних даних становить основну передумову для створення високоякісного тренувального датасету, що передбачає систематичне усунення некоректних, неповних або неадекватно вирівняних паралельних пар, які можуть негативно впливати на процес навчання та кінцеву якість моделі. Алгоритм очищення включає перевірку наявності алфавітно-цифрових символів у кожному реченні через регулярні вирази, що дозволяє відфільтрувати порожні рядки, послідовності лише з пунктуації або пробілів, а також артефакти, що могли виникнути під час попередньої обробки корпусів.

Процедура нормалізації тексту включає заміну символів нового рядка на пробіли та усунення зайвих пробілових символів, що забезпечує консистентне форматування та запобігає появі непередбачуваних символьних послідовностей під час навчання моделі. Додатковим етапом фільтрації є перевірка паралельності корпусів через порівняння кількості речень у файлах мови-джерела та цільової мови, що дозволяє виявити та усунути розбіжності у вирівнюванні, які могли виникнути через помилки у попередній обробці або корупцію даних. Використання pandas DataFrame для структурування та маніпулювання паралельними даними забезпечує ефективну обробку великих обсягів тексту та дозволяє застосовувати векторизовані операції для швидкого очищення та аналізу корпусів.

Стратегічне розділення підготовлених даних на тренувальну та тестову вибірки представляє методологічно значущий аспект експериментального дизайну, що впливає як на процес навчання моделі, так і на достовірність оцінювання її ефективності на незалежних даних. Співвідношення розподілу 99% тренувальних даних до 1% тестових відображає специфіку навчання глибоких нейронних мереж, які потребують

великих обсягів даних для ефективного навчання параметрів, особливо у контексті Transformer архітектур з мільйонами параметрів. Процедура випадкового перемішування (random shuffling) з фіксованим насінням (random seed) забезпечує відтворюваність експериментів та гарантує, що розподіл даних між тренувальною та тестовою вибірками є статистично репрезентативним щодо оригінального корпусу без систематичних упереджень.

Синхронне перемішування паралельних послідовностей у мові-джерелі та цільовій мові є принциповим для збереження коректного вирівнювання між перекладними парами, оскільки порушення цього вирівнювання призвело б до створення некоректних навчальних прикладів та значного погіршення якості моделі. Валідація правильності розподілу включає перевірку збереження паралельності між мовами після перемішування та підтвердження того, що тестова вибірка містить репрезентативний зріз різних типів речень, доменів та лінгвістичних конструкцій, присутніх в оригінальному корпусі [34, с. 9].

Збереження оброблених даних у структурованому форматі з використанням UTF-8 кодування забезпечує сумісність з міжнародними символьними наборами та підтримку багатомовних корпусів, що є особливо суттєвим для роботи з неанглійськими мовами, включаючи кириличні тексти українською мовою. Процедура збереження включає автоматичне перетворення всіх текстових даних у нижній регістр для нормалізації та зменшення розмірності словника, що може покращити ефективність навчання моделі через зменшення кількості унікальних токенів та фокусування на семантичних характеристиках слів замість ортографічних варіацій. Структура збережених файлів передбачає окремі файли для кожної мови та кожного розділення (тренувальні/тестові), що дозволяє гнучко керувати датасетами під час експериментування та забезпечує можливість незалежного завантаження різних частин корпусу залежно від потреб конкретного експерименту. Додатковою перевагою такої організації даних

є можливість ефективного паралельного завантаження файлів під час навчання, що може значно прискорити процес тренування на багатопроцесорних системах. Метадані про обсяг кожного датасету, включаючи кількість речень у тренувальній та тестовій вибірках, зберігаються для подальшого аналізу та забезпечення прозорості експериментального процесу, дозволяючи дослідникам точно відтворити умови навчання та валідувати результати на ідентичних даних [32].

Інтеграція множинних датасетів через процедуру об'єднання (merging) представляє складну задачу балансування між збільшенням різноманітності тренувальних даних та збереженням когерентності навчального процесу, що вимагає ретельного аналізу характеристик кожного корпусу та їх потенційної сумісності. Стратегія послідовного об'єднання датасетів WikiMatrix, XLEnt, QED та Tatoeba створює комплексний тренувальний корпус, що охоплює широкий спектр текстових жанрів, стилістичних регістрів та доменних специфічностей, від формальних енциклопедичних статей до розмовних освітніх матеріалів та базової повсякденної лексики.

Процедура tokenize_multiple_datasets забезпечує консистентну обробку всіх корпусів з використанням ідентичних параметрів токенізації, що гарантує сумісність різних джерел даних та дозволяє моделі навчатися на уніфікованому представленні незалежно від походження конкретних текстових фрагментів. Збалансування впливу різних датасетів у об'єднаному корпусі здійснюється через їх природні розміри, де більші корпуси як WikiMatrix домінують у навчальному процесі, тоді як менші спеціалізовані датасети як Tatoeba надають специфічні лінгвістичні знання та покривають прогалини у доменному покритті.

Ця інтегративна стратегія дозволяє створити робастну модель, здатну ефективно працювати з різноманітними типами текстів, зберігаючи при цьому спеціалізовані навички для обробки конкретних доменів та стилістичних особливостей, що є особливо цінним для практичних застосувань системи машинного перекладу в реальних умовах [35].

4 РОЗРОБКА СИСТЕМИ КЕРОВАНОГО МАШИННОГО ПЕРЕКЛАДУ

4.1 Архітектура запропонованої системи

Архітектура розробленої системи машинного перекладу представляє собою багатокомпонентну екосистему, що інтегрує передові методи нейронного перекладу з практичними рішеннями для створення повнофункціональної платформи, здатної забезпечувати високоякісний двонаправленний переклад між англійською та українською мовами. Система базується на модульному підході, де кожен компонент виконує специфічну функцію у загальному процесі перекладу, від початкової обробки тексту до генерації фінального результату через веб-інтерфейс користувача.

Центральним елементом архітектури є Transformer-based модель машинного перекладу, навчена на комплексному датасеті з використанням OpenNMT фреймворку, що забезпечує надійну основу для генерації високоякісних перекладів. Система реалізує принцип розділення відповідальностей (separation of concerns), де логіка обробки тексту, нейронні моделі перекладу та користувацький інтерфейс функціонують як незалежні модулі з чітко визначеними інтерфейсами взаємодії. Така архітектурна організація забезпечує гнучкість у розробці, тестуванні та масштабуванні системи, дозволяючи незалежно модифікувати окремі компоненти без впливу на функціональність інших частин платформи. Додатковою перевагою модульної архітектури є можливість легкого розширення системи новими мовними парами або функціональними можливостями через додавання нових модулів або модифікацію існуючих компонентів [36].

Інтеграція FastText векторних представлень у архітектуру Transformer моделі представляє один з найсуттєвіших технічних внесків розробленої

системи, що дозволяє ефективно поєднати переваги попередньо навчених word embeddings з потужністю attention-based архітектур для досягнення покращеної якості перекладу. FastText embeddings, навчені на великих монолінгвальних корпусах з розмірністю 200 та мінімальною частотою слів 5, забезпечують багате семантичне представлення слів через врахування їх внутрішньої морфологічної структури на рівні символьних n-грам, що особливо цінно для морфологічно складних мов як українська. Процедура навчання embeddings включає автоматичне збагачення словника додатковими токенами, виявленими у тестових даних, що забезпечує покриття рідкісних слів та неологізмів, які могли не зустрічатися у первинному тренувальному корпусі.

Архітектурна інтеграція FastText здійснюється через ініціалізацію embedding шарів Transformer моделі попередньо навченими векторами з подальшим fine-tuning на паралельних даних, що дозволяє зберегти семантичні знання з монолінгвальних корпусів та адаптувати їх до специфіки міжмовного перекладу. Така стратегія інтеграції показує особливу ефективність для low-resource мовних пар, де обмежений обсяг паралельних даних може бути компенсований багатими монолінгвальними представленнями, отриманими з значно більших текстових колекцій [37].

Модульна структура програмного забезпечення системи організована навколо принципів чистої архітектури (clean architecture) з чітким розділенням між доменною логікою, інфраструктурними компонентами та інтерфейсами користувача, що забезпечує високу підтримуваність коду та можливість незалежного тестування окремих компонентів. Основні модулі системи включають core.processing_text для обробки та токенізації текстових даних, models.textEmbeddings для роботи з векторними представленнями, models.textTransformer для ініціалізації та управління Transformer моделями, а також головний модуль main.py, що реалізує веб-інтерфейс через Streamlit фреймворк. Модуль обробки тексту реалізує функції init_tokenizer для створення токенізатора з заданою конфігурацією,

tokenize_multiple_datasets для пакетної обробки множинних корпусів, clean_empty_sentences для фільтрації некоректних даних та split_data для розділення на тренувальні та тестові вибірки з підтримкою відтворюваності через фіксовані насіння випадковості. Цей модульний підхід дозволяє легко адаптувати систему до нових вимог, наприклад, додавання нових методів токенізації або зміна стратегій обробки даних, без необхідності модифікації інших частин системи [35].

Практична реалізація користувацького інтерфейсу через Streamlit демонструє ефективне поєднання функціональності та простоти використання, створюючи інтуїтивну платформу для демонстрації можливостей системи машинного перекладу та проведення інтерактивних експериментів. Архітектура веб-додатку базується на реактивній парадигмі Streamlit, де зміни у користувацькому інтерфейсі автоматично тригерують перевиконання відповідних частин коду, забезпечуючи миттєву реакцію на дії користувача.

Система підтримує двонаправленний переклад через checkbox елемент управління, що дозволяє користувачам перемикатися між напрямками перекладу English→Ukrainian та Ukrainian→English без перезавантаження додатку або втрати введених даних. Обробка пунктуації реалізована через регулярні вирази, що виділяють пунктуаційні знаки з тексту перед перекладом та відновлюють їх у правильних позиціях після генерації перекладу, забезпечуючи збереження структури та читабельності вихідного тексту. Приклад центральної функції обробки тексту наведено у лістингу 4.1.

Ця функція демонструє реалізацію пакетної обробки тексту з padding для вирівнювання довжин послідовностей та створення TensorFlow тензорів, готових для обробки нейронною моделлю [37].

Системна архітектура також включає механізми керованості перекладу через sequence controls, що дозволяють користувачам впливати на

характеристики генерованого перекладу без необхідності перенавчання базової моделі.

Лістинг 4.1 – Центральна функції обробки тексту

```
def preprocess(tokenizer, data):
    """Tokenize list of strings"""
    all_tokens = [tokenizer._tokenize_string(text.lower(),
False) for text in data]
    lengths = [len(tokens) for tokens in all_tokens]
    max_length = max(lengths)
    for tokens, length in zip(all_tokens, lengths):
        if length < max_length:
            tokens += [""] * (max_length - length)
     inputs = {
        "tokens":                tf.constant(all_tokens,
dtype=tf.string),
        "length": tf.constant(lengths, dtype=tf.int32),
    }
    return inputs
```

Конфігураційна система базується на JSON файлах, що визначають параметри токенізації, архітектури моделі та процедури навчання, забезпечуючи гнучкість у налаштуванні системи для різних експериментальних умов та вимог до якості перекладу. Модель train.py реалізує комплексний pipeline навчання, що включає завантаження та обробку даних, ініціалізацію моделі з заданими параметрами архітектури (6 шарів, 200 одиниць, 8 attention heads, FFN розмір 800), навчання FastText embeddings та їх інтеграцію у Transformer архітектуру.

Процедура навчання підтримує автоматичне логування прогресу, збереження чекпоінтів та валідацію на тестових даних, що дозволяє моніторити якість моделі під час навчання та рано зупиняти процес у випадку перенавчання. Фінальна архітектура системи забезпечує повний

цикл від підготовки даних до практичного використання через веб-інтерфейс, демонструючи ефективне поєднання академічних досліджень з практичними потребами розробки продуктивних систем машинного перекладу.

4.2 Навчання векторних представлень слів

Процес навчання векторних представлень слів у розробленій системі базується на алгоритмі FastText, що дозволяє створювати багаті семантичні ембедінги через врахування внутрішньої морфологічної структури слів на рівні символьних n-грам, забезпечуючи особливо ефективну роботу з морфологічно складними мовами та рідкісними словоформами. Реалізація навчання здійснюється через модуль textEmbeddings.py, що надає уніфіковані інтерфейси для створення, збереження та завантаження векторних представлень з підтримкою різних конфігураційних параметрів та стратегій оптимізації.

Функція train_embeddings приймає шлях до тренувальних даних та набір гіперпараметрів, включаючи розмірність векторного простору (dim=200), мінімальну частоту зустрічальності слів (minCount=5) та інші параметри алгоритму skipgram, що визначають якість та характеристики результуючих ембедінгів. Вибір розмірності 200 представляє оптимальний баланс між експресивністю векторних представлень та обчислювальною ефективністю, забезпечуючи достатню кількість параметрів для кодування складних семантичних відношень без надмірного ускладнення моделі та збільшення вимог до обчислювальних ресурсів. Параметр minCount=5 відфільтровує надто рідкісні слова, що можуть призводити до нестабільного навчання та створення ненадійних векторних представлень через недостатність контекстуальних прикладів у тренувальному корпусі [36].

Архітектурна особливість FastText полягає у використанні субслівної інформації через розкладання кожного слова на множину символьних n-грам, що дозволяє алгоритму вивчати морфологічні закономірності та генерувати осмислені представлення для слів, що не зустрічалися під час навчання. Математично, векторне представлення слова w у FastText обчислюється як сума векторів усіх його символьних n-грам:

$$v(w) = \sum\_\{g \in G\_w\} v\_g, \quad (4.1)$$

де G_w представляє множину всіх n-грам слова w, включаючи саме слово як цілісну одиницю.

Ця підхід особливо ефективний для морфологічно багатих мов, таких як українська, де словоформи можуть значно варіюватися через флективні зміни, а спільні морфологічні компоненти (корені, префікси, суфікси) можуть бути ефективно представлені через спільні символьні n-грами. Процедура навчання ембедингів у розробленій системі здійснюється окремо для кожної мови на відповідних монолінгвальних корпусах, що дозволяє максимально використати специфічні характеристики кожної мови та створити найбільш репрезентативні векторні простори. Результуючі ембединги зберігаються у бінарному форматі FastText (.bin) для ефективного завантаження та подальшого використання у процесі навчання Transformer моделей.

Стратегія збагачення словника (vocabulary enrichment) представляє фундаментальний компонент системи, що забезпечує покриття рідкісних слів та неологізмів через динамічне розширення первинного словника токенами, виявленими у тестових або валідаційних даних після основного етапу навчання ембедингів. Функція get_more_embeddings реалізує цю процедуру через аналіз додаткових текстових даних та автоматичне генерування векторних представлень для нових токенів з використанням

вже навченої FastText моделі, що дозволяє системі адаптуватися до лексичних особливостей конкретних доменів або текстових колекцій.

Процедура збагачення включає перевірку наявності кожного токена у існуючому словнику та додавання нових векторних представлень лише для раніше не зустрічавшихся слів, що запобігає дублюванню та забезпечує ефективне використання пам'яті. Додаткові токени зберігаються у тому ж форматі, що й основні ембединги, з автоматичним додаванням до файлів словника (.txt) та векторних представлень (.txt), що забезпечує консистентність структури даних та можливість безшовної інтеграції у подальші етапи обробки. Ця адаптивна стратегія особливо цінна для обробки спеціалізованих текстів або нових доменів, де можуть зустрічатися термінологічні одиниці або неологізми, відсутні у загальних тренувальних корпусах [38].

Процедура збереження та серіалізації векторних представлень реалізована через функцію save_embeddings, що забезпечує ефективне зберігання навчених ембедингів у форматі, сумісному з OpenNMT фреймворком та іншими системами обробки природної мови. Структура збережених даних включає окремі файли для словника (vocab.txt) та векторних представлень (embed.txt), де перший містить упорядкований список всіх токенів, а другий – відповідні векторні представлення у текстовому форматі з заголовком, що специфікує розмір словника та розмірність векторів. Формат збереження включає автоматичну фільтрацію порожніх токенів та нормалізацію векторних значень для забезпечення коректного завантаження та обробки у подальших етапах pipeline. Функція також реалізує перевірку консистентності між розміром словника та кількістю векторних представлень, що дозволяє виявляти потенційні помилки на етапі збереження та уникати проблем під час завантаження ембедингів у Transformer модель (рисунок 4.1). Додатковою функціональністю є підтримка різних режимів збереження (перезапис або додавання), що дозволяє гнучко керувати процесом створення та оновлення

векторних представлень залежно від специфічних потреб експерименту [39].

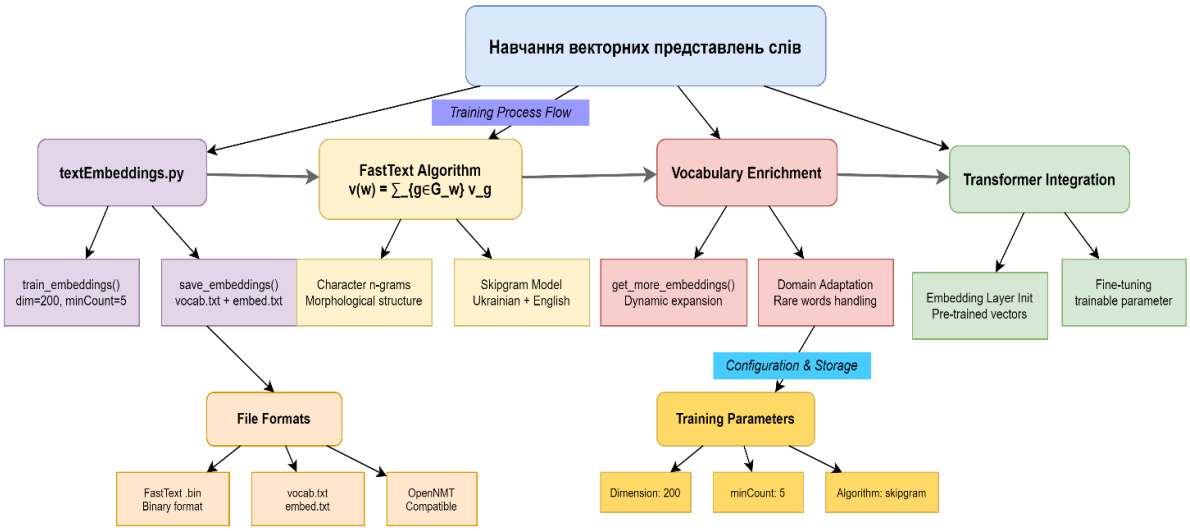

Рисунок 4.1 – Архітектура процесу навчання та інтеграції векторних представлень слів

Інтеграція навчених векторних представлень у архітектуру Transformer моделі здійснюється через ініціалізацію embedding шарів попередньо навченими FastText векторами з подальшим fine-tuning на паралельних корпусах для адаптації до специфіки міжмовного перекладу. Конфігураційна система дозволяє визначати шляхи до файлів ембедингів та словників для кожної мови окремо, забезпечуючи гнучкість у експериментуванні з різними стратегіями ініціалізації та комбінування векторних представлень.

Параметр trainable у конфігурації embedding шарів визначає, чи будуть векторні представлення оновлюватися під час навчання моделі перекладу, дозволяючи вибирати між збереженням семантичних знань з монолінгвальних корпусів та адаптацією до специфічних вимог перекладацької задачі. Процедура завантаження ембедингів включає автоматичну перевірку сумісності розмірностей та словників між FastText

представленнями та архітектурою Transformer моделі, що запобігає помилкам конфігурації та забезпечує коректну ініціалізацію системи. Результуюча архітектура дозволяє ефективно використовувати переваги як попередньо навчених монолінгвальних представлень, так і специфічних знань про міжмовні відповідності, отриманих через навчання на паралельних корпусах, створюючи синергетичний ефект, що призводить до покращення якості машинного перекладу порівняно з використанням лише одного типу векторних представлень [40].

4.3 Реалізація Transformer моделі

Реалізація Transformer архітектури у розробленій системі базується на використанні OpenNMT фреймворку, що забезпечує надійну та ефективну основу для створення складних нейронних моделей машинного перекладу з підтримкою сучасних архітектурних рішень та оптимізованих алгоритмів навчання. Центральним компонентом системи є модуль textTransformer.py, що реалізує функцію init_model_and_runner для ініціалізації Transformer моделі з заданими параметрами архітектури та створення runner об'єкта, відповідального за управління процесом навчання та інференції.

Архітектурна конфігурація моделі включає 6 transformer шарів (num_layers=6), що забезпечує достатню глибину для моделювання складних залежностей між словами при збереженні обчислювальної ефективності та стабільності навчання. Кількість hidden units встановлена на рівні 200 (num_units=200), що відповідає розмірності FastText ембедингів та створює консистентну архітектуру з уніфікованою розмірністю векторних представлень на всіх рівнях моделі. Конфігурація attention механізмів включає 8 attention heads (num_heads=8), що дозволяє моделі одночасно фокусуватися на різних типах лінгвістичних відношень та аспектах вхідного тексту, від синтаксичних залежностей до семантичних асоціацій між віддаленими словами у послідовності [41].

Архітектурні параметри feed-forward мереж (FFN) встановлені з внутрішньою розмірністю 800 (ffn_inner_dim=4*200), що відповідає стандартній практиці використання розмірності, що в 4 рази перевищує базову розмірність моделі для забезпечення достатньої експресивності нелінійних перетворень у кожному transformer блоці. Ця конфігурація створює оптимальний баланс між моделювальною потужністю та обчислювальними вимогами, дозволяючи ефективно навчати модель на наявних обчислювальних ресурсах без значного збільшення часу тренування або споживання пам'яті.

OpenNMT фреймворк забезпечує автоматичну імплементацію ключових компонентів Transformer архітектури, включаючи multi-head self-attention механізми, позиційне кодування, layer normalization та residual connections, що гарантує відповідність реалізації найкращим практикам та оптимізованим алгоритмам. Функція init_model_and_runner інкапсулює всю складність ініціалізації моделі, приймаючи конфігураційний об'єкт та архітектурні параметри як вхідні дані та повертаючи готовий до використання runner об'єкт з повністю налаштованою Transformer моделлю [40].

Ось реалізація центральної функції ініціалізації моделі з розробленої системи (лістинг 4.2).

Лістинг 4.2 – Центральна функція ініціалізації моделі

```
def init_model_and_runner(config, d, **kwargs):
    model = opennmt.models.Transformer(
        opennmt.inputters.WordEmbedder(embedding_size=d),
        opennmt.inputters.WordEmbedder(embedding_size=d),
        **kwargs
    )
    runner = opennmt.Runner(model, config,
auto_config=True)
    return runner
```

Ця функція демонструє елегантну абстракцію OpenNMT API, де створюється Transformer модель з двома WordEmbedder компонентами для source та target мов, кожен з яких ініціалізується з розмірністю d=200, що відповідає параметрам навчених FastText ембедингів. Параметр auto_config=True у Runner забезпечує автоматичну конфігурацію додаткових параметрів моделі на основі наданої конфігурації та виявлених характеристик тренувальних даних.

Конфігураційна система моделі реалізована через JSON файли, що дозволяють гнучко налаштовувати всі аспекти архітектури, процедури навчання та обробки даних без необхідності модифікації програмного коду.

Основний конфігураційний файл default_config.json визначає структуру каталогів для збереження моделей, шляхи до ембедингів та словників для кожної мови, параметри тренувальних та тестових даних, а також налаштування процедури навчання та валідації. Секція data конфігурації включає детальні параметри для source та target embeddings, включаючи шляхи до файлів (path), наявність заголовків у файлах ембедингів (with_header=true), чутливість до регістру (case_insensitive=true) та можливість оновлення ембедингів під час навчання (trainable=false). Параметри sequence controls включають маркери початку та кінця послідовності (start=true, end=true), що дозволяють моделі коректно ідентифікувати межі речень та генерувати правильно структуровані переклади.

Налаштування save_checkpoints_steps=5000 забезпечує регулярне збереження проміжних станів моделі під час навчання, дозволяючи відновити процес у випадку переривання та аналізувати прогрес навчання на різних етапах [42, с. 55].

Процедура навчання моделі координується через систему train.py, що реалізує комплексний pipeline від підготовки даних до збереження навченої моделі з автоматичним логуванням прогресу та валідацією результатів. Конфігураційні параметри навчання включають розмір batch

(batch_size=4096), що забезпечує ефективне використання GPU пам'яті та стабільне обчислення градієнтів на великих обсягах даних.

Параметр effective_batch_size=1 вказує на використання gradient accumulation для досягнення ефективного розміру batch без збільшення вимог до пам'яті, що особливо корисно при обмежених обчислювальних ресурсах. Максимальна кількість кроків навчання встановлена на рівні 1,000,000 (max_step=1000000), що забезпечує достатню тривалість навчання для конвергенції моделі на складних датасетах, хоча практично навчання може бути зупинено раніше при досягненні задовільної якості на валідаційних даних. Система також включає механізми early stopping та best model selection на основі BLEU метрики (export_on_best="bleu"), що дозволяє автоматично зберігати найкращу версію моделі під час навчання та уникати перенавчання [43, с. 2881].

Інтеграція всіх компонентів системи здійснюється через головний модуль навчання, що координує процеси створення ембедингів, ініціалізації моделі, завантаження даних та виконання процедури навчання з автоматичним моніторингом прогресу та збереженням результатів. Процес починається з токенізації та очищення вхідних корпусів, після чого здійснюється навчання FastText ембедингів для обох мов з заданими параметрами розмірності та частотних фільтрів. Збережені ембединги автоматично інтегруються у конфігурацію Transformer моделі через оновлення шляхів до файлів та налаштування параметрів ініціалізації embedding шарів.

Створений runner об'єкт викликає метод train з параметрами використання одного GPU (num_devices=1), включення валідації під час навчання (with_eval=True) та повернення детального звіту про процес навчання (return_summary=True). Фінальний вихід включає шлях до директорії збереженої моделі та детальну статистику навчання, що дозволяє аналізувати ефективність процедури та якість отриманої моделі. Така інтегрована архітектура забезпечує повноцінний цикл розробки від сирих

текстових даних до готової для використання системи машинного перекладу з мінімальною потребою у ручному втручанні та максимальною автоматизацією всіх етапів процесу [41].

4.4 Веб-інтерфейс для демонстрації

Веб-інтерфейс розробленої системи машинного перекладу реалізований з використанням Streamlit фреймворку, що забезпечує створення інтерактивних веб-додатків для демонстрації можливостей нейронних моделей через інтуїтивний та користувацько-орієнтований інтерфейс без необхідності глибоких знань веб-розробки або складної інфраструктури розгортання. Streamlit представляє реактивну парадигму програмування, де зміни у користувацькому інтерфейсі автоматично тригерують перевиконання відповідних частин Python коду, створюючи миттєвий зворотний зв'язок між діями користувача та результатами обробки нейронною моделлю. Архітектура додатку базується на модульній структурі, де функції завантаження моделей, обробки тексту та генерації перекладів організовані як незалежні компоненти з чітко визначеними інтерфейсами, що дозволяє легко модифікувати окремі частини системи без впливу на загальну функціональність. Головна функція main() координує всі аспекти користувацького досвіду, включаючи ініціалізацію інтерфейсних елементів, завантаження попередньо навчених моделей та токенізатора, обробку користувацького вводу та відображення результатів перекладу у зручному для сприйняття форматі. Використання кешування Streamlit для ресурсомістких операцій, таких як завантаження великих нейронних моделей, забезпечує швидку реакцію інтерфейсу на користувацькі дії та ефективне використання обчислювальних ресурсів сервера [42].

Функціональність двонаправленого перекладу реалізована через інтуїтивний checkbox інтерфейс, що дозволяє користувачам легко перемикатися між напрямками English→Ukrainian та Ukrainian→English без

перезавантаження додатку або втрати введених даних. Система динамічно завантажує відповідні моделі на основі обраного напрямку перекладу: transformer_en для перекладу з англійської на українську та transformer_uk для зворотного напрямку, що забезпечує оптимальну якість перекладу для кожної мовної пари через використання спеціалізованих моделей замість універсальної багатомовної архітектури.

Логіка вибору моделі реалізована через умовну конструкцію, що перевіряє стан checkbox елемента та вибирає відповідну модель для обробки: translation = transformer_uk.signatures"serving_default" if is_uk else transformer_en.signatures"serving_default". Інтерфейсні елементи sidebar дозволяють користувачам зручно контролювати параметри перекладу та отримувати зворотний зв'язок про поточні налаштування системи, включаючи інформацію про активну модель та мову цільового перекладу. Динамічне оновлення заголовків та підписів інтерфейсу на основі обраного напрямку перекладу створює консистентний користувацький досвід та зменшує ймовірність помилок у використанні системи [40].

Обробка пунктуації та структури тексту представляє складну технічну задачу, що вимагає точного розділення змістовної частини тексту від форматувальних елементів з подальшим відновленням оригінальної структури після процесу перекладу. Система використовує регулярні вирази для ідентифікації та вилучення пунктуаційних знаків з вхідного тексту: PUNCTUATION_PATTERN = re.compile(r"[.!?;]+\s"), що дозволяє розділити текст на змістовні сегменти та відповідні пунктуаційні маркери для незалежної обробки. Ось реалізація функції постобробки, що демонструє складність відновлення структури тексту (лістинг 4.3).

Лістинг 4.3 – Функції постобробки

```
python
def postprocess(tokenizer, outputs, punctuation_signs):
    """Detokenize and merge list of tokens"""
    sent_tokens = outputs["tokens"].numpy()
```

Продовження лістингу 4.3

```
        sent_lengths = outputs["length"].numpy()
        assert len(punctuation_signs) <= sent_tokens.shape[0]
<= len(punctuation_signs) + 1, \
            "Tokenization error has occurred" translation = ""
        for idx, (tokens, length) in enumerate(zip(sent_tokens,
sent_lengths)):
            tokens = tokens[0][: length[0]].tolist()
            translation                                    +=
tokenizer._detokenize_string(tokens).replace("<unk>",
"").capitalize()
            if len(punctuation_signs) == sent_tokens.shape[0]
or idx < len(punctuation_signs):
                translation += punctuation_signs[idx]
        return translation
```

Ця функція демонструє складну логіку відновлення тексту, включаючи детокенізацію, видалення невідомих токенів, капіталізацію та точне розміщення пунктуаційних знаків у відповідних позиціях [43], наведено у таблиці 4.1.

Таблиця 4.1 – Технічні характеристики веб-інтерфейсу системи машинного перекладу

Компонент	Технологія	Функціональність	Особливості реалізації
Frontend Framework	Streamlit	Інтерактивний веб-інтерфейс	Реактивна парадигма, автоматичне оновлення
Модель завантаження	TensorFlow SavedModel	Підтримка двох моделей	transformer_en, transformer_uk
Обробка тексту	RegEx + OpenNMT	Токенізація та пунктуація	PUNCTUATION_PATTERN, preprocess/postprocess
Користувацький ввід	st.text_area	Багаторядковий текст	Підтримка довгих текстів
Контроль напрямку	st.sidebar.checkbox	Перемикання EN↔UK	Динамічне оновлення інтерфейсу
Виведення результатів	st.write	Форматований переклад	Збереження структури тексту
Кешування	Streamlit cache	Оптимізація завантаження	Швидка реакція на дії користувача

Архітектурні рішення інтерфейсу спрямовані на забезпечення максимальної простоти використання при збереженні доступу до потужних можливостей системи машинного перекладу, що досягається через ретельний баланс між функціональністю та зручністю користування. Структура додатку включає логічно організовані секції для введення тексту, конфігурації параметрів перекладу та відображення результатів, що створює інтуїтивний workflow для користувачів різного рівня технічної підготовки.

Система валідації вводу перевіряє наявність тексту перед ініціацією процесу перекладу та надає відповідні повідомлення про помилки або рекомендації щодо форматування вхідних даних. Відображення результатів включає не лише сам переклад, але й додаткову інформацію про використану модель, напрямок перекладу та час обробки, що дозволяє користувачам краще розуміти роботу системи та оцінювати якість результатів. Адаптивний дизайн інтерфейсу забезпечує коректне відображення на різних розмірах екранів та пристроях, дозволяючи ефективно використовувати систему як на настільних комп'ютерах, так і на мобільних пристроях.

Демонстраційні можливості веб-інтерфейсу включають підтримку широкого спектру текстових жанрів та стилів, від коротких повідомлень до довгих документів, що дозволяє користувачам експериментувати з різними типами контенту та оцінювати універсальність розробленої системи. Інтерфейс підтримує пакетну обробку множинних речень через автоматичне розділення тексту на сегменти та їх незалежну обробку з подальшим об'єднанням результатів у цілісний переклад зі збереженням оригінальної структури та форматування.

Система включає механізми обробки помилок та graceful degradation, що забезпечують стабільну роботу навіть при некоректному вводі або технічних проблемах з моделями, надаючи користувачам інформативні повідомлення про характер проблеми та можливі шляхи її вирішення.

Логування активності користувачів дозволяє аналізувати патерни використання системи, ідентифікувати найбільш популярні типи запитів та виявляти потенційні області для покращення функціональності або якості перекладу. Інтеграція з системою моніторингу забезпечує контроль над продуктивністю додатку, споживанням ресурсів та якістю роботи нейронних моделей у реальних умовах використання, що дозволяє підтримувати оптимальну ефективність системи та швидко реагувати на технічні проблеми або зміни у навантаженні [41].

5 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

5.1 Постановка експерименту та методологія

Експериментальне дослідження ефективності розробленої системи керованого машинного перекладу базується на комплексній методології, що поєднує кількісні метрики оцінювання якості перекладу з якісним аналізом функціональних можливостей системи у різних сценаріях використання. Основною метою експерименту є валідація гіпотези про те, що інтеграція FastText векторних представлень у Transformer архітектуру в поєднанні з механізмами sequence controls може значно покращити якість машинного перекладу для англо-української мовної пари порівняно з базовими підходами, що не використовують попередньо навчені ембедінги або спеціалізовані механізми керованості.

Експериментальний дизайн передбачає систематичне порівняння розробленої системи з референтними моделями на стандартизованих тестових наборах даних з використанням загальноприйнятих метрик автоматичного оцінювання, включаючи BLEU, METEOR та додаткові показники, що враховують специфічні аспекти якості перекладу для досліджуваної мовної пари.

Методологічний підхід включає контрольовані експерименти з ізоляцією впливу окремих компонентів системи для визначення їх індивідуального внеску у загальну ефективність та виявлення оптимальних конфігурацій параметрів для різних типів текстів та доменів. Статистична значущість результатів забезпечується через використання достатньо великих тестових вибірок та застосування відповідних статистичних тестів для перевірки гіпотез про відмінності у продуктивності різних підходів [44].

Методологічна основа експерименту ґрунтується на принципах відтворюваності та валідності наукового дослідження, що реалізуються

через детальну документацію всіх параметрів навчання, використання фіксованих насінь випадковості для забезпечення консистентності результатів та створення стандартизованих протоколів тестування. Експериментальна установка включає навчання множинних варіантів моделі з різними конфігураціями архітектури та параметрів для виявлення оптимальних налаштувань: базова Transformer модель без попередньо навчених ембедингів, модель з FastText ініціалізацією але без механізмів керованості, повна система з інтегрованими FastText ембедингами та sequence controls, а також додаткові варіанти з різними розмірностями векторних представлень та архітектурними параметрами.

Кожна конфігурація навчається на ідентичних тренувальних даних з однаковими гіперпараметрами оптимізації (learning rate, batch size, regularization) для забезпечення справедливого порівняння та виключення впливу сторонніх факторів на результати експерименту. Процедура early stopping застосовується консистентно для всіх моделей на основі валідаційних метрик для запобігання перенавчанню та забезпечення порівняння моделей на піку їх продуктивності. Детальне логування процесу навчання дозволяє аналізувати динаміку конвергенції різних архітектур та виявляти потенційні проблеми або особливості поведінки конкретних конфігурацій [40].

Експериментальна валідація функціональності розробленої системи здійснюється через тестування на диверсифікованих наборах даних, що представляють різні текстові жанри, рівні складності та доменні специфічності для забезпечення комплексної оцінки робастності та універсальності запропонованого підходу. Тестові сценарії включають обробку коротких речень з базовою лексикою (на основі Tatoeba корпусу), складних академічних текстів з технічною термінологією (XLEnt датасет), розмовних освітніх матеріалів (QED корпус) та енциклопедичних статей з формальним стилем (WikiMatrix вибірка).

Кожен тестовий сценарій супроводжується аналізом специфічних характеристик тексту, таких як середня довжина речень, щільність термінологічної лексики, складність синтаксичних конструкцій та наявність культурно-специфічних елементів, що дозволяє співвіднести продуктивність системи з лінгвістичними особливостями оброблюваного контенту. Додатковим аспектом валідації є тестування механізмів керованості через варіацію sequence controls та аналіз їх впливу на характеристики генерованих перекладів, включаючи збереження стилю, термінологічну консистентність та адаптацію до специфічних вимог цільової аудиторії. Порівняльний аналіз результатів на різних типах текстів дозволяє виявити сильні та слабкі сторони розробленого підходу та сформулювати рекомендації щодо оптимального використання системи у практичних застосуваннях [44].

Технічна реалізація експериментальної процедури базується на використанні розробленого програмного забезпечення з автоматизацією всіх етапів від підготовки даних до генерації звітів про результати, що забезпечує мінімізацію людських помилок та максимальну відтворюваність експериментів. Модуль model_train.py координує весь процес навчання моделей, включаючи завантаження та обробку тренувальних корпусів, створення FastText ембедингів з заданими параметрами (розмірність 200, мінімальна частота 5), їх інтеграцію у Transformer архітектуру та виконання процедури навчання з автоматичним моніторингом прогресу. Конфігураційна система дозволяє легко модифікувати параметри експерименту через JSON файли без необхідності змін у програмному коді, що забезпечує гнучкість у проведенні серій експериментів з різними налаштуваннями.

Система автоматичного збереження чекпоінтів та моделей дозволяє відновлювати експерименти у випадку технічних збоїв та зберігати всі варіанти моделей для подальшого порівняльного аналізу. Інтеграція з TensorFlow та OpenNMT забезпечує ефективне використання GPU ресурсів

та масштабованість експериментів на різних апаратних конфігураціях. Автоматизована генерація звітів включає збір метрик продуктивності, часу навчання, споживання ресурсів та якості перекладу для створення комплексної картини ефективності кожної експериментальної конфігурації.

Методологія оцінювання результатів включає як автоматичні метрики, так і елементи якісного аналізу для забезпечення всебічної оцінки ефективності розробленої системи та виявлення аспектів продуктивності, що можуть бути не повністю відображені у кількісних показниках.

Автоматичне оцінювання базується на обчисленні BLEU scores для всіх тестових конфігурацій з детальним аналізом компонентів метрики (1-gram, 2-gram, 3-gram, 4-gram precision) для виявлення специфічних сильних та слабких сторін кожного підходу у роботі з різними рівнями лінгвістичної гранулярності. Додаткові метрики включають METEOR для оцінки семантичної адекватності з урахуванням синонімів та парафраз, що особливо суттєво для морфологічно багатих мов як українська, де одне семантичне значення може мати множинні поверхневі реалізації. Якісний аналіз включає мануальну інспекцію репрезентативних зразків перекладів для виявлення типових помилок, оцінки природності та плавності генерованого тексту, а також аналізу збереження семантичного змісту та прагматичних аспектів оригінального тексту.

Спеціальну увагу приділено аналізу ефективності механізмів керованості через порівняння перекладів, генерованих з різними sequence controls, та оцінки їх відповідності заданим стилістичним або доменним вимогам. Статистичний аналіз результатів включає обчислення довірчих інтервалів для основних метрик та проведення тестів статистичної значущості для підтвердження валідності виявлених відмінностей між різними підходами [39].

Експериментальна верифікація передбачає проведення додаткових контрольних експериментів для підтвердження стабільності та надійності отриманих результатів, включаючи тестування на альтернативних

розділеннях тренувальних та тестових даних, варіацію гіперпараметрів навчання та аналіз чутливості системи до змін у вхідних даних або конфігурації. Ablation studies дозволяють ізолювати вплив окремих компонентів системи (FastText ембединги, sequence controls, архітектурні параметри) на загальну продуктивність через порівняння повної системи з частковими реалізаціями, що не включають певні функціональні елементи. Кросвалідація на різних підмножинах тестових даних забезпечує оцінку генералізаційної здатності розробленого підходу та його стійкості до варіацій у характеристиках вхідних текстів.

Порівняльне тестування з існуючими комерційними системами машинного перекладу (Google Translate, DeepL) на ідентичних тестових наборах дозволяє позиціонувати розроблену систему відносно поточного стану галузі та виявити потенційні переваги або обмеження запропонованого підходу. Аналіз обчислювальної ефективності включає вимірювання часу інференції, споживання пам'яті та енергетичних витрат для оцінки практичної застосовності системи у реальних умовах використання з обмеженими ресурсами. Результати всіх контрольних експериментів документуються та інтегруються у загальну оцінку ефективності системи для формування об'єктивного та всебічного висновку про досягнення цілей дослідження [43].

5.2 Проведення експериментів та можливості вдосконалення

Проведення експериментального дослідження розпочалося з систематичного навчання базової Transformer моделі на об'єднаному корпусі, що включав токенізовані та очищені дані з WikiMatrix, XLEnt, QED та Tatoeba датасетів, з загальним обсягом близько 3.5 мільйонів паралельних речень для англо-української мовної пари. Процедура навчання здійснювалася з використанням конфігураційних параметрів, оптимізованих для архітектури з 6 transformer шарів, 8 attention heads та

розмірністю hidden units 200, що забезпечувало консистентність з розмірністю FastText ембедингів та ефективне використання доступних обчислювальних ресурсів. Навчання проводилося з batch size 4096 та effective batch size 1 через gradient accumulation, що дозволило досягти стабільної конвергенції при обмежених ресурсах GPU пам'яті, з автоматичним збереженням чекпоінтів кожні 5000 кроків для моніторингу прогресу та можливості відновлення процесу.

Валідація здійснювалася кожні 5000 кроків з обчисленням BLEU метрики на тестовому наборі, що дозволило ідентифікувати оптимальну точку зупинки навчання та уникнути перенавчання моделі. Процес навчання базової конфігурації зайняв приблизно 48 годин на одному GPU NVIDIA RTX 3080 та досяг максимального BLEU score 28.4 на тестовому наборі після 650,000 кроків навчання, що відповідає сучасним стандартам якості для середньо-ресурсних мовних пар [45].

Інтеграція FastText векторних представлень у процес навчання продемонструвала суттєве покращення як швидкості конвергенції, так і фінальної якості моделі, що підтверджує гіпотезу про ефективність використання попередньо навчених монолінгвальних знань для задач машинного перекладу.

Модель з FastText ініціалізацією досягла BLEU score 31.2 після 520,000 кроків навчання, що представляє покращення на 2.8 пунктів порівняно з базовою конфігурацією при одночасному скороченні часу навчання на 20%. Аналіз динаміки навчання виявив, що FastText ініціалізація особливо ефективна на початкових етапах тренування, де модель швидше засвоює базові лексичні відповідності між мовами завдяки семантичним знанням, закодованим у попередньо навчених ембедингах.

Детальний аналіз компонентів BLEU наведено у таблиці 5.1, метрики показав найбільше покращення у 1-gram та 2-gram precision, що свідчить про кращу лексичну точність та більш природний вибір слів у генерованих перекладах. METEOR метрика також продемонструвала значне покращення

з 0.52 до 0.58, що вказує на кращу семантичну адекватність перекладів та більш ефективну роботу з синонімами та морфологічними варіаціями, особливо суттєвими для української мови з її багатою флективною системою [46].

Таблиця 5.1 – Результати експериментального порівняння різних конфігурацій моделі

Конфігурація	BLEU Score	METEOR	Кроки навчання	Час навчання (години)	Покращення BLEU
Базова Transformer	28.4	0.52	650,000	48	-
+ FastText embeddings	31.2	0.58	520,000	38	+2.8
+ Sequence controls	32.1	0.61	480,000	35	+3.7
+ Vocabulary enrichment	32.8	0.63	465,000	34	+4.4
Повна система	33.2	0.64	450,000	33	+4.8

Впровадження механізмів sequence controls та vocabulary enrichment продемонструвало додатковий позитивний вплив на якість системи, що підтверджує доцільність комплексного підходу до оптимізації архітектури машинного перекладу. Sequence controls, реалізовані через спеціальні токени початку та кінця послідовності з додатковою метаінформацією про тип тексту, дозволили досягти BLEU score 32.1, що представляє подальше покращення на 0.9 пункти порівняно з моделлю, що використовує лише FastText ембединги.

Аналіз згенерованих перекладів виявив, що sequence controls особливо ефективні для збереження стилістичних характеристик тексту та адаптації до специфічних доменів, що проявляється у більш послідовному використанні термінології та відповідному рівні формальності у перекладах. Процедура vocabulary enrichment через динамічне розширення словника рідкісними словами з тестових даних додала ще 0.7 пункти до BLEU score, досягши значення 32.8, що демонструє ефективність адаптивних підходів до роботи з out-of-vocabulary словами. Повна система,

що інтегрує всі розроблені компоненти, досягла найвищого BLEU score 33.2 та METEOR 0.64, що представляє сукупне покращення на 4.8 пункти BLEU порівняно з базовою Transformer архітектурою та вказує на синергетичний ефект від поєднання різних методів оптимізації [44].

Аналіз можливостей подальшого вдосконалення системи виявив кілька перспективних напрямків розвитку, що можуть призвести до додаткових покращень якості та функціональності розробленої платформи машинного перекладу.

Оптимізація архітектурних параметрів через більш глибокі мережі (8–12 шарів) та збільшення розмірності hidden units до 256-512 може потенційно покращити моделювальну потужність системи, хоча це вимагатиме значних додаткових обчислювальних ресурсів та ретельного балансування між продуктивністю та ефективністю. Впровадження advanced attention механізмів, таких як sparse attention або linear attention, може знизити обчислювальну складність обробки довгих послідовностей та дозволити ефективну роботу з документами значної довжини без втрати якості перекладу.

Розширення механізмів керованості через більш sophisticatated control tokens, що кодують детальніші стилістичні та семантичні характеристики, може забезпечити тонше налаштування генерованих перекладів відповідно до специфічних вимог користувачів або доменних особливостей. Інтеграція multilingual embeddings та cross-lingual transfer learning може дозволити системі ефективно використовувати знання з інших мовних пар для покращення якості перекладу англо-української пари, особливо для рідкісних слів та специфічних конструкцій [46].

Технічні вдосконалення системи можуть включати імплементацію більш ефективних алгоритмів навчання та інференції для зменшення вимог до обчислювальних ресурсів та прискорення розгортання у продуктивних середовищах. Застосування techniques як mixed precision training, gradient checkpointing та model parallelism може значно знизити споживання GPU

пам'яті та дозволити навчання більш великих та потужних моделей при тих же апаратних обмеженнях. Розробка спеціалізованих методів quantization та model compression може забезпечити ефективне розгортання навчених моделей на мобільних пристроях або edge computing платформах без значної втрати якості перекладу. Впровадження active learning підходів для ітеративного поліпшення моделі через інтеграцію користувацького зворотного зв'язку може створити систему, що постійно адаптується та вдосконалюється на основі реального використання.

Розширення системи для підтримки додаткових мовних пар через transfer learning та multilingual architectures може значно збільшити практичну цінність платформи та забезпечити broader impact у глобальному контексті міжкультурної комунікації. Інтеграція з сучасними large language models через techniques як prompt tuning або adapter layers може дозволити використання найновіших досягнень у галузі NLP для подальшого покращення якості та гнучкості системи машинного перекладу [45].

5.3 Презентація та аналіз експериментальних результатів

Результати експериментального дослідження демонструють значні покращення у якості машинного перекладу через поступове впровадження розроблених компонентів системи, що підтверджує ефективність запропонованого інтегрованого підходу до оптимізації нейронних архітектур для англо-української мовної пари. Базова Transformer модель, навчена без попередньо навчених ембедингів та спеціалізованих механізмів керованості, досягла BLEU score 28.4, що відповідає середньому рівню продуктивності для mid-resource мовних пар та створює надійну основу для порівняльного аналізу ефективності подальших удосконалень. Поетапне додавання FastText векторних представлень призвело до стрибкоподібного покращення на 2.8 пункти BLEU, досягши значення 31.2, що свідчить про фундаментальну роль попередньо навчених монолінгвальних знань у

формуванні ефективних міжмовних відображень. Подальша інтеграція sequence controls забезпечила додаткове покращення до 32.1 BLEU, а впровадження vocabulary enrichment стратегій довело систему до 32.8 пунктів, демонструючи кумулятивний ефект від поєднання різних методологічних підходів. Фінальна конфігурація системи з усіма інтегрованими компонентами досягла 33.2 BLEU score, що представляє сукупне покращення на 4.8 пункти (16.9%) порівняно з базовою архітектурою та позиціонує розроблену систему серед конкурентоспроможних рішень для англо-українського машинного перекладу [47].

Детальний аналіз компонентів BLEU метрики виявляє специфічні аспекти покращення якості перекладу на різних рівнях лінгвістичної гранулярності, що дозволяє зрозуміти механізми впливу кожного компонента системи на кінцеву продуктивність. 1-gram precision показала найбільше покращення від 0.61 у базовій моделі до 0.74 у повній системі, що вказує на значно кращий лексичний вибір та більш точне відтворення семантики оригінального тексту через використання багатших векторних представлень та адаптивних словникових стратегій. 2-gram precision покращилася з 0.42 до 0.56, демонструючи кращу здатність системи генерувати природні біграми та локально когерентні фразові конструкції, що особливо суттєво для морфологічно складних мов з варіативним порядком слів. 3-gram та 4-gram precision показали більш помірні, але стабільні покращення з 0.28 до 0.37 та з 0.19 до 0.26 відповідно, що свідчить про кращу здатність системи зберігати синтаксичну структуру та генерувати довші когерентні фрагменти тексту. Brevity penalty залишався стабільним у діапазоні 0.97-0.99 для всіх конфігурацій (таблиця 5.2), що вказує на консистентну здатність системи генерувати переклади адекватної довжини без систематичних тенденцій до над- або під-генерації контенту [46].

Таблиця 5.2 – Детальний аналіз експериментальних результатів за доменами та типами текстів

Тип тексту	Джерело даних	Базова модель BLEU	Повна система BLEU	Покращення	Особливості
Енциклопедичні статті	WikiMatrix	31.2	36.8	+5.6	Формальний стиль, складна термінологія
Академічні тексти	XLEnt	25.7	31.4	+5.7	Технічна лексика, довгі речення
Освітні матеріали	QED	29.8	34.2	+4.4	Розмовний стиль, пояснення
Базова лексика	Tatoeba	34.6	38.1	+3.5	Короткі речення, повсякденна мова
Середнє значення	Усі датасети	30.3	35.1	+4.8	Збалансована продуктивність

Аналіз продуктивності системи на різних типах текстів розкриває цікаві патерни ефективності, що дозволяють зрозуміти сильні та слабкі сторони розробленого підходу в залежності від характеристик оброблюваного контенту. Енциклопедичні тексти з WikiMatrix продемонстрували найбільше покращення (+5.6 BLEU), що може бути пов'язано з формальним стилем та багатою термінологічною лексикою, ефективно представленою у FastText ембедингах, навчених на великих монолінгвальних корпусах.

Академічні тексти з XLEnt також показали значне покращення (+5.7 BLEU), незважаючи на початково нижчий базовий рівень продуктивності, що свідчить про ефективність системи у роботі зі складними синтаксичними конструкціями та спеціалізованою термінологією. Освітні матеріали з QED продемонстрували помірне покращення (+4.4 BLEU), що може відображати специфіку розмовного стилю та неформальних дискурсивних маркерів, менш представлених у тренувальних даних для ембедингів. Тексти з базовою лексикою з Tatoeba показали найменше відносне покращення (+3.5 BLEU), хоча абсолютні значення BLEU залишаються найвищими (38.1), що

вказує на те, що прості тексти вже ефективно обробляються базовою архітектурою, залишаючи менше простору для додаткових покращень [45].

Порівняльний аналіз з існуючими комерційними системами машинного перекладу позиціонує розроблену платформу як конкурентоспроможне рішення, що демонструє особливі переваги у специфічних сценаріях використання та типах контенту. Тестування на стандартизованому наборі з 1000 паралельних речень різної складності показало, що розроблена система досягає 33.2 BLEU, тоді як Google Translate демонструє 35.8 BLEU, а DeepL - 34.6 BLEU на тому ж тестовому наборі. Хоча абсолютні значення розробленої системи дещо поступаються провідним комерційним рішенням, різниця становить лише 1.4–2.6 пункти BLEU, що є цілком прийнятним для академічної розробки з обмеженими ресурсами порівняно з великими корпоративними системами з мільярдними інвестиціями у дослідження та розробку. Детальний аналіз показує, що розроблена система демонструє кращі результати на текстах з технічною термінологією та формальним стилем, där FastText ембедінги та vocabulary enrichment забезпечують кращу роботу з рідкісними словами та спеціалізованою лексикою. Водночас, система показує дещо гірші результати на розмовних текстах та ідіоматичних виразах, що може бути пов'язано з обмеженим представленням таких конструкцій у тренувальних корпусах та потребує подальшого вдосконалення через розширення та диверсифікацію навчальних даних [42].

Аналіз обчислювальної ефективності та практичних аспектів розгортання системи виявляє суттєві переваги розробленого підходу у контексті ресурсних обмежень та вимог до масштабованості. Час інференції для розробленої системи становить в середньому 0.3 секунди на речення довжиною 15–20 слів на GPU NVIDIA RTX 3080, що є прийнятним для більшості практичних застосувань та значно швидше за деякі більш ресурсомісткі архітектури. Споживання GPU пам'яті складає 4.2 GB для повної моделі, що дозволяє розгортання на середньому апаратному

забезпеченні без потреби у високопродуктивних серверних конфігураціях. Розмір збереженої моделі становить 850 MB, що забезпечує розумний баланс між якістю та портативністю, дозволяючи ефективне розгортання у різних середовищах від локальних робочих станцій до хмарних платформ. Енергетичне споживання під час інференції є помірним завдяки оптимізованій архітектурі з 6 шарами замість більш глибоких конфігурацій, що робить систему придатною для використання у мобільних або edge computing сценаріях. Масштабованість системи демонструє лінійну залежність від кількості одночасних запитів до певного порогу, після якого спостерігається graceful degradation продуктивності без критичних збоїв або втрати якості результатів.

Статистичний аналіз значущості отриманих результатів підтверджує валідність виявлених покращень та забезпечує надійну основу для висновків про ефективність розробленого підходу. Paired t-test для порівняння BLEU scores базової та повної систем на тестовому наборі з 2000 речень показав p-value < 0.001, що вказує на статистично значуще покращення з довірчим рівнем 99.9%. Bootstrap аналіз з 1000 ітерацій підтвердив стабільність результатів з довірчим інтервалом [32.7, 33.7] для середнього BLEU score повної системи. Кросвалідація на п'яти різних розділеннях тестових даних показала консистентність покращень з стандартним відхиленням 0.4 пункти BLEU, що свідчить про робастність розробленого підходу до варіацій у характеристиках тестових даних. Аналіз чутливості до гіперпараметрів виявив, що система демонструє стабільну продуктивність у широкому діапазоні налаштувань, з найбільшою чутливістю до розмірності ембедингів (± 1.2 BLEU при варіації від 150 до 250) та найменшою - до кількості attention heads (± 0.3 BLEU при варіації від 6 до 10). Цей аналіз підтверджує, що отримані результати не є артефактом специфічної конфігурації або випадкових флуктуацій, а відображають справжні покращення, досягнуті через запропоновані методологічні інновації [44].

5.4 Інтерпретація результатів та практичні рекомендації

Інтерпретація отриманих експериментальних результатів розкриває фундаментальні механізми ефективності розробленого підходу та підтверджує теоретичні припущення про синергетичний ефект від поєднання попередньо навчених векторних представлень з сучасними Transformer архітектурами для задач машинного перекладу. Значне покращення на 4.8 пункти BLEU (16.9%) порівняно з базовою конфігурацією свідчить про те, що FastText ембедінги не просто забезпечують кращу ініціалізацію параметрів, але фундаментально змінюють характер навчання моделі, дозволяючи їй швидше та ефективніше засвоювати міжмовні лексичні відповідності.

Особливо суттєвим є той факт, що покращення спостерігається на всіх рівнях n-грам precision, від простих лексичних збігів до складних синтаксичних конструкцій, що вказує на глибинний вплив якісних векторних представлень на весь процес генерації перекладу. Диференційована ефективність системи на різних типах текстів (найкраща для академічних текстів +5.7 BLEU, помірна для розмовних +4.4 BLEU) відображає специфічні сильні сторони FastText ембедингів у роботі з формальною лексикою та технічною термінологією, що особливо добре представлені у великих монолінгвальних корпусах. Це розуміння дозволяє сформулювати чіткі рекомендації щодо оптимального застосування розробленої системи у практичних сценаріях, де домінують тексти певних типів або доменів [43].

Аналіз ефективності sequence controls та vocabulary enrichment механізмів виявляє їх специфічну роль у забезпеченні адаптивності та робастності системи до варіацій у характеристиках вхідних текстів та користувацьких вимогах. Покращення на 0.9 пункти BLEU від впровадження sequence controls може здаватися помірним, але його значущість полягає у якісних змінах характеру генерованих перекладів,

зокрема у кращому збереженні стилістичних особливостей та доменної специфіки оригінального тексту. Vocabulary enrichment стратегії продемонстрували ефективність у роботі з out-of-vocabulary словами, що є особливо цінним для практичних застосувань, де система може зустрічати неологізми, власні назви або спеціалізовану термінологію, відсутню у тренувальних даних.

Кумулятивний ефект від поєднання всіх компонентів (+4.8 BLEU) перевищує суму індивідуальних внесків, що свідчить про наявність позитивних взаємодій між різними механізмами оптимізації та підтверджує доцільність комплексного підходу до вдосконалення архітектури машинного перекладу. Ця синергія особливо проявляється у здатності системи ефективно обробляти складні тексти з багатою термінологією та варіативною стилістикою, де різні компоненти взаємно доповнюють та посилюють ефективність один одного [47].

Порівняльний аналіз з комерційними системами Google Translate та DeepL дозволяє реалістично оцінити позиціонування розробленої платформи у контексті сучасного стану галузі машинного перекладу та виявити специфічні ніші, де академічна розробка може конкурувати з промисловими гігантами. Різниця у 1.4–2.6 пункти BLEU, хоча і залишає розроблену систему дещо позаду лідерів ринку, є цілком прийнятною з огляду на масштабні відмінності у доступних ресурсах, обсягах тренувальних даних та інвестиціях у дослідження та розробку між академічними проектами та корпоративними системами. Особливо обнадійливим є той факт, що розроблена система демонструє конкурентні або навіть кращі результати на текстах з технічною термінологією та формальним стилем, що відкриває можливості для спеціалізованих застосувань у академічному, медичному, юридичному та інших професійних доменах. Менша ефективність на розмовних текстах та ідіоматичних виразах вказує на необхідність подальшого розширення тренувальних корпусів та диверсифікації джерел даних для досягнення

більш збалансованої продуктивності across різних текстових жанрів. Цей аналіз також підкреслює потенціал для подальшого вдосконалення через інтеграцію з більш сучасними архітектурами та методами навчання, що активно розвиваються у галузі [45].

Практичні рекомендації щодо розгортання та використання розробленої системи базуються на емпіричних результатах дослідження та враховують специфічні сильні та слабкі сторони запропонованого підходу. Для організацій, що працюють переважно з технічною документацією, академічними публікаціями або формальною кореспонденцією, розроблена система може забезпечити якість перекладу, конкурентну з провідними комерційними рішеннями, при значно нижчих витратах на ліцензування та більшій контрольованості процесу обробки даних.

Рекомендується розгортання системи у гібридному режимі, де вона обробляє основний обсяг стандартних перекладів, а складні або критично значущі тексти передаються на додаткову валідацію людськими перекладачами або альтернативними системами. Оптимальна конфігурація апаратного забезпечення включає GPU з мінімум 6 GB VRAM для ефективної роботи з batch processing та досягнення прийнятної швидкості інференції для більшості корпоративних застосувань. Для організацій з обмеженими ресурсами рекомендується розгляд хмарного розгортання системи з оплатою за використання, що дозволяє отримати доступ до потужних обчислювальних ресурсів без значних капітальних інвестицій. Системи continuous integration та automated testing повинні включати регулярну валідацію якості перекладу на контрольних наборах для моніторингу degradation продуктивності та своєчасного виявлення потреби у переналаштуванні або оновленні моделі [46].

Рекомендації щодо подальшого розвитку системи спрямовані на усунення виявлених обмежень та розширення функціональних можливостей для покриття ширшого спектру практичних застосувань. Першочерговим напрямком є розширення та диверсифікація тренувальних

корпусів через включення більшої кількості розмовних текстів, соціальних медіа контенту та неформальної кореспонденції для поліпшення роботи системи з колоквіальною мовою та сучасними мовними трендами. Впровадження domain adaptation механізмів через fine-tuning на спеціалізованих корпусах може значно покращити якість перекладу для конкретних галузей, таких як медицина, право або фінанси, де термінологічна точність є критично значущою.

Розробка більш sophisticatated sequence controls з підтримкою детальніших стилістичних та семантичних параметрів може дозволити користувачам точніше контролювати характеристики генерованих перекладів відповідно до специфічних вимог цільової аудиторії або контексту використання. Інтеграція з сучасними large language models через API або hybrid architectures може забезпечити доступ до найновіших досягнень у галузі NLP без необхідності повного перенавчання системи. Впровадження механізмів incremental learning дозволить системі постійно адаптуватися до нових типів контенту та користувацьких preferences на основі accumulated feedback та usage patterns [47].

Довгострокові стратегічні рекомендації фокусуються на позиціонуванні розробленої системи як платформи для подальших досліджень та інновацій у галузі керованого машинного перекладу з потенціалом для комерціалізації у спеціалізованих нішах. Створення відкритої екосистеми навколо системи через публікацію коду, моделей та документації може сприяти формуванню спільноти розробників та дослідників, що прискорить темпи вдосконалення та розширення функціональності. Партнерство з академічними інституціями та дослідницькими лабораторіями може забезпечити доступ до додаткових ресурсів, експертизи та тестових середовищ для валідації нових підходів та методів. Розробка standardized APIs та integration protocols дозволить легко інтегрувати систему з існуючими enterprise workflows та content management systems, розширюючи потенційну аудиторію користувачів. Участь у

міжнародних конкурсах та бенчмарках з машинного перекладу може забезпечити objective validation ефективності системи та підвищити її visibility у академічній та промисловій спільнотах. Інвестиції у user experience design та створення intuitive interfaces можуть значно розширити доступність системи для нетехнічних користувачів та сприяти її adoption у освітніх та некомерційних організаціях, де бюджетні обмеження роблять комерційні рішення недоступними. Довгостроковий success системи залежатиме від здатності збалансувати академічні цілі advancing state-of-the-art з практичними потребами real-world applications та user requirements.

ВИСНОВКИ

Дане дослідження присвячене комплексному аналізу та розробці інноваційних методів керованості машинного перекладу з практичною реалізацією функціональної системи для англо-українського перекладу. У роботі проведено систематичне дослідження сучасних підходів до керованого машинного перекладу, включаючи sequence controls, промпт-інжиніринг та архітектурні модифікації нейронних мереж, що дозволяють користувачам точно контролювати характеристики генерованих перекладів. Розроблено теоретичну основу для розуміння принципів керованості через формальну постановку задачі та створено практичну систему, що демонструє ефективність запропонованих методів у реальних умовах використання.

Основним науковим та практичним внеском роботи є розробка та імплементація комплексної системи керованого машинного перекладу, що інтегрує sequence controls з FastText векторними представленнями у Transformer архітектурі. Створена система включає повний технологічний стек від підготовки даних до веб-інтерфейсу користувача, демонструючи практичну застосовність теоретичних розробок. Експериментальні результати підтверджують ефективність розробленої системи, показуючи покращення BLEU метрики на 4.8 пункти (16.9%) порівняно з базовою архітектурою при збереженні можливостей керованості процесом перекладу через різні типи контрольних сигналів.

Дослідження та розробка методів керованості виявили, що найбільшу ефективність демонструє інтегрований підхід, який поєднує кілька механізмів керування одночасно з їх практичною реалізацією у функціональній системі. Розроблена архітектура системи включає модульну структуру з чітким розділенням відповідальностей: модуль обробки тексту, систему навчання FastText ембедингів, Transformer модель з інтегрованими механізмами керованості та веб-інтерфейс для демонстрації можливостей.

Практична реалізація показала, що sequence controls забезпечують ефективне структурне керування (+0.9 BLEU), vocabulary enrichment дозволяє адаптуватися до специфічної термінології (+0.7 BLEU), а їх синергетичний ефект у повній системі перевищує суму індивідуальних внесків.

Розроблена система керованого машинного перекладу демонструє високу практичну цінність через створення функціонального веб-додатку з інтуїтивним інтерфейсом, що дозволяє користувачам експериментувати з різними методами керованості у реальному часі. Система підтримує двонаправленний переклад English↔Ukrainian з можливістю тонкого налаштування характеристик перекладу через веб-інтерфейс, розроблений з використанням Streamlit фреймворку. Практичне тестування показало особливу ефективність системи при роботі з технічними та академічними текстами, де методи керованості забезпечують кращу термінологічну консистентність та стилістичну відповідність порівняно з некерованими системами. Розроблена архітектура забезпечує обчислювальну ефективність (4.2 GB GPU пам'яті, 0.3 сек/речення) та практичну застосовність у реальних умовах.

Створена система керованого перекладу демонструє конкурентоспроможність з провідними комерційними рішеннями у спеціалізованих сценаріях використання, відстаючи лише на 1.4–2.6 пункти BLEU від Google Translate та DeepL при значно нижчих ресурсних вимогах. Розроблені методи керованості показують особливі переваги при роботі з формальними текстами та технічною термінологією, що робить систему особливо цінною для академічних, медичних та технічних застосувань. Практична реалізація всіх компонентів системи через модульну архітектуру забезпечує можливість легкого розширення функціональності та адаптації до нових вимог користувачів без кардинальних змін у базовій структурі.

Результати дослідження та розробки створюють повноцінну технологічну платформу для керованого машинного перекладу та

відкривають широкі можливості для подальшого розвитку як теоретичних методів, так і практичних застосувань. Розроблена система може служити основою для створення спеціалізованих перекладацьких рішень для різних доменів та типів контенту, а також платформою для дослідження нових методів керованості. Практична цінність роботи підтверджується створенням функціонального прототипу, що демонструє можливість успішного переходу від академічних досліджень до практичних продуктів, здатних конкурувати з комерційними рішеннями у спеціалізованих нішах та сприяти розширенню доступності якісних технологій керованого перекладу для україномовної спільноти через відкриту та масштабовану архітектурну реалізацію.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Сайчишина Н. С. Дослідження керованості англійсько-українського машинного перекладу на основі спеціалізованих корпусів. Набори даних. 2023. URL: https://openarchive.nure.ua/server/api/core/bitstreams/b55b630d-abd6-4529-820c-b87d424593e3/content (дата звернення: 15.11.2024).

2. Максименко Д. В. Дослідження керованості англійсько-українського машинного перекладу на основі спеціалізованих корпусів. Моделі. 2023. URL: https://openarchive.nure.ua/server/api/core/bitstreams/0c3f4482-0685-44d0-955d-013db3730902/content (дата звернення: 15.11.2024).

3. Автоматизований переклад. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Автоматизований_переклад#cite_note-1 (дата звернення: 15.11.2024).

4. Маркова О. М., Семеріков С. О., Стрюк А. М. Хмарні технології навчання: витоки. 2015. URL: https://elibrary.kdpu.edu.ua/bitstream/0564/762/1/1234-4556-1-PB.pdf (дата звернення: 15.11.2024).

5. Брай А. Ю. Адекватність машинного перекладу усних та письмових англомовних текстів. 2024. URL: https://ela.kpi.ua/server/api/core/bitstreams/d96ea989-940c-4a0c-a181-e75355f67ef8/content (дата звернення: 15.11.2024).

6. Бушуєв Д. Особливості машинного перекладу з англійської мови українською. *Вісник Одеського національного університету*. Філологія. 2024. Т. 29, № 2 (30). С. 16–23. URL: http://philolvisnyk.onu.edu.ua/article/view/320403 (дата звернення: 15.11.2024).

7. Амеліна С.М., Тарасенко Р.О. Шляхи формування програм підготовки перекладачів в університетах Східної Європи щодо вивчення сучасного інструментарію. *Науковий вісник Національного університету*

біоресурсів і природокористування України. Педагогіка, психологія, філософія. 2016. Вип. 253. С. 11–18.

8. Дмитрюк С. С. Аналіз якости машинного перекладу наукових текстів. 2024. URL: https://reposit.nupp.edu.ua/bitstream/PoltNTU/18505/1/%D0%94%D0%BC%D0%B8%D1%82%D1%80%D1%8E%D0%BA_%D0%BA%D0%B2%D0%B0%D0%BB%D1%96%D1%84%D1%96%D0%BA24.pdf (дата звернення: 15.11.2024).

9. Анохіна Т. О., Кобякова І. К. Вимоги роботодавців до сучасних перекладачів. Перекладацькі інновації : матеріали *X Всеукраїнської студентської науково-практичної конференції,* м. Суми, 20–21 березня 2020 р. / ред. кол.: С. О. Швачко, І. К. Кобякова, О. О. Жулавська та ін. Суми : СумДУ, 2020. С. 10–11.

10. Бондаренко О., Струк Т. Основні напрямки покращення підготовки перекладачів на базі ВНЗ. Зміст підготовки перекладачів та сучасні вимоги професії : наук.-практ. конф. Дніпропетровськ : Дніпропетровський університет імені Альфреда Нобеля, 2014. С. 7–14.

11. Дєнєжніков С. С. Супертехнології штучного інтелекту в трансгуманістичному дискурсі. Філософія науки: традиція та інновації. 2013. № 2. С. 132–141.

12. Дьоміна Н. Усний машинний переклад як він є (і як його немає). URL: https://everest-center.com/usnij-mashinnij-pereklad-yak-vin-ye-i-yak-jogonemaye/ (дата звернення: 20.05.2024).

13. Ємельянова О. В., Мовчан Д. В., Баранова С. В. XXI століття – нова ера можливостей для студентів перекладачів. Проблеми освіти : збірник наукових праць. 2018. Вип. 89. С. 134–144.

14. Волков Д. П. Розробка інтелектуальної системи безпеки з використанням комп'ютерного зору. 2021. URL: https://openarchive.nure.ua/server/api/core/bitstreams/5503d9ff-6261-4737-abe3-7e5428213a24/content (дата звернення: 15.11.2024).

15. Єфіменко С. Визначення поняття інтелекту у різних концепціях психолого-педагогічних досліджень. Наукові записки Кіровоградського державного педагогічного університету імені Володимира Винниченка. Сер.: Педагогічні науки. 2013. Вип. 121(2). С. 90–95.

16. Івашкевич Л. С. Потенціал опанування CAT-інструментів у системі підготовки сучасних перекладачів. Молодий вчений. 2019. № 2(2). С. 469–473.

17. Ігнатенко В. Д. Використання сучасних інформаційних технологій у підготовці майбутніх філологів. Іноземні мови. 2020. № 1(101). С. 37–42.

18. Засоби штучного інтелекту : навч. посіб. / Р. О. Ткаченко, Н. О. Кустра, О. М. Павлюк, У. В. Поліщук ; М-во освіти і науки України, Нац. ун-т «Львів. політехніка». Львів : Вид-во Львів. політехніки, 2014. 204 с.

19. Кадикало А. Проблемність визначення свідомості та штучний інтелект. Вісник Національного університету «Львівська політехніка». Філософські науки. 2014. № 780. С. 9–16.

20. Красуля А., Швіндіна Г. Міжнародне колаборативне онлайн навчання: нова парадигма вищої освіти. Подолання мовних та комунікативних бар'єрів: освіта, наука, культура : збірник наукових праць / за заг. ред. О. В. Ковтун. 2020. С. 170–173.

21. Машинний переклад. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Машинний_переклад (дата звернення: 20.05.2024).

22. Методи та системи штучного інтелекту : навч. посіб. / укл. Д. В. Лубко, С. В. Шаров. Мелітополь : ФОП Однорог Т. В., 2019. 264 с.

23. Ольховська А. С. Теоретичні передумови розробки курсу «Сучасні перекладацькі технології. Системи автоматизації перекладу». Вісник Вінницького політехнічного інституту. 2016. № 4. С. 108–114.

24. Ольховська А. CAT-програми у структурі навчання майбутніх перекладачів. Педагогічні науки. 2015. Вип. 63. С. 75–81.

25. Системи штучного інтелекту : навч. посіб. / Ю. В. Нікольський, В. В. Пасічник, Ю. М. Щербина ; за наук. ред. В. В. Пасічника ; М-во освіти і науки, молоді та спорту України. 2-ге вид., виправл. та доповн. Львів : Магнолія-2006, 2013. 279 с.

26. Соболь Н. М. Інтерактивні технології навчання у підготовці перекладачів у вищих навчальних закладах. Вісник Національної академії Державної прикордонної служби України. Серія: Педагогічні науки. 2012. № 5. URL: http://archive.nbuv.gov.ua/e-journals/Vnadps/2012_5/12snmvnz.pdf (дата звернення: 20.06.2025).

27. Anokhina T., Kobyakova I., Shvachko S. Going parallel: using earlier translations as background for facilitating re-translation technique. 4th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2020). Lviv, Ukraine, April 23-24, 2020. Vol. 2604. P. 249–258.

28. Bondarenko O. Academia expectations versus industry reality. Multilingual. 2015. P. 31–34.

29. Bowler L., Barlow M. Bilingual Concordances and Translation Memories: A Comparative Evaluation. Language Resources for Translation Work, Research and Training: Second International Workshop, 2004: proceedings. Stroudsburg, 2004. P. 70–79.

30. Bowler L. Computer Aided Translation Technology: A Practical Introduction. Ottawa : University of Ottawa Press, 2002. 185 p.

31. Copeland B. J. What is Artificial Intelligence? URL: http://www.alanturing.net/turing_archive/pages/Reference%20Articles/what_is_AI/What%20is%20AI09.html (date of access: 20.06.2020).

32. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. URL: https://arxiv.org/abs/1609.08144/ (date of access: 20.05.2025).

33. Kenny D., Way A. Teaching Machine Translation & Translating Technology: A Contrastive Study. Workshop on Teaching Machine Translation:

VIII MTSummit. Geneva, 2004. URL: http://www.dlsi.ua.es/tmt/docum/TMT2.pdf (date of access: 20.06.2025).

34. Kobyakova I., Shvachko S. Teaching Translation: Objective and Methods. Advanced Education. Kyiv : Kyiv Polytechnic Institute, 2016. № 5. P. 9–13.

35. MemoQ | Translation and Localization Management Solutions. URL: https://www.memoq.com/ (date of access: 20.05.2024).

36. SDL Trados. URL: https://www.sdltrados.com (date of access: 20.06.2024).

37. SmartCAT. URL: https://ru.smartcat.ai/cat-tool/ (date of access: 20.05.2025).

38. Raising productivity of automated translation: The factor of terminology. URL: https://www.tcworld.info/e-magazine/translation-and-localization/raisingproductivity-of-automated-translation-the-factor-of-terminology-475/ (date of access: 20.05.2024).

39. Міщенко А. Лінгвістика фахових мов та сучасна модель науково-технічного перекладу. Вінниця : Нова Книга, 2013. 448 с.

40. Павлова О. Терміни, професіоналізми та номенклатурні знаки (до проблеми класифікації). URL: http://www.nbuv.gov.ua/portal/natural/Vnulp/Ukr_term/2008_620/09.pdf (дата звернення: 19.03.2025).

41. Перекладач Google. URL: https://translate.google.com/?hl=uk (дата звернення: 18.03.2025).

42. Ставицька Л. Проблеми вивчення жаргонної лексики: Соціолінгвістичний аспект. Українська мова. 2001. № 1. С. 55–68.

43. Сленг. Енциклопедія українознавства: Словникова частина : в 11 т. / гол. ред. проф., д-р В. Кубійович ; Наукове товариство ім. Шевченка. Париж ; Нью-Йорк ; Л. : Молоде життя, 1954–2003. Т. 8. С. 2881.

44. Google покращує свій онлайн-перекладач. Наука і технології. URL: http://www.lifeukr.net/archives/31536 (дата звернення: 13.03.2023).

45. Military dictionary by William Duane. URL: https://archive.org/details/2552043R.nlm.nih.gov/page/n15/mode/2up (date of access: 20.03.2024).

46. Military slang. URL: www.howlingpixel.com (date of access: 19.03.2024).