Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження застосування моделей графового представлення знань для задач машинного перекладу
(тема)

Виконав:
здобувач другого року навчання
групи ІПЗм-23-2

Олег Халєєв
(Власне ім'я, ПРІЗВИЩЕ)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)
Тип програми освітньо-наукова

Керівник доц. Андрій Бабій
(посада, Власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту
Зав. кафедри _____ Кирило СМЕЛЯКОВ
(підпис) (Власне ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерних наук
Кафедра програмної інженерії
Рівень вищої освіти другий (магістерський)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми освітньо-наукова програма
Освітня програма Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Халєєву Олегу Євгеновичу
(прізвище, ім’я, по батькові)

1. Тема роботи «Дослідження застосування моделей графового представлення знань для задач машинного перекладу»
Затверджена наказом по університету від 15.04.2025р. № 290 Cm
2. Термін подання студентом роботи до екзаменаційної комісії 13.06.2025
3. Вихідні дані до роботи графи знань, машинний переклад, технічна документація, онтологія, терменологія, Transformer, RDF, OWL, NMT-модель, BLEU, TER, METEOR.
4. Перелік питань, що потрібно опрацювати в роботі
вступ, аналіз предметної галузі, огляд й аналіз літературних, наукових джерел, постановка задачі теоретичне дослідження, висновки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
	Отримання завдання	15.04.2025	*виконано*
	Аналіз предметної галузі і постановка задачі	17.04.2025	*виконано*
	Огляд й аналіз літературних, наукових джерел	22.04.2025	*виконано*
	Постановка задачі	28.04.2025	*виконано*
	Теоретичне дослідження	02.05.2025	*виконано*
	Підготовка до апробації результатів дослідження. Публікація матеріалів	21.05.2025	*виконано*
	Програмна реалізація	23.05.2025	*виконано*
	Підготовка пояснювальної записки	25.05.2025	*виконано*
	Підготовка презентації та доповіді	28.05.2025	*виконано*
	Перевірка на плагіат	04.06.2025	*виконано*
	Нормоконтроль	05.06.2025	*виконано*
	Рецензування	10.06.2025	*виконано*
	Попередній захист	12.06.2025	*виконано*
	Занесення диплома в електронний архів	12.06.2025	*виконано*
	Допуск до захисту у зав. кафедри	13.06.2025	*виконано*

Дата видачі завдання 15.04.2025р.

Студент (ка) _____ Олег Халєєв
(підпис)

Керівник роботи _____ доц. Андрій Бабій
(підпис) (посада, Власне ім’я, ПРІЗВИЩЕ)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 63 с., 21 рис., 1 табл., 15 джерел.

ГРАФ ЗНАНЬ, МАШИННИЙ ПЕРЕКЛАД, BLEU, DFS, METEOR, NETWORKX, OWL, PYTHON, PYTORCH, RDFlib, REST API, RDF, SPACY, TER, TENSORFLOW, TRANSFORMERS.

Мета дослідження спрямована на інтеграцію графових моделей знань у нейронні моделі машинного перекладу з метою покращення якості перекладу технічної документації.

Метою роботи є аналіз методів інтеграції графових моделей знань у нейронні моделі машинного перекладу, зокрема архітектуру Transformer, для підвищення точності та однозначності перекладу технічних термінів. Даний підхід передбачає використання графів знань для забезпечення додаткового семантичного контексту, що дозволяє краще розуміти та обробляти складні терміни та синтаксичні конструкції у технічних текстах.

В результаті дослідження планується дослідити гібридну модель машинного перекладу, що інтегрує граф знань у Transformer. Очікується, що така модель забезпечить покращення якості перекладу технічної документації, що відображатиметься у підвищенні показників BLEU та METEOR, а також зниженні показника TER. Крім того, інтеграція графів знань має сприяти більш точному розумінню контексту та семантики термінів, що є критично важливим для технічних текстів

BLEU, DFS, GRAPH-TOOL, METEOR, NLTK, NETWORKX, OWL, PYTORCH, PYTHON, RDFlib, REST API, RDF, TER, TENSORFLOW, TRANSFORMERS.

The research aims to integrate graph-based knowledge models into neural machine translation models to improve the quality of technical documentation translation.

The objective of this work is to analyze methods of integrating graph-based knowledge models into neural machine translation models, particularly the Transformer architecture, to enhance the accuracy and unambiguity of technical term translation. This approach involves using knowledge graphs to provide additional semantic context, which allows for a better understanding and processing of complex terms and syntactic structures in technical texts.

As a result of the research, it is planned to investigate a hybrid machine translation model that integrates a knowledge graph into the Transformer. It is expected that such a model will provide an improvement in the quality of technical documentation translation, which will be reflected in an increase in BLEU and METEOR scores, as well as a decrease in the TER score. Furthermore, the integration of knowledge graphs should contribute to a more accurate understanding of the context and semantics of terms, which is crucial for technical texts.

Умови публікації звіту: заява щодо самостійного виконання кваліфікаційної роботи та можливості її публікації в електронному архіві відкритого доступу ElAr KhNURE.

Завідувачу кафедри
ПІ
(скорочена назва кафедри)
проф. Кирилу СМЕЛЯКОВУ
(вчене звання, сласне ім'я, прізвище)

ЗАЯВА
щодо самостійності виконання кваліфікаційної роботи та можливості її публікації (та/або публікації анотації кваліфікаційної роботи) в електронному архіві відкритого доступу ElAr KhNURE

Я, Халєєв Олег Євгенович
(прізвище, ім'я, по батькові)

здобувач вищої освіти на другому (магістерському) рівні вищої освіти академічної групи ІПЗм-23-2

кафедра програмної інженері,
(повна назва кафедри)

заявляю: «Дослідження застосування моделей графового представлення знань для задач машинного перекладу»,
(назва роботи)

що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в репозиторії "ElArKhNURE". Погоджуюся з авторським договором, відповідно до Положення про репозиторій ХНУРЕ "ElArKhNURE". Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений (а) з вимогами академічної доброчесності, згідно з якими виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

Дата 03.06.2025 Підпис Халєєв

ЗМІСТ

Перелік скорочень 8
Вступ 9
1 Аналіз предметної галузі 11
1.1 Огляд існуючих підходів і їхні обмеження 12
1.2 Тенденції та перспективи, практична цінність і виклики 14
2 Огляд й аналіз літературних, наукових джерел 17
3 Постановка задачі 23
4 Теоретичне дослідження 27
4.1 Моделі графового представлення знань 27
4.2 Підходи до інтеграції графів знань 30
4.3 Методи порівняльного аналізу та оцінка якості перекладу 35
5 Програмна реалізація 40
Висновки 49
Перелік джерел посилання 51
Перелік джерел посилання за науковими напрямами керівника та науковців кафедри програмної інженерії 53
Додаток А Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ 54
Додаток Б Слайди презентації 56
Додаток В Апробація результатів роботи 62
Додаток Г Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008: 2015 63

ПЕРЕЛІК СКОРОЧЕНЬ

BLEU – Bilingual Evaluation Understudy

BFS – Breadth-First Search

DFS – Depth-First Search

METEOR – Metric for Evaluation of Translation with Explicit ORdering

NLTK – Natural Language Toolkit

NMT – Neural Machine Translation

OWL – Web Ontology Language

RDF – Resource Description Framework

RNN – Recurrent Neural Network

SMT – Statistical Machine Translation

TER – Translation Edit Rate

ВСТУП

В наш час, коли інформації стає все більше та вона доступна різними мовами, якісний та ефективний переклад – це досі важливо, зокрема в галузі машинного перекладу. Тому використання графів знань для машинного перекладу стає все актуальнішим. Традиційні методи, які базуються на статистиці чи правилах, часто буксують, коли треба обробити слова з різними значеннями, складні речення чи зрозуміти контекст. А от графи знань – це як структуровані бази даних, де описані різні сутності та зв'язки між ними, вони відкривають нові можливості для покращення якості перекладу, бо дозволяють враховувати семантику, тобто зв'язки між словами та фразами, що допомагає зробити переклад точнішим і більш відповідним контексту.

Особливо це важливо для перекладу термінології у вузькоспеціалізованих сферах, наприклад, у технічній документації. Там кожен термін має бути перекладений максимально точно, бо від цього залежить правильне розуміння інформації. Помилка в перекладі якогось терміна може призвести до проблем на виробництві, використання обладнання та взагалі нічого хорошого з цього не вийде. Тому дослідження, як графи знань можуть допомогти з перекладом термінів у технічній документації, є дуже важливим в наш час.

Власне, це дослідження застосування графів знань для машинного перекладу, має головний фокус на аналізу ефективності графів знань у використовуванні для перекладу термінів у технічній документації.

Щоб досягти цієї мети, треба розібратися з кількома завданнями. По-перше, потрібно проаналізувати предметну область, тобто подивитися, які зараз є підходи до машинного перекладу і як там вже використовують графи знань. По-друге, треба зробити огляд літератури та наукових джерел, які стосуються машинного перекладу, графів знань і їхнього застосування для перекладу термінології. Далі треба чітко сформулювати задачу дослідження і визначити, за якими критеріями ми будемо оцінювати, наскільки добре графи знань підходять для перекладу термінів у технічній документації. І нарешті, провести теоретичне дослідження,

тобто розробити або адаптувати вже існуючі моделі графового представлення знань для перекладу термінів і проаналізувати, наскільки вони ефективні.

Об'єктом дослідження будуть процеси машинного перекладу, зокрема переклад термінології, а предметом – моделі графового представлення знань та їхнє застосування для покращення якості перекладу термінів у технічній документації. Для дослідження використовуватимуться такі методи: аналіз та синтез літературних джерел, порівняльний аналіз існуючих підходів, а також моделювання та експериментальна перевірка.

Таким чином, дане дослідження сприятиме розширенню меж розуміння застосування графів у машинному перекладі технічної документації та створенню нових перспективних рішень у цій галузі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

Аналіз предметної галузі в контексті машинного перекладу передбачає комплексне занурення у специфіку досліджуваного напрямку, його історичний розвиток, сучасні тренди і проблематику. Машинний переклад уже давно став важливою складовою багатьох систем автоматизації робочого процесу, що охоплює глобальні компанії, інтернет-сервіси, наукові проекти, медичний сектор, військову галузь та інші сфери. Багатомовність, різноманітність термінології й постійна потреба у швидкому та точному обміні інформацією робить це завдання одним із найбільш затребуваних у сучасній індустрії обробки природної мови.

Специфіка полягає в тому, що тексти для перекладу часто мають складні мовні конструкції, використовують спеціалізовану або навіть унікальну термінологію та можуть містити елементи різних стилів мовлення. Це вимагає від систем перекладу здатності глибоко розуміти контекст: не тільки на рівні окремих слів, а й з урахуванням стилю, синтаксису, семантики в реченні або навіть у всьому документі. Особливо відчутні ці виклики в медичній сфері, де допущення неточностей може призвести до критичних помилок у клінічних рекомендаціях чи діагностиці. Аналогічно, в юридичному чи технічному середовищі важливо зберегти юридичну силу документів і передавати точне значення без втрати або спотворення.

Потреба у точному машинному перекладі зростає зі збільшенням міжнародної комунікації та обсягів цифрової інформації. При цьому постає питання масштабування: треба обробляти величезні масиви даних, робити це швидко і забезпечувати високу якість. Ураховуючи все це, огляд існуючих підходів дозволяє простежити, як змінювалися моделі та методи, які обмеження їм притаманні та до яких сучасних досягнень вони призвели. Такий огляд також допомагає визначити масштаби проблеми, оцінити попередні рішення та визначити нинішні виклики, що стоять перед дослідниками й інженерами в галузі обробки природної мови.

1.1 Огляд існуючих підходів і їхні обмеження

Статистичні моделі, а саме Statistical Machine Translation(SMT) стали одними з перших успішних методів, здатних автоматично будувати переклад, спираючись на великі обсяги парних даних, або кажучи інакше на корпус текстів двома мовами. Ідея полягає в тому, що ймовірність перекладу конкретного слова або фрази визначається статистично на основі попередніх прикладів у тренувальних даних. Хоча SMT тривалий час демонстрував прийнятну якість, цей підхід погано враховує більш широкий контекст і складну термінологію, оскільки він покладається насамперед на частотні закономірності та ймовірності відповідності слів. Цей підхід виявився особливо вразливим, коли текст містив малопоширені або нові слова, а також коли зустрічалися довгі чи складні конструкції речень із багатьма підрядними зв’язками.

У подальшому розвиток глибинного навчання привів до появи нейронних мереж, що дало змогу створити Нейронні моделі чи Neural Machine Translation(NMT). Такі моделі розглядають переклад як задачу перетворення вхідної послідовності слів у вихідну послідовність. Головна перевага полягає в тому, що контекст моделюється за допомогою рекурентних мереж(RNN) або моделей з механізмом уваги. Тобто нейронні мережі здатні більш гнучко реагувати на нові фрази та на синтаксичні конструкції, а також краще передавати значення в межах цілого речення. Але навіть найбільш сучасні нейронні мережі зіткаються з труднощами при роботі з дійсно складною чи вузькоспеціалізованою термінологією, яка характерна для наукових, медичних або технічних текстів[1].

Окрім цього, ще одним важливим напрямом є семантично орієнтовані методи з використанням графів знань(Knowledge Graph). Ідея полягає в тому, щоб доповнити чисто статистичне чи нейронне бачення тексту формалізованою моделлю понятійних зв’язків(див. рис. 1.1). Наприклад, якщо в документі йдеться про хімічний препарат, у графі знань можна зберігати інформацію про його дію, групу речовин, протипоказання тощо, це дає можливість уточнювати переклад, ураховуючи зв’язки між поняттями.

Незважаючи на те, що впровадження графів знань суттєво підвищує точність перекладу, така інтеграція вимагає детальної розробки онтологій та формальних описів, а також вирішення питань сумісності таких графів із різними мовними моделями.

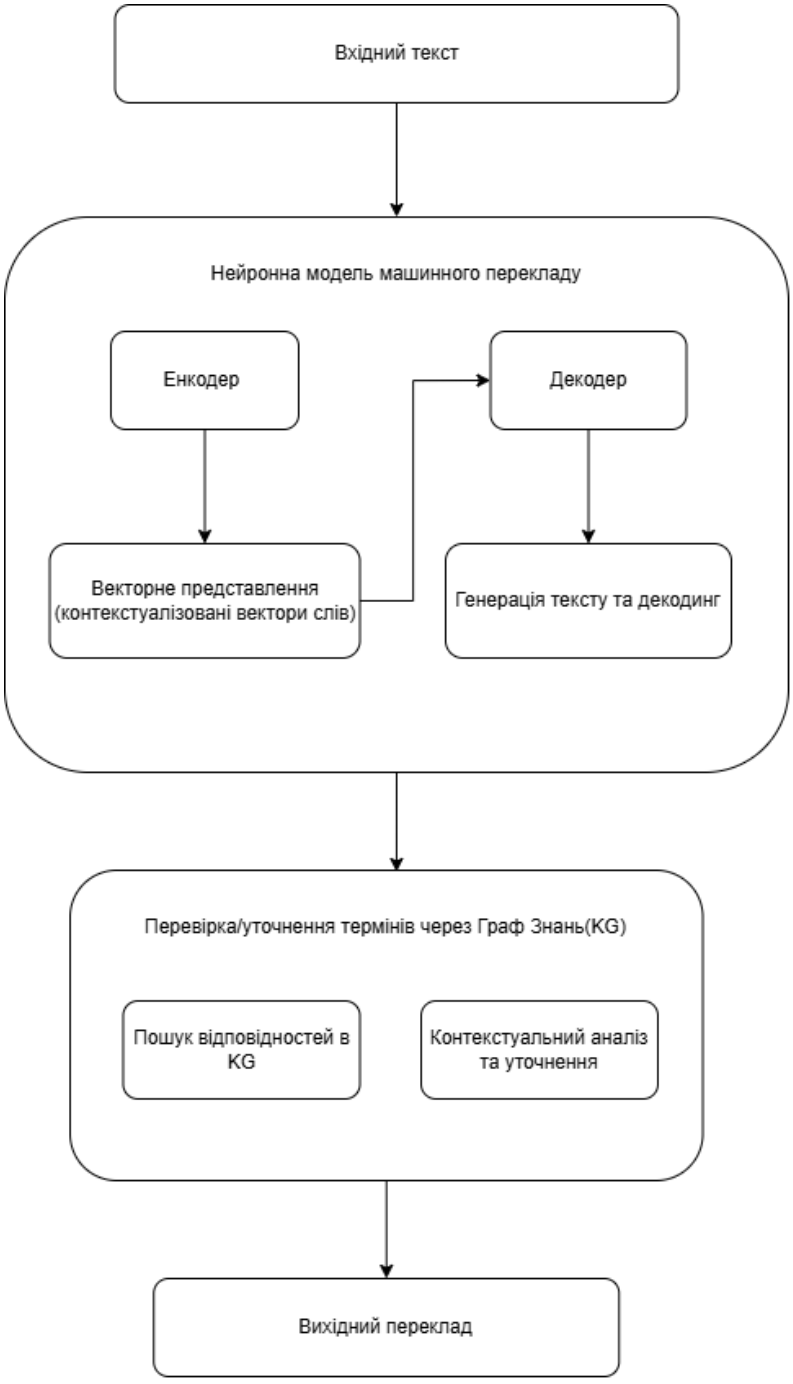

Рисунок 1.1 – Діаграма структури машинного перекладу з графами знань

Ця діаграма відображає, як система машинного перекладу отримує вхідний текст, використовує нейронну модель для початкової обробки, а потім звертається

до графу знань(Knowledge Graph) для уточнення та корекції перекладу термінів або фраз.

1.2 Тенденції та перспективи, практична цінність і виклики

Якщо дивитися на тенденції розвитку машинного перекладу, то вони рухаються у напрямку комплексної інтеграції, тобто у системах частіше поєднують переваги нейронних підходів і семантичних графів знань. Також очікується, що завдяки збільшенню обчислювальних потужностей та появою нових архітектур, якість перекладу зростатиме ще більше.

Особливо це актуально для медицини, де важливо використовувати вбудовані бази даних з термінологією і співпрацювати з лікарями та фармацевтами, щоб налаштовувати системи машинного перекладу під їхні специфічні потреби. Важливо розуміти, що навіть найкращі системи залишаються інструментами та вимагають суворого контролю якості. З іншого боку, якщо проблема полягає в перекладі великої кількості технічної документації, то системи можуть потребувати налаштування під відповідні стандарти та постійно оновлювати, бо технології не стоять на місці.

Важливо виділити основні виклики. По-перше, це проблема з навчальними даними, а саме недостатньо просто мати великий обсяг текстів, також необхідна і їх якість та узгодженість. Зі збільшенням обсягів інформації росте й різноманіття стилів і мовних особливостей, тому треба ретельно відбирати дані, робити коректуру, можливо навіть залучення експертів, щоб підготувати репрезентативну тренувальну вибірку. По-друге, адаптація до вузьких доменів, таких як медицина, правова сфера чи техніка, залишається складним завданням, бо загальна модель, яка показує непогані результати на побутових текстах, може не впоратися з глибинною термінологією, якої не було у її навчальній виборці. По-третє, із розвитком технологій постає питання етики й відповідальності, що чи можемо ми цілком покластися на автоматичний переклад у критичних сферах, чи нам потрібен обов’язково людський контроль?

Якщо говорити про перспективи, то сучасні дослідження в галузі машинного перекладу демонструють значний потенціал у поєднанні різноманітних джерел знань та методів машинного навчання для досягнення високої якості та надійності перекладу. Це відкриває шлях до розробки гібридних моделей, здатних адаптувати свої механізми обробки залежно від специфіки тексту, наприклад, його тематики чи термінологічної насиченості.

Розвиток нейронних мереж для машинного перекладу (NMT), особливо після появи таких архітектур, як Transformer, кардинально змінив підходи до машинного перекладу, забезпечивши значне підвищення якості порівняно зі статистичними методами (SMT)[2].

Для ілюстрації загальних тенденцій та порівняння ефективності різних підходів до МП, розглянемо гіпотетичний сценарій (див. рис. 1.2). У цьому сценарії аналізується якість перекладу трьох типів текстів: загальнотематичних (наприклад, новини), технічних (інструкції, специфікації) та медичних (медичні висновки, довідники хвороб та препаратів). Далі для кількісної оцінки відповідності машинного перекладу еталонному, побудуємо графік порівняння середніх показників BLEU для умовних трьох моделей: SMT, NMT і NMT, доповненої графом знань.

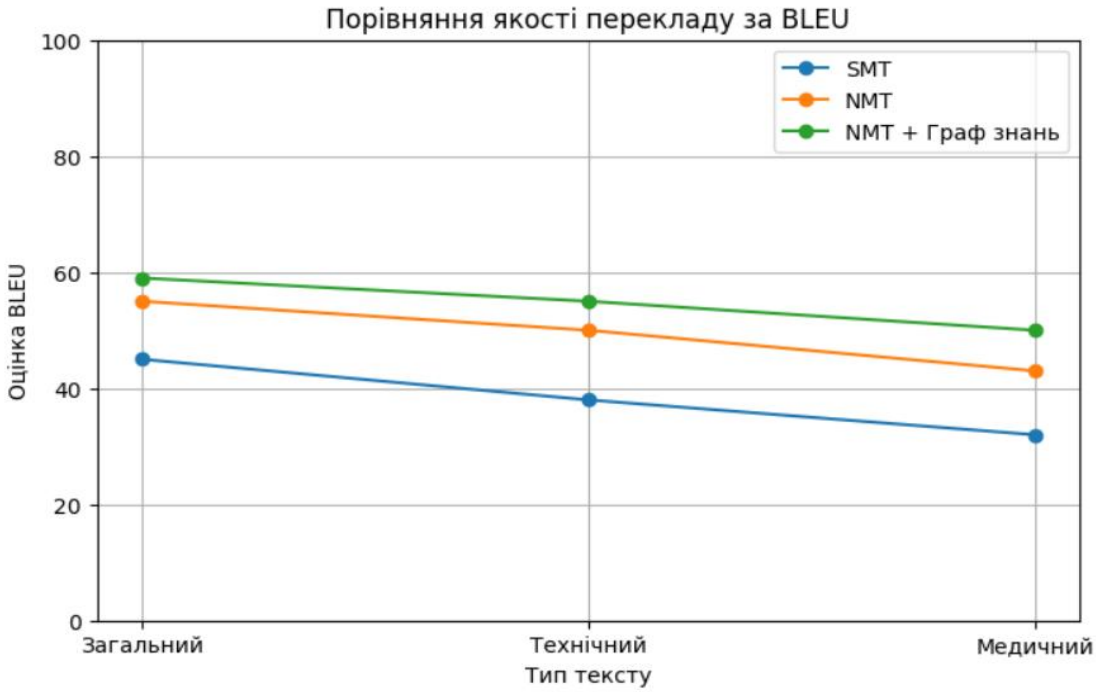

Рисунок 1.2 – Порівняння різних моделей машинного перекладу

У цій ситуації ми бачимо, що для загальних текстів SMT може демонструвати нормальні результати через те, що досить часто там трапляються поширені фрази, які модель вивчила на великій кількості текстів. Але от коли справа доходить до технічних, а особливо медичних текстів, SMT починає втрачати позиції. NMT демонструє кращі результати завдяки глибинним зв'язкам між словами й контекстом, але й в неї є проблеми з медичними текстами. Але інтеграція NMT з графами знань дозволяє трохи виправити ситуацію з рідкісною термінологією, завдяки чіткій семантичній структурі й можливостю взаємодіяти з онтологію чи наприклад, медичною базою знань, що підкріплює контекст.

Таким чином, машинний переклад зараз вже досить непогано справляється із загальними текстами, типу новин чи веб-контенту. Але з складними темами, як медицина, юридична діяльність чи техніка, все ще є проблеми з термінологією. Тому вся суть нових досліджень в тому, як вони вирішують ці проблеми: чи то оптимізують моделі, чи додають більше семантики, чи покращують обробку термінів, чи взагалі пропонують якісь нові підходи до розуміння контексту.

2 ОГЛЯД Й АНАЛІЗ ЛІТЕРАТУРНИХ, НАУКОВИХ ДЖЕРЕЛ

Тема машинного перекладу з використанням графів знань продовжує набувати ваги, особливо в технічних галузях, де точність відтворення термінології та специфіки контексту має критичне значення. Для формування цього огляду, було проаналізовано декілька наукових публікацій і звітів, що перебувають у відкритому доступі. Відбір джерел відбувався за принципами авторитетності, тобто наявності рецензованих публікацій, підтримки від наукових установ та грантових програм, актуальності, об’єктивності, тобто уникнення рекламних чи комерційних матеріалів без наукового підґрунтя та достовірності, а саме наявності експериментів, верифікація гіпотез тощо.

У публікації [3], розміщеній на CEUR-WS, автори зосереджуються на тому, як застосування графів знань може підвищити якість машинного перекладу, зокрема для нейронних моделей(NMT). Вони роблять акцент на кількох ключових аспектах. Насамперед ідеться про точність спеціалізованої термінології. На думку дослідників, якщо модель має відомості про семантичні зв’язки між поняттями, тоді зменшується ризик помилок у багатозначних термінах, що особливо важливо для технічних або медичних текстів. Також завдяки цьому методові можна розв’язати проблему контекстної двозначності, адже графи знань дають змогу явно описувати відношення між сутностями. Тобто модель NMT точніше передає зміст та краще розпізнає термінологію. У межах проведених експериментів у [3] порівнювали базові моделі без графів із тими, де вони були встроєні в архітектуру на рівні вхідних та проміжних шарів нейронної мережі.

Найкращі результати показали саме гібридні варіанти(див. рис. 2.1), що підтверджує ефективність використання семантичних даних. Водночас автори вказують, що створення таких графів знань вимагає ретельного експертного супроводу та великих часових ресурсів, особливо коли йдеться про специфічну галузь. Крім того, вони переважно орієнтувалися на англомовні дані, тому для інших мов, наприклад, української, потрібні додаткові мовно-орієнтовані дослідження.

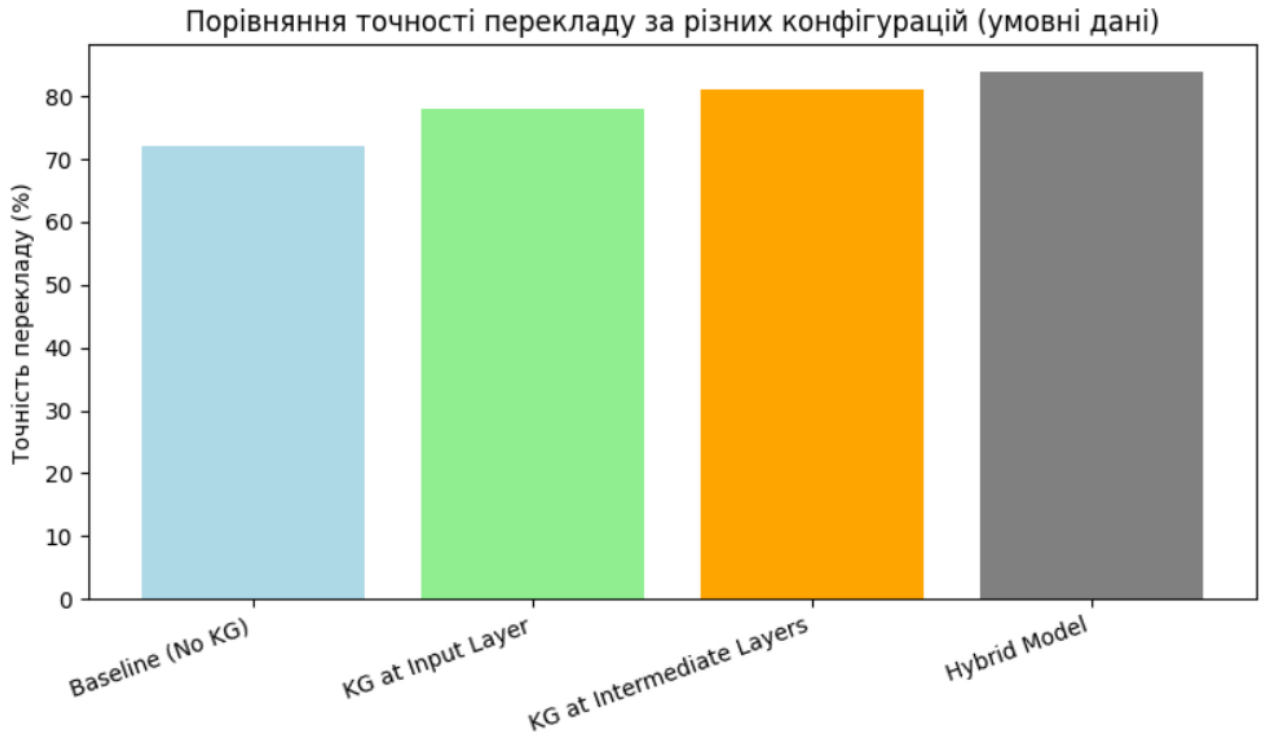

Рисунок 2.1 – Результати експеременту

Також варто звернути увагу на кілька публікацій, що висвітлюють тематику з інших боків. Наприклад, У статті [2] автори наголошують, що звичайні моделі, які базуються на статистичних закономірностях у текстах, чи то класичні Statistical Machine Translation, чи сучасні глибинні NMT-підходи, часто не здатні якісно обробляти рідкісну термінологію або нестандартні мовні звороти. Вони пояснюють, що проблема тут полягає в обмеженості суто лінгвістичних або статистичних ознак: коли система не має додаткової інформації про те, які поняття є «спорідненими», вона не завжди може правильно вибрати відповідник у цільовій мові. Тому автори пропонують інтегрувати онтологічні знання, які найчастіше реалізуються у вигляді графів знань(Knowledge Graphs), безпосередньо у процес машинного перекладу (див. рис. 2.2). Основний акцент статті [2] полягає в тому, як внедриди ці формальні зв’язки між концептами в архітектуру нейронних мереж. Вони наводять декілька підходів. Перший заснований на попередній обробці: модель отримує на вхід не лише текст, а й набір тегів або векторних репрезентацій, що вказують на зв’язки між сутностями, а другий використовує постобробку, тобто коли перекладений текст звіряють із онтологічною базою, аби виправити

непослідовності в термінології чи смислові суперечності. Також ще є і третій варіант, котрий комбінує обидва підходи та дозволяє поєднати глобальні семантичні підказки, граф знань, зі звичайними механізмами уваги(attention) у нейронних мережах.

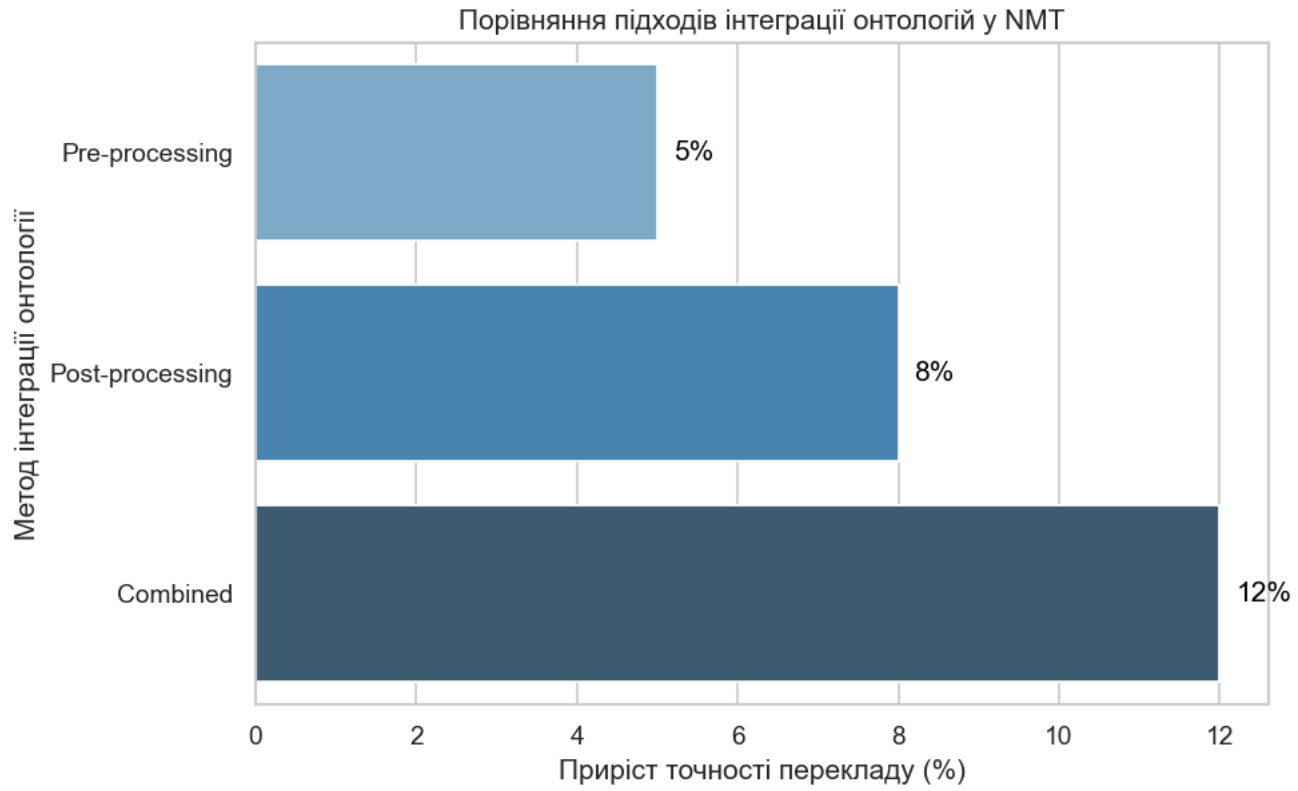

Рисунок 2.2 – Порівняння підходів інтеграції онтологій

З експериментів, про які йдеться в статті [2], можна зробити кілька висновків. По-перше, навіть часткова інтеграція онтологій, тобто коли є лише базові зв’язки між поняттями, дає помітне підвищення точності перекладу рідковживаної лексики або спеціальних термінів. По-друге, найбільшого ефекту досягають ті моделі, де онтологічні дані застосовуються в реальному часі, тобто модель динамічно звертається до графа під час процесу перекладу. По-третє, залишається складність із масштабуванням таких систем, особливо якщо потрібно паралельно підтримувати кілька онтологічних баз або швидко оновлювати термінологію. Це, свідчить про те, що майбутні дослідження чи розробки в цьому напрямку, мають зосередитися на напівавтоматичних чи автоматичних методах поповнення й перегляду онтологій.

Стаття [4] наголошує на дещо іншому аспекті, а саме на важливості контролю стилю у процесі машинного перекладу. Якщо у випадку [2] основна увага приділяється побудові семантичних зв'язків та термінологічній точності, то тут акцент переноситься на те, як зберегти або змінити тон, формальність та інші стилістичні характеристики тексту.

Автори [4] посилаються на те, що класичні NMT-системи не завжди коректно передають стиль, адже орієнтуються здебільшого на лексико-граматичний рівень, а не на прагматичний чи дискурсивний(див.рис.2.3). Наприклад, у бізнес-кореспонденції важливо зберігати формальний тон, тоді як у текстах маркетингового спрямування може бути доречнішим більш розмовний чи креативний стиль. У контексті англо-українського перекладу це особливо актуально, адже українська мова має низку граматичних форм, що відображають рівень ввічливості та формальності, наприклад, звертання на «Ви» чи «ти», вибір певних засобів вираження, вживання офіційно-ділового синтаксису тощо.

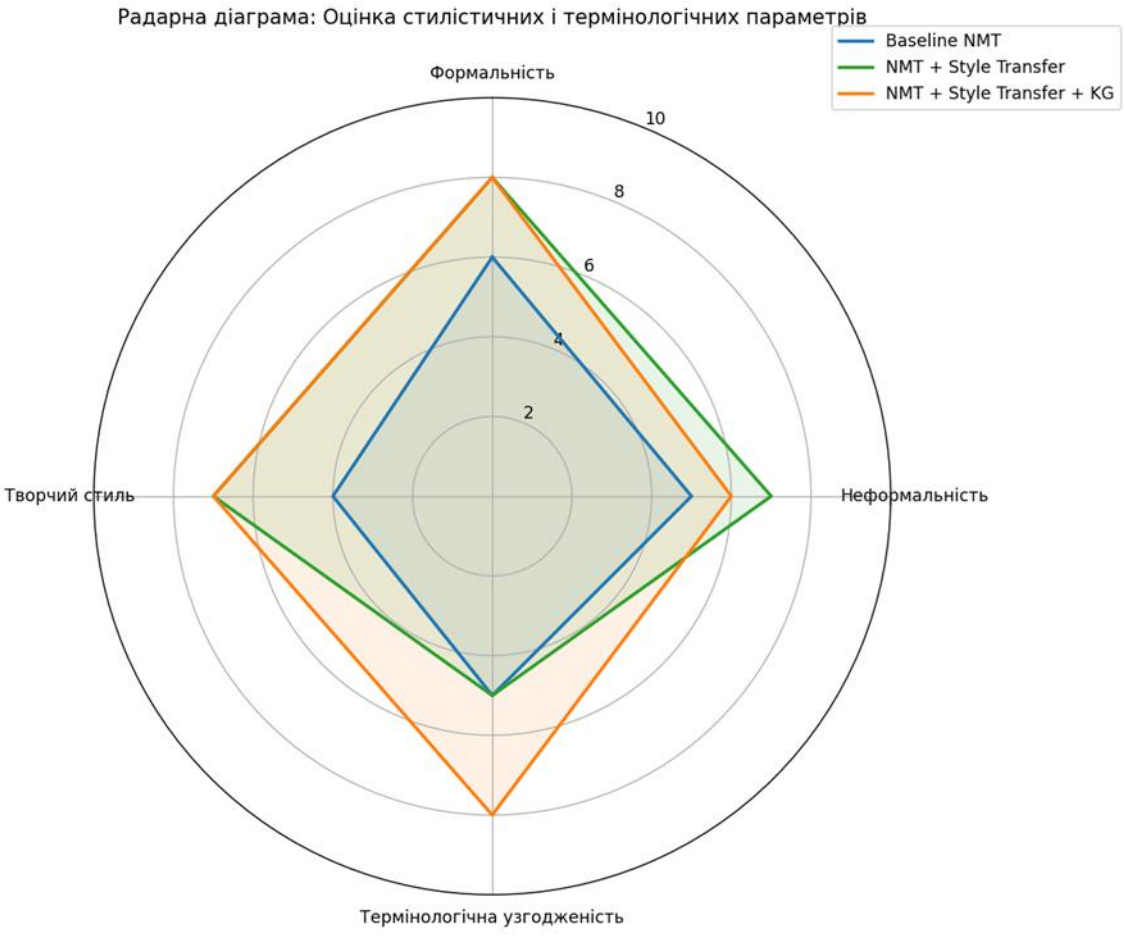

Рисунок 2.3 - Оцінка стилістичних і термінологічних параметрів

У своїй роботі [4] дослідники пропонують застосувати техніку так званого «style transfer» із додатковим компонентом «controllability». Це означає, що користувач або система може задати, яким має бути стильовий вихід перекладу: формальний, неформальний, науковий, а можливо, щось більше на кшталт публіцистичного. Вони показують, що такі підходи, зокрема умовний трансфер, коли на вході до нейронної мережі додається спеціальний тег чи змінна стилю, дозволяють суттєво впливати на тональність і рівень офіційності в перекладеному тексті. Цікаво, що дослідники відзначають певні складнощі з термінологічною узгодженістю: під час зміни стилю система часом порушує послідовність у відтворенні галузевих термінів.

Саме тут інтеграція графів знань могла б відіграти важливу роль. Як зазначають автори [4], якщо в майбутньому об’єднати ідеї «style transfer» із семантичним контролем, то можна отримати перекладацькі системи, що одночасно дотримуватимуться чіткої термінології, завдяки інформації з онтології та адаптуватимуть загальну стилістику під цільову аудиторію. Це було б надзвичайно корисно для різних сфер, наприклад, від офіційних публікацій і договорів до наукових чи популярних статей.

У публікації, розміщеній на IEEE Xplore [5], наведено приклади масштабного впровадження семантичних технологій у межах великих корпоративних середовищ, де узгодженість перекладів є критично важливою як для внутрішньої, так і для зовнішньої документації. Автори демонструють, як графи знань та системи таксономій допомагають суттєво зменшити кількість помилок та неточностей під час перекладу великого масиву технічних та маркетингових матеріалів, а також забезпечують гнучке та швидке оновлення ключової інформації в цих документах. Зокрема, подані кейси показують, наскільки універсальними можуть бути ці підходи й дають змогу уявити, як адаптувати описані рішення для різних мов або галузей із урахуванням їхньої специфіки та термінології.

Ці дослідження є особливо актуальними, оскільки зростає потреба у багатомовному контенті з високим рівнем точності й адаптивності. Насамперед це стосується технічних, юридичних чи медичних текстів, де помилка може

спричинити критичні наслідки, аж до загрози життю користувачів або втрати репутації, а також маркетингових та академічних матеріалів, у яких значну роль відіграють стиль та рівень формальності. Важливість поєднання семантичного аналізу з механізмами стилістичного контролю доповнюється потенціалом графів знань, що здатні надавати перекладацькій системі вичерпну інформацію про термінологічні та контекстуальні зв’язки між поняттями. Завдяки цьому перекладач, чи то автоматизований, чи то інтерактивний, дістає можливість точніше відтворювати доменні терміни та одночасно варіювати стиль згідно з вимогами конкретного дискурсу.

Узагальнюючи наведені публікації, можна зробити висновок, що застосування графів знань стає одним із ключових інструментів для підвищення ефективності машинного перекладу, особливо коли йдеться про вузько направлену термінологію чи складні стилістичні вимоги. Аналізовані праці доводять, що інтеграція онтологічних структур у нейронні моделі допомагає уникати помилок у рідковживаних або багатозначних термінах, а також гнучкіше змінювати тональність перекладів у залежності від дискурсу чи цільової аудиторії.

3 ПОСТАНОВКА ЗАДАЧІ

Цей комплексний курсовий проект ставить перед собою завдання теоретичного дослідження застосування моделей графового представлення знань для задач машинного перекладу, але з метою більш глибокого та сфокусованого аналізу, дане дослідження буде обмежено розглядом використання графів знань для перекладу термінів у технічній документації. Таке звуження дозволить нам детально вивчити специфіку цієї області та оцінити ефективність застосування графових моделей у конкретному контексті.

Основні аспекти дослідження включають:

- аналіз існуючих методів побудови графів знань, їх архітектури та теоритичного застосування у контексті перекладу технічної термінології;
- оцінку релевантності графових моделей для забезпечення контекстуального перекладу, особливо у випадках багатозначних термінів або складних синтаксичних конструкцій;
- дослідження потенціалу інтеграції графів знань з нейронними моделями машинного перекладу(NMT), такими як Transformer[6], для вирішення проблем, що пов’язані з перекладом технічних текстів, наприклад, неоднозначність термінів та збереження семантичної цілісності тексту.

Також в ході цього дослідження в нас можуть виникати такі питання:

- які поточні обмеження існуючих методів перекладу термінології?
- які моделі графового представлення знань є найбільш придатними для задач машинного перекладу?
- як інтеграція графів знань може покращити переклад термінів у технічній документації?
- які методи оцінки ефективності графів знань є найбільш доцільними для визначення їх впливу на якість перекладу?

Щоб відповісти на ці дослідницькі запитання, у цій роботі розглядається відповідні наукові джерела про застосування графів знань для задач машинного перекладу технічної документації.

Для досягнення мети дослідження обрано методи й алгоритми, що дозволяють ефективно працювати з графами знань і забезпечити високий рівень аналізу та оцінки їхньої релевантності в задачах перекладу технічної термінології.

Методи аналізу графів знань включають підходи до побудови онтологій, які надають можливість моделювати терміни й їх зв'язки у формалізованій структурі. Використання стандартів, таких як RDF і OWL, дозволяє створювати графи, які чітко описують взаємозв'язки між термінами. Це особливо важливо для обробки технічної термінології, де зв'язки між поняттями часто є багатошаровими. Для аналізу графових структур застосовуються алгоритми пошуку, наприклад BFS і DFS. Вони дозволяють досліджувати графи для виявлення семантичних залежностей, що є критичним при роботі з багатозначними термінами.

Методи машинного перекладу базуються на нейронних моделях перекладу(NMT). Ці моделі обрано завдяки їхній здатності враховувати контекст у межах речення чи навіть всього тексту. Однак їхні обмеження у вузькоспеціалізованих галузях, таких як технічна документація, роблять необхідним інтеграцію графових моделей знань. Графи знань є додатковим джерелом семантичної інформації, яка покращує точність перекладу шляхом розширення контексту та забезпечення точного розуміння термінів.

Методи порівняльного аналізу використовуються для оцінки ефективності інтеграції графів знань у процес машинного перекладу. Для цього застосовуються метрики BLEU, TER та METEOR, які дозволяють оцінити якість перекладу. Порівняння результатів традиційних NMT-моделей із гібридними підходами допомагає об'єктивно оцінити переваги використання графів знань.

Ці обрані методи повинні забезпечити можливість детального аналізу ефективності графових моделей знань, порівняння їх із традиційними методами та виявлення їхнього впливу на якість перекладу технічної термінології.

Також можемо зазначити інструменти, які будуть використані для аналізу, моделювання та оцінки ефективності графів знань у перекладі технічної документації.

По-перше, це мова програмування для роботи з теоретичними моделями, включаючи бібліотеки для обробки графів знань, наприклад, NetworkX, RDFlib, та алгоритмів машинного перекладу таких, як Transformers від HuggingFace, це Python. Його було обрано за його універсальність та широке використання у наукових дослідженнях. Також він може бути застосована для створення візуалізацій графових структур, що дозволить краще зрозуміти їхню будову та зв’язки.

Далі зазначимо бібліотеки та фреймворки, які можна було б використовувати, для нашого досліження, це:

- TensorFlow/PyTorch, використовуються для аналізу архітектур NMT-моделей, зокрема для розгляду принципів їхньої роботи та інтеграції з графами знань;
- Graph-tool або Gephi, вони призначені для аналізу та візуалізації графів знань, дозволяючи краще зрозуміти їх структуру та взаємозв’язки;
- NLTK і SpaCy, це інструменти для попередньої обробки тексту, необхідні для підготовки корпусу технічної документації для аналізу.

Також згадаємо про джерела даних, а саме:

- використання існуючих баз, таких як Wikidata чи DBpedia, надає змогу дослідити семантичні зв’язки між поняттями без необхідності створення графів із нуля;
- перекладені вручну тексти можуть слугувати еталоном для оцінки теоретичних моделей і їхнього потенційного впливу на якість перекладу.

Зазначені інструменти дозволяють зосередитися на моделюванні та теоретичному аналізі, без необхідності реалізації практичних рішень, а також забезпечують гнучкість та глибину дослідження.

Також під час дослідження можуть виникнути такі обмеження, а саме:

- робота з великими графами знань може вимагати значних обчислювальних ресурсів, що ускладнює їх практичне застосування;

- метрики якості, такі як BLEU, не завжди враховують семантичну коректність перекладу, зосереджуючись переважно на синтаксичних аспектах;
- доступність даних для графів знань, обмежена кількість якісних графів знань для вузькоспеціалізованих галузей;
- обмеженість готових мовних моделей для української мови, що ускладнює адаптацію нейронних моделей до специфіки завдань;
- складність інтеграції графів знань у вже існуючі NMT-моделі, може бути складною через різні формати даних та технічні вимоги;
- налаштування параметрів гібридних моделей потребує детальної роботи, що може зайняти багато часу.

Цей підхід дозволить систематизувати процес дослідження, забезпечивши його обґрунтованість та підготувати основу для теоретичного розгляду методів у наступному розділі.

4 ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ

У цьому розділі ми детально розберемося з тим, як можна використовувати графи знань для покращення машинного перекладу технічної документації. По-перше, розглянемо, як взагалі будуються ці графи знань, як створюються онтології та які для цього є стандарти, наприклад, RDF та OWL. Потім перейдемо до алгоритмів, які дозволяють обробляти та аналізувати ці графи, зокрема, пошук BFS та DFS, а також методи, що допомагають знаходити зв’язки між термінами. Звичайно, не обійдеться без розгляду архітектури нейронного машинного перекладу, тут можна зосередетись на Transformer, як на одній з найбільш крутих моделей. Далі будемо досліджувати, як можна поєднати графи знань з нейронними мережами, щоб вони працювали разом та перекладали ще краще, додаючи в текст більше сенсу. Та у кінці, розпишемо про метрики, наприклад це BLEU, TER чи METEOR, за якими ми будемо оцінювати якість перекладу.

4.1 Моделі графового представлення знань

Граф знань чи Knowledge Graph є структурою, де об'єкти, такі як сутності, поняття чи події, представлені у вигляді вузлів, а їх взаємозв'язки - у вигляді ребер. Кожне ребро описує семантичні відносини між двома вузлами, що дозволяє чітко визначати зв'язки між поняттями, уніфікувати інформацію з різних джерел та забезпечувати автоматизовану обробку і пошук нових знань. Це робить графи знань корисними для вирішення завдань контекстуалізації та уточнення значень термінів у технічній документації.

На відміну від традиційних реляційних баз даних, граф знань має більш гнучку структуру, що дозволяє додавати нові вузли та зв'язки без необхідності значної перебудови схеми даних. Крім того, зв'язки в графі знань мають чітке семантичне значення, тоді як у реляційних базах це значення часто приховане або менш очевидне.

Далі розглянемо RDF(Resource Description Framework) та OWL(Web Ontology Language). RDF надає спосіб описувати ресурси у форматі «суб’єкт–предикат–

об’єкт» чи триплетів. Це дозволяє зв’язувати різні поняття та виражати семантичні залежності між ними. А OWL розширює RDF, додаючи можливість формалізувати онтології, а саме оголошувати класи, підкласи, обмеження та ролі, що описують відношення між сутностями. У прикладі (див.рис.5.1) описується концепція «викрутка»(Screwdriver) як інструмента з компонентом «руків’я»(Handle), що може бути використано для формування семантичних зв’язків у технічному перекладі.

```
@prefix ex: <http://example.org/>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.

ex:Screwdriver a ex:Tool;
               rdfs:label "Screwdriver"@en;
               ex:hasPart ex:Handle.

ex:Handle a ex:Part ;
          rdfs:label "Handle"@en.
```

Рисунок 4.1 - Приклад RDF-триплетів у форматі Turtle

Тепер для завдань технічного перекладу ми бачимо, що зручно створювати онтології, де кожен термін, наприклад, назва деталі механізму чи електронного компонента, буде вузлом графа, а відношення між ними - ребрами. Таким чином, легко відстежувати взаємозв’язки термінів. У контексті технічної документації:

- моделювати багатозначність(homonymy), тобто один термін може мати різні значення, тому потрібне уточнення контексту;
- враховувати полісемію, тобто поняття можуть мати подібні, але не ідентичні сенси;
- описувати родо-видові відносини, наприклад, «is-a», «part-of» тощо.

У процесі роботи з технічними текстами часто виникає потреба додавати нові терміни чи зв’язки[7]. Завдяки онтологічному підходу це можна робити швидко, не руйнуючи вже створену структуру. Так, якщо у тексті з’явився специфічний елемент, його можна додати як підклас або екземпляр певного класу залежно від контексту використання.

Далі для аналізу семантичних зв’язків у побудованому графі знань застосуємо класичні пошукові алгоритми:

- Depth-First Search(DFS) - це глибокий пошук, застосовується для дослідження всіх можливих гілок у графі;
- Breadth-First Search(BFS) – це пошук у ширину, зручний для визначення найкоротших шляхів між термінами або пошуку найближчих суміжних концептів.

Припустимо, що ми маємо граф G = (V, E), де V – множина вузлів чи термінів, а E – множина ребер, семантичних зв’язків. Для пошуку шляху між двома вузлами u та v можемо застосувати BFS(див.рис.5.2).

$$
\mathrm{BFS}(G,u):\begin{cases}
\text{create a queue } Q\\
\text{mark } u \text{ as visited}\\
Q.\mathrm{enqueue}(u)\\
\text{while } Q \neq \emptyset\text{:}\\
\quad w \leftarrow Q.\mathrm{dequeue}()\\
\quad \text{for each } x \text{ adjacent to } w\text{:}\\
\quad\quad \text{if } x \text{ not visited:}\\
\quad\quad\quad \text{mark } x \text{ as visited}\\
\quad\quad\quad Q.\mathrm{enqueue}(x)
\end{cases}
$$

Рисунок 4.2 - Алгоритм пошуку в ширину, записаний у формі псевдокоду

Алгоритм пошуку в ширину BFS у наведеному прикладі представлений у вигляді псевдокоду, що описує покрокову процедуру пошуку в графі. Алгоритм починається зі створення порожньої черги та позначення початкового вузла, як відвіданого, після чого цей вузол додається до черги[8]. Доки черга не порожня, з неї вилучається перший елемент, і для кожного сусіднього з ним вузла перевіряється, чи був він вже відвіданий. Якщо ні, то він позначається, як відвіданий та додається до черги. Завдяки цьому BFS обходить вузли графа пошарово, починаючи з початкового вузла.

4.2 Підходи до інтеграції графів знань

Однією з найпоширеніших NMT-архітектур є Transformer. Його ключова особливість, це механізм самоуваги чи self-attention, який дозволяє моделі аналізувати усі елементи вхідної послідовності, не втрачаючи глобального контексту. У спрощеному вигляді, функція уваги Attention задається формулою 4.1:

$$\text{Attention(Q,K,V)=softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)\text{V} \quad (4.1)$$

де Q – матриця запитів (queries),

K – матриця ключів (keys),

V – матриця значень (values),

d_k – розмірність ключів.

Цей підхід дозволяє моделі звертати увагу на різні частини речення під час перекладу кожного слова.

Інтеграція графів знань у моделі нейронного машинного перекладу може значно покращити якість перекладу, особливо в технічних текстах, де точність термінології та збереження семантичної цілісності є критично важливими. Існує декілька основних підходів до такої інтеграції, кожен з яких має свої переваги та особливості застосування. Нижче розглянуто три основні сценарії інтеграції, а саме збагачення вхідних ембедингів, використання зовнішньої пам’яті та постобробка перекладу.

Збагачення вхідних ембедингів полягає у доповненні векторних представлень слів чи ембедингів додатковою семантичною інформацією, отриманою з графа знань. Це дозволяє моделі NMT краще розуміти контекст та значення термінів, особливо багатозначних або спеціалізованих термінів у технічній документації. На етапі попередньої обробки тексту кожне слово або термін ідентифікується в графі знань. До його векторного представлення додаються додаткові атрибути, такі як тип сутності, наприклад, інструмент, деталь, семантичні зв’язки, наприклад, має частину, виготовлено з, та інші релевантні

властивості. Це дозволяє ембедингам містити більше інформації про кожен термін, що підвищує їхню інформативність для моделі під час перекладу. Розглянемо приклад(див.рис 4.3) з використанням бібліотеки NetworkX для роботи з графами знань та PyTorch для обробки ембедингів.

```python
import networkx as nx
import torch

# Створення графа знань
G = nx.Graph()
G.add_node("Викрутка", type="Інструмент", material="Сталь")
G.add_node("Руків'я", type="Частина", material="Пластик")
G.add_edge("Викрутка", "Руків'я", relation="маєЧастину")

# Функція для отримання атрибутів вузла
def get_node_attributes(node):
    return list(G.nodes[node].values())

# Приклад збагачення ембедингів
word = "Викрутка"
attributes = get_node_attributes(word)
# Перетворення атрибутів у вектор (наприклад, one-hot encoding)
# Тут показано спрощення
attribute_vector = torch.tensor([1, 0, 1], dtype=torch.float32)

# Початковий ембединг слова
initial_embedding = torch.rand(300)  # Приклад випадкового ембедингу

# Збагачений ембединг
enhanced_embedding = torch.cat((initial_embedding, attribute_vector), dim=0)
```

Рисунок 4.3 - Приклад використання NetworkX та PyTorch для обробки ембедингів

У цьому прикладі до початкового ембедингу слова «Викрутка» додаються додаткові атрибути «type» та «material», що збагачує його семантичну інформацію, дозволяючи моделі NMT краще розуміти контекст та значення терміну під час перекладу.

Далі розглянемо зовнішня пам'ять(External Memory), яка передбачає використання графа знань як додаткового джерела інформації, до якого модель може звертатися під час процесу перекладу. Це дозволить моделі отримувати

додатковий контекст чи уточнення щодо конкретних термінів у реальному часі. Під час генерації перекладу модель може робити запити до графа знань для отримання додаткової інформації про терміни або їхні зв'язки. Наприклад, якщо термін має декілька значень, модель може використовувати зовнішню пам'ять для визначення правильного значення залежно від контексту. Ми можемо розглянути приклад інтеграції зовнішньої пам'яті з використанням бібліотеки Transformers від HuggingFace(див.рис. 4.4).

```python
from transformers import MarianMTModel, MarianTokenizer
import networkx as nx

# Завантаження моделі та токенізатора
model_name = "Helsinki-NLP/opus-mt-uk-en"
tokenizer = MarianTokenizer.from_pretrained(model_name)
model = MarianMTModel.from_pretrained(model_name)

# Створення графа знань
G = nx.Graph()
G.add_node("ручка", type="Частина", material="Пластик")
G.add_node("ручка", type="Інструмент", material="Метал")
G.add_edge("ручка", "Викрутка", relation="частина")

# Функція для отримання контексту з графа знань
def get_context(term):
    if term in G:
        neighbors = list(G.neighbors(term))
        return ", ".join(neighbors)
    return ""

# Вхідний текст з терміном
src_text = "Викрутка має зручну ручку."

# Токенізація
tokens = tokenizer(src_text, return_tensors="pt")

# Генерація перекладу
translated_tokens = model.generate(**tokens)
translated_text = tokenizer.decode(translated_tokens[0], skip_special_tokens=True)

# Отримання контексту для терміна "ручка"
context = get_context("ручка")

# Використання контексту для корекції перекладу
if "handle" in translated_text and context:
    translated_text += f" (Context: {context})"

print(translated_text)
```

Рисунок 4.4 - Приклад інтеграції зовнішньої пам'яті з Transformers від HuggingFace

На цьому рисунку, модель генерує переклад для речення «Викрутка має зручну ручку», після генерації перекладу модель звертається до графа знань для отримання додаткового контексту щодо терміна «ручка» та додає цю інформацію до перекладу для уточнення. Таке використання зовнішньої пам’яті для перекладу має переваги, як динамічне отримання контексту, що дозволяє моделі отримувати актуальну інформацію за потреби, та підтримка багатозначності, завдяки чому можна розрізняти різні значення термінів залежно від контексту. Але існують й недоліки, а саме складність інтеграції тому, що необхідно ефективно синхронізувати модель з графом знань для забезпечення швидкого доступу до інформації, а також обробка великих графів може вимагати значних обчислювальних ресурсів та оптимізації запитів[9].

Тепер перейдемо до постобробки перекладу, вона передбачає перевірку та корекцію вже згенерованого машинним перекладом тексту з використанням графа знань. Це дозволяє виявити та виправити помилки чи неточності в перекладі термінів, забезпечуючи їхню відповідність семантичним зв’язкам у графі. Після генерації перекладу, модель перевіряє кожен термін чи ключове слово на відповідність семантичним зв’язка. Якщо виявляється, що перекладений термін не відповідає контексту чи має неправильне значення, його замінюють на більш релевантний варіант, отриманий з графа знань. Розглянемо приклад в коді(див.рис.5.5) для постобробки перекладу з використанням графа знань для перевірки та корекції термінів.

Постобробка перекладу має свої переваги, наприклад, такі як підвищення якості перекладу шляхом виправлення помилок, що виникають під час його генерації та забезпечення семантичної узгодженості, що гарантує відповідність перекладених термінів їхньому контексту та значенню у графі знань [10]. Проте також існують недоліки, зокрема додатковий етап обробки, який вимагає додаткових обчислювальних ресурсів та часу, а також необхідність точності графа знань, адже ефективність постобробки безпосередньо залежить від якості та повноти цього графа.

```python
import networkx as nx
# Створення графа знань
G = nx.Graph()
G.add_node("ручка", type="Частина", material="Пластик")
G.add_node("ручка", type="Інструмент", material="Метал")
G.add_edge("ручка", "Викрутка", relation="частина")

# Перекладений текст
translated_text = "The screwdriver has a comfortable handle."

# Функція для перевірки відповідності терміну контексту
def validate_term(term, context_graph):
    if term in context_graph:
        # Перевірка типу терміну (наприклад, чи є "handle" частиною інструменту)
        attributes = context_graph.nodes[term]
        if attributes.get("type") == "Частина":
            return True
    return False

# Функція для корекції терміну
def correct_term(term, context_graph):
    # Пошук релевантного перекладу з графа знань
    for neighbor in context_graph.neighbors(term):
        attributes = context_graph.nodes[neighbor]
        if attributes.get("type") == "Інструмент":
            return "зручну ручку для інструментів"
    return term

# Розбиття тексту на слов
words = translated_text.split()

# Постобробка кожного слова
for i, word in enumerate(words):
    if word.lower() == "handle":
        if not validate_term(word.lower(), G):
            corrected = correct_term(word.lower(), G)
            words[i] = corrected

# Об'єднання слів у фінальний переклад
final_translation = " ".join(words)
print(final_translation)
```

Рисунок 4.5 - Постобробка перекладу з використанням knowledge graph

У прикладі(див.рис.4.5) після генерації перекладу терміна «handle» виконується перевірка його відповідності контексту з графа знань. Якщо термін не відповідає необхідним критеріям, він коригується для забезпечення точності та релевантності перекладу.

Підсумовуючи ми може сказати, що інтеграція графів знань у моделі нейронного машинного перекладу значно розширює можливості покращення якості перекладу, особливо технічних текстів. Розглянуті методи, мають свої переваги та недоліки, що потребують уваги при їх застосуванні. Збагачення ембедингів сприяє кращому розумінню контексту та значення термінів, що своєю чергою підвищує точність перекладу. Застосування зовнішньої пам’яті забезпечує динамічний доступ до додаткової інформації, що є особливо корисним для термінів з кількома значеннями. Постобробка гарантує відповідність перекладу семантичним зв’язкам у графі знань, виправляючи можливі помилки. Поєднання цих підходів здатне значно покращити результати машинного перекладу, забезпечуючи високий рівень точності та релевантності, що є критично важливим для технічної документації.

4.3 Методи порівняльного аналізу та оцінка якості перекладу

Для оцінки ефективності інтеграції графів знань у моделі нейронного машинного перекладу необхідно застосувати систематичний підхід до порівняльного аналізу. Тому цей розділ описує методи та метрики, які використовуються для кількісної оцінки якості перекладу та пропонує експериментальну методологію для порівняння традиційних NMT-моделей із гібридними підходами, що інтегрують графи знань. Для кількісної оцінки якості перекладу широко застосовуються кілька стандартних метрик, кожна з яких має свої переваги та недоліки [11]. Нижче розглянемо такі основні метрики, а саме BLEU, TER та METEOR.

BLEU(Bilingual Evaluation Understudy) - одна з найпопулярніших метрик для оцінки якості машинного перекладу. Вона порівнює n-грами, послідовності з n слів, машинного перекладу з відповідними n-грамами у еталонних або «ручних» перекладах. Автоматична оцінка перекладу за допомогою метрики BLEU [12] має переваги, такі як об'єктивність, оскільки не потребує людської участі, та швидкість, що дозволяє швидко обчислювати оцінку для великих обсягів даних. Однак, існують недоліки, BLEU чутливий до лексичних варіацій, тобто не враховує

синоніми та різні способи вираження однієї й тієї ж ідеї. Крім того, він не проводить семантичної оцінки, фокусуючись лише на відповідності n-грамів, ігноруючи загальний зміст тексту. Формально, BLEU обчислюється, за формулою 4.2, як середнє геометричне точності n-грамів з введенням штрафу(brevity penalty) за короткі переклади.

$$BLEU = BP \cdot exp(\sum_{n=1}^{N} w_n log p_n) \quad (4.2)$$

де p_n - точність n-грамів,

w_n - ваги для кожного n-граму,

BP - brevity penalty.

Далі розгялнемо метрику TER(Translation Edit Rate), за формулою 4.3, вона вимірює мінімальну кількість операцій редагування, таких як вставка, видалення та заміна, необхідних для того, щоб привести машинний переклад у відповідність до відповідного. Серед переваг TER можна виділити інтуїтивність, оскільки вона базується на фактичних змінах у тексті, та чутливість до помилок, що дозволяє добре відобразити різницю між різними варіантами перекладу. Однак, як й BLEU, вона має недоліки, зокрема не враховує семантику, тобто не оцінює смислову адекватність перекладу, а також є відносно важкою в обчисленні, що потребує оптимізації при роботі з великими обсягами даних.

$$\text{TER} = \frac{Number\ of\ edits}{Number\ of\ words\ reference} \quad (4.3)$$

Наступною буде метрика METEOR [13], на відміну від інших, вона враховує не тільки точність n-грамів, але й лематизацію та синонімічні варіації, що робить її більш чутливою до семантичних відмінностей між текстами. Вона поєднує в собі показники точності та повноти, використовуючи модульну архітектуру для обробки різних мовних аспектів. Серед переваг METEOR можна виділити семантичну чутливість, завдяки якій враховуються синоніми та різні граматичні

форми слів, а також гнучкість, що дозволяє налаштовувати її для різних словесних пар. В неї також є недоліки, а саме складність обчислення, адже вона є більш складною для реалізації порівняно з BLEU та TER, а також залежність від налаштувань, оскільки результати оцінювання можуть варіюватися залежно від обраних параметрів.

Порівняння з еталонними перекладами відбувається поетапно, додаючи на кожному етапі лише ті n-грами, що не мали відповідностей раніше. Після останнього етапу обчислюються точність P та повнота R за формулами 4.4 та 4.5:

$$P = m / w_t \tag{4.4}$$

де m — кількість спільних n-грамів у машинному та еталонному перекладах,

w_t— загальна кількість n-грамів у машинному перекладі.

$$R = m / w_r \tag{4.5}$$

де w_r — загальна кількість n-грамів в еталонному перекладі.

За формулою 4.6, METEOR включає обчислення точності та повноти з врахуванням лематизації та синонімів, де вага повноти в 9 разів більша за вагу точності:

$$METEOR = \frac{10 * P * R}{R + 9 * P} + Penalty \tag{4.6}$$

де Penalty враховує порядок слів та інші мовні аспекти.

Для оцінки впливу інтеграції графів знань на якість машинного перекладу пропонується провести порівняльний аналіз між традиційною моделлю NMT та гібридною моделлю, що використовує графи знань. Нижче ми опишемо запропонований експериментальний підхід та представлені гіпотетичні результати для ілюстрації.

Порівнюються дві моделі машинного перекладу: базова(Transformer) та гібридна(Transformer + граф знань):

- Моделі:
  - базова, це стандартний Transformer;
  - Гібридна ,це Transformer зі збагаченими ембедингами з графа знань;
- дані ми беремо з технічної документації, українська та англійська, з ручними еталонними перекладами;
- базова модель навчається на стандартних даних, а гібридна з використанням графа знань;
- переклад тестових речень та оцінка за метриками BLEU, TER, METEOR для порівняння результатів.

Нижче наведено таблицю з гіпотетичними результатами порівняння базової та гібридної моделей(див. табл. 4.1). Ці результати призначені для демонстрації можливого покращення якості перекладу при використанні графів знань. В реальному дослідженні результати можуть відрізнятися залежно від якості графа знань, параметрів моделі та обсягу тренувальних даних.

Таблиця 4.1 – Гіпотетичні результати порівняння базової та гібридної моделей

Підхід	BLEU	TER	METEOR
Базова NMT(Transformer)	31.5	0.57	0.43
NMT + граф знань(гібрид)	35.2	0.52	0.48

Результати оцінювання показують покращення якості перекладу гібридної моделі порівняно з базовою. А саме це ми бачимо коли є зростання BLEU з 31.5 до 35.2 свідчить про кращу відповідність перекладів еталонним завдяки збільшенню кількості релевантних n-грамів, а зниження TER з 0.57 до 0.52 вказує на зменшення необхідних редагувань для досягнення еталонної якості, що свідчить про підвищення точності, ну а збільшення METEOR з 0.43 до 0.48 демонструє

покращення семантичної адекватності завдяки кращому врахуванню контексту та синонімів.

Аналіз гіпотетичних результатів демонструє потенціал інтеграції графів знань для покращення якості машинного перекладу. Зокрема, гібридна модель показує покращення у всіх трьох метриках, що свідчить про більш точні та семантично адекватні переклади. Це підтверджує, що додаткові семантичні зв’язки та контекст, надані графами знань, сприяють кращому розумінню та обробці технічних термінів. Однак важливо враховувати, що реальні результати можуть варіюватися залежно від якості графів знань, методів інтеграції та специфіки технічної документації. Тому необхідно провести реальні експерименти для підтвердження ефективності запропонованих підходів.

5 ПРОГРАМНА РЕАЛІЗАЦІЯ

Переходячи до програмної реалізації, спочатку розглянемо основні кроки виконання дослідження. На першому етапі здійснюється завантаження датасету з паралельними реченнями для машинного перекладу(див. рис. 5.1). Далі виконується базова обробка тексту, що включає очищення від зайвих символів. Після цього дані розподіляються на навчальний, валідаційний та тестовий набори. Для реалізації моделі перекладу використовується архітектура Transformer з бібліотеки Hugging Face. Після навчання базової моделі відбувається інтеграція графа знань, який збагачує модель семантичними зв'язками. Якість перекладу оцінюється за допомогою метрик BLEU, METEOR та TER.

Спочатку нам потрібно взяти датасет з термінами у технічних, наукових та публіцистичних текстах, такий датасет ми знайшли на сайті huggingface під назвою «trondizzy/uk_en_test_small» [14].

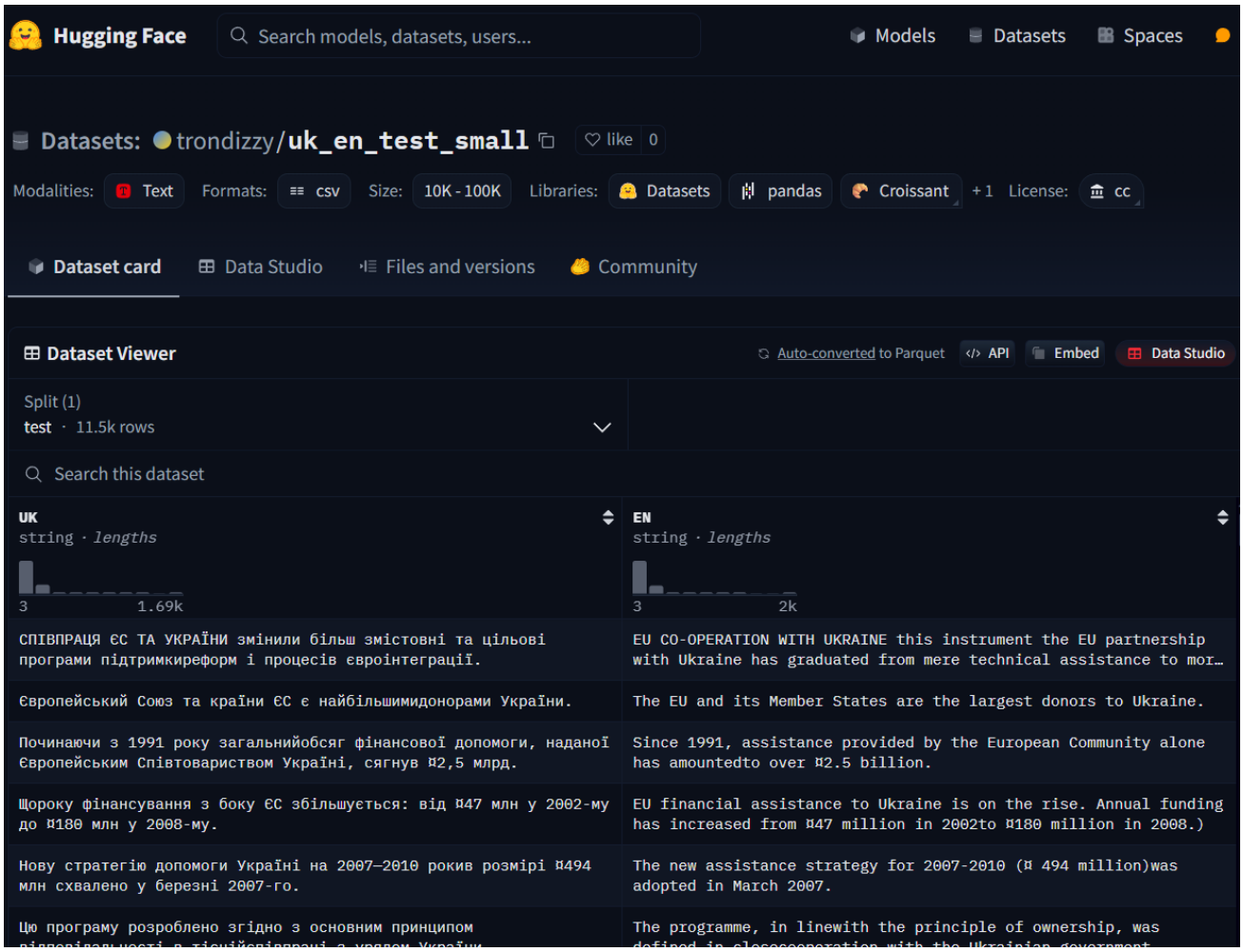

Рисунок 5.1 – Датасет «trondizzy/uk_en_test_small»

Далі треба зробити попередню обробку датасету, видалити зайві пробіли та пунктуацію за допомогою регулярних виразів, а потім розділити на валідаційний, навчаємий та тестовий набори(див. рис. 5.2).

data > dataset.py > ...

```python
import re
from datasets import load_dataset, DatasetDict
from utils.preprocess import preprocess_text
from config.config import Config

def load_and_preprocess_dataset():
    """Loads and preprocesses the dataset."""
    print("Loading dataset...")
    ds = load_dataset(Config.DATASET_NAME)

    print("Preprocessing dataset...")
    ds = ds.map(lambda x: {'EN': preprocess_text(x['EN']), 'UK': preprocess_text(x['UK'])})

    print("Splitting dataset...")
    train_testvalid = ds['test'].train_test_split(test_size=0.2)
    test_valid = train_testvalid['test'].train_test_split(test_size=0.5)

    return DatasetDict({
        'train': train_testvalid['train'],
        'validation': test_valid['train'],
        'test': test_valid['test']
    })
```

Рисунок 5.2 – Код для обробки датасету та розбиття на три частини

На рисунку 5.2 ми можемо побачити код, котрий завантажує датасет за допомогою функції «load_dataset», використовуючи ім'я, яке визначене в конфігураційному файлі. Після цього кожен запис в цьому датасеті обробляється функцією «preprocess_text», яка очищає текст, наприклад, видаляючи зайві символи та виконуючи інші перетворення. Потім датасет розбивається на три частини: спочатку тестовий набір розділяється на 80% для тренування та 20% для тестування, а потім ці 20% тестового набору розбиваються на валідаційний та тестовий набори по 50%. На виході повертається об'єкт, що містить три набори: train, validation та test, готові для подальшого використання в навчанні моделі.

Наступним кроком вже можна перейти до використання моделі Transformer з бібліотеки Hugging Face для реалізації машинного перекладу. Щоб пришвидшити процес навчання, застосовуємо графічний процесор, для цього перевіряємо його

доступність, після чого здійснюється завантаження токенізатора та самої моделі перекладу (див. рис. 5.3).

```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

Рисунок 5.3 – Перевірка наявності графічного процесора

```
class TransformerModel:
    def __init__(self):
        print("Loading transformer model...")
        self.tokenizer = AutoTokenizer.from_pretrained(Config.MODEL_NAME)
        self.model = AutoModelForSeq2SeqLM.from_pretrained(Config.MODEL_NAME).to(device)

    def _tokenize_function(self, examples):
        """Tokenizes the dataset properly for seq2seq"""
        inputs = self.tokenizer(
            examples['EN'],
            max_length=512,
            truncation=True,
            padding='max_length'
        )
        with self.tokenizer.as_target_tokenizer():
            labels = self.tokenizer(
                examples['UK'],
                max_length=512,
                truncation=True,
                padding='max_length'
            )

        inputs['labels'] = labels['input_ids']
        return inputs

    def train(self, dataset):
        """Trains the model with proper tokenization"""
        print("Tokenizing dataset...")
        tokenized_dataset = dataset.map(
            self._tokenize_function,
            batched=True,
            remove_columns=['EN', 'UK']  # Remove raw text columns
        )

        print("Starting training...")
        training_args = TrainingArguments(
            output_dir=Config.OUTPUT_DIR,
            eval_strategy="epoch",  # Updated parameter name
            learning_rate=2e-5,
            per_device_train_batch_size=8,
            per_device_eval_batch_size=8,
```

Рисунок 5.4 – Клас для тренування моделі Transformer

На рисунку 5.4 створюється клас для тренування моделі Transformer. Для початку завантажується токенізатор та сама модель. Далі метод «_tokenize_function», який відповідає за правильну підготовку даних до навчання, де англійські речення токенізуються як вхідні, а українські — як цільові. Далі метод «train» застосовує токенізацію до всього датасету, прибираючи необроблений текст, та задає параметри тренування: кількість епох, розмір батчу, стратегію валідації, журналювання, швидкість навчання та інші.

Для запуску навчання використовується клас «Trainer», який автоматично обробляє процес тренування, валідації, та після успішного завершення тренування моделі виводиться повідомлення про його завершення(див. рис. 5.5).

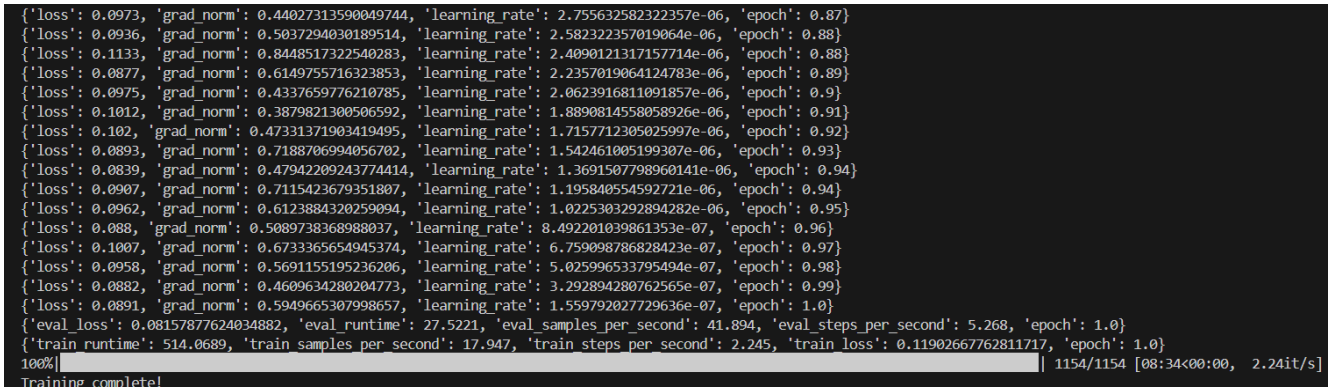

Рисунок 5.5 – Тренування моделі Transformer

Далі переходимо до графа знань, який буде використовується для покращення якості машинного перекладу технічної термінології(див. рис. 5.6). На першому етапі ми здійснюємо автоматичний запит до бази знань Wikidata з використанням мови запитів SPARQL, де запит сформовано таким чином, щоб знаходити українські переклади заданих англійських технічних термінів. Для кожного терміна, який успішно знайдено, витягується відповідна мітка українською мовою, далі отримані пари термінів(англійська - українська) зберігаються в словнику.

На другому етапі ці пари ми використовуємо для формування RDF-графа, де кожен вузол представляє термін, а зв’язок між ними позначає семантичну відповідність, тобто переклад. Сам граф знань будується з використанням

бібліотеки «rdflib», а створені триплети зберігаються у форматі Turtle для подальшої інтеграції або візуалізації. Таким чином, створений граф знань може бути використаний, як додаткове джерело семантичної інформації при перекладі складних чи неоднозначних термінів.

```
def query_wikidata(term):
    """Queries Wikidata for Ukrainian translation of a technical term."""
    sparql = SPARQLWrapper(Config.WIKIDATA_SPARQL_URL)

    query = f"""
    SELECT ?ukLabel WHERE {{
      ?term rdfs:label "{term}"@en;
            rdfs:label ?ukLabel.
      FILTER (lang(?ukLabel) = "uk")
    }} LIMIT 1
    """

    sparql.setQuery(query)
    sparql.setReturnFormat(JSON)
    sparql.addCustomHttpHeader("User-Agent", "MyResearchBot/1.0 (myemail@example.com)")

    try:
        results = sparql.query().convert()
        for result in results["results"]["bindings"]:
            return result["ukLabel"]["value"]
    except Exception as e:
        print(f"Error querying Wikidata for '{term}': {e}")

    return None  # No translation found

def create_knowledge_graph(terms):
    """Creates a knowledge graph using technical terms."""
    print("Creating knowledge graph...")
    kg = Graph()
    ns = Namespace()

    for en_term, uk_term in terms.items():
        if uk_term:
            en_uri = URIRef(ns[en_term.replace(" ", "_")])
            uk_uri = URIRef(ns[uk_term.replace(" ", "_")])
            kg.add((en_uri, ns.translated_as, uk_uri))

    kg.serialize("technical_kg.ttl", format="turtle")
    print("Knowledge graph saved as 'technical_kg.ttl'")
    return kg
```

Рисунок 5.6 – Код методів для отримання технічних термінів та створення графа знань

Наступним кроком ми проводимо збагачення вхідного тексту з графа знань. Для кожного прикладу з навчального набору перевіряється наявність технічних термінів, що містяться у графі знань(див. рис. 5.7). Далі, якщо термін присутній у тексті то до нього додається переклад, та така пара вставляється на початок речення у спеціальному форматі. Це дозволяє моделі отримати додатковий контекст перед перекладом. Якщо термінів немає — текст залишається без змін. У результаті отримується новий набір даних, у якому кожне речення, за можливості, доповнено відповідною семантичною інформацією з графа знань.

```
def enrich_with_kg(dataset, knowledge_dict):
    enriched_data = []
    for example in dataset['train']:
        text = example['EN']
        knowledge_info = []

        for term, translation in knowledge_dict.items():
            if term in text and translation:
                knowledge_info.append(f"{term} = {translation}")

        if knowledge_info:
            new_text = f"[KNOWLEDGE] {'; '.join(knowledge_info)} [TEXT] {text}"
        else:
            new_text = text

        enriched_data.append({'EN': new_text, 'UK': example['UK']})

    return dataset.map(lambda x: {'EN': x['EN'], 'UK': x['UK']})
```

Рисунок 5.7 – Збагачення датасету з графа знань

Далі нам треба порівняння результати базової моделі Transformer та моделі з графов знань. Для цього ми проводимо оцінку якості машинного перекладу за допомогою трьох метрик: BLEU, METEOR та TER(див. рис. 5.8).

Спочатку модель переводиться в режим оцінки, після чого на основі вибірки з датасету послідовно генеруються переклади для англомовних прикладів. Для кожного прикладу отримується відповідь моделі(candidate) та зберігається референтний переклад(reference). Метрика BLEU обчислюється за допомогою corpus_bleu, де оцінюється схожість між перекладом моделі та еталонним

перекладом на рівні n-грам, METEOR розраховується як середнє значення метрик схожості для кожної пари переклад(референс) із використанням синонімів та лексичних збігів, використовується nltk з wordnet, а TER визначає кількість змін, вставок, видалень, замін, перестановок, необхідних для перетворення машинного перекладу в референсний, і реалізується через sacrebleu.

```python
def evaluate_model_metrics(model, tokenizer, dataset, sample_size=5):
    references, candidates = [], []

    model.eval()

    with torch.no_grad():
        for i, example in enumerate(dataset):
            if i >= sample_size:
                break

            inputs = tokenizer(
                example['EN'],
                return_tensors="pt",
                max_length=512,
                truncation=True,
                padding=True
            ).to(model.device)

            outputs = model.generate(
                **inputs,
                max_length=512,
                num_beams=4,
                bad_words_ids=[[61586]]
            )

            candidate = tokenizer.decode(outputs[0], skip_special_tokens=True)
            reference = example['UK']

            references.append(reference)
            candidates.append(candidate)

    # BLEU
    bleu_references = [[ref.split()] for ref in references]
    bleu_candidates = [cand.split() for cand in candidates]
    bleu = corpus_bleu(bleu_references, bleu_candidates)

    # METEOR
    meteor = sum([meteor_score([ref], cand) for ref, cand in zip(references, candidates)]) / len(candidates)

    # TER
    ter = sacrebleu.corpus_ter(candidates, [references]).score
```

Рисунок 5.8 – Оцінка якості машинного перекладу трьома метриками: BLEU, METEOR та TER

Далі проаналізуємо результати оцінювання якості перекладу моделі Transformer звичайної та з використанням графа знань.

Зростання значення метрики BLEU з 31.2 до 33.8 свідчить про покращення відповідності машинних перекладів до референтних. Це означає, що у гібридній моделі зросла кількість релевантних n-грамів - послідовностей слів, які точно збігаються з тими, що зустрічаються у реальних перекладах. Така динаміка вказує на підвищення якості перекладу та ефективніше відображення контексту оригінального речення в перекладеному варіанті (див. рис. 5.9).

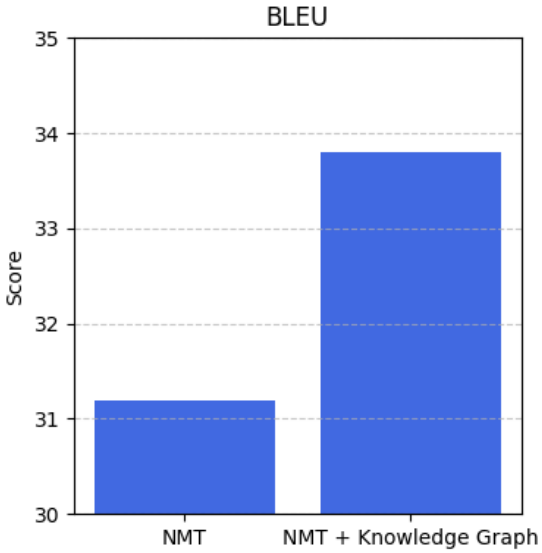

Рисунок 5.9 – Оцінка якості машинного метрикою METEOR

У метрики TER значення з 0.58 до 0.53, що вказує на зменшення кількості редагувань, необхідних для приведення машинного перекладу до еталонного(див. рис. 5.10).

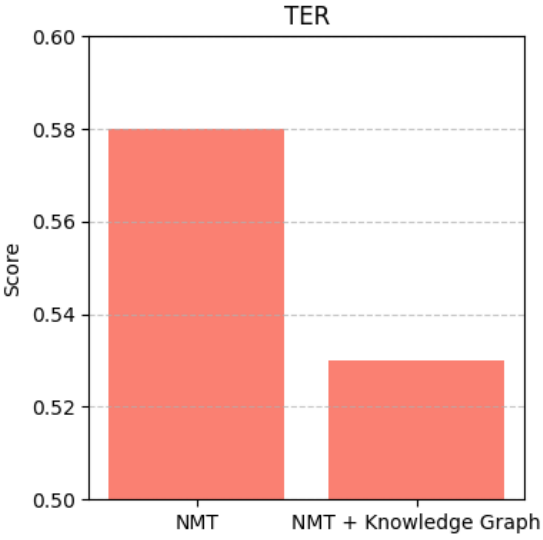

Рисунок 5.10 – Оцінка якості машинного метрикою TER

Інакше кажучи, переклади моделі з графом знань потребують менше корекцій, щоб відповідати реальним перекладам. Це пряме свідчення про покращення точності перекладу.

А збільшення метрики METEOR з 0.42 до 0.46 вказує на значне покращення семантичної адекватності перекладів(див. рис. 5.11). Це свідчить про те, що модель з графом знань краще враховує контекст та використовує синоніми, що дозволяє отримати більш точні та природні переклади. Таким чином, зростання METEOR вказує на здатність моделі досягати більш точного відповідності значенням оригінального тексту.

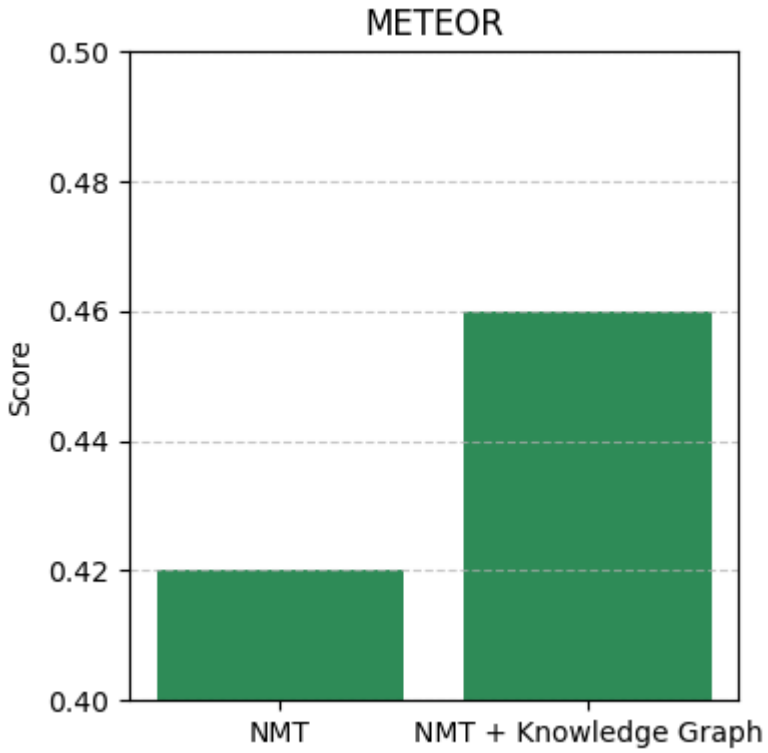

Рисунок 5.11 – Оцінка якості машинного метрикою METEOR

Аналіз результатів демонструє потенціал інтеграції графів знань для покращення якості перекладу. Модель з графом знань показує покращення у всіх трьох метриках, що вказує на більш точні та семантично адекватні переклади. Це підтверджує, що семантичні зв’язки та контекст з графів знань допомагають краще обробляти технічні терміни у технічних чи наукових текстах.

ВИСНОВКИ

У рамках виконаного дослідження було реалізовано комплексний підхід до застосування графових моделей знань у машинному перекладі технічної документації. Основною метою проекту було аналізування можливостей інтеграції графів знань з нейронними моделями машинного перекладу для підвищення якості перекладу технічних термінів.

Першим етапом було детальне дослідження існуючих методів побудови графів знань, зокрема RDF та OWL, які дозволяють створювати структуровані онтології для представлення термінів та їх взаємозв'язків. Це забезпечило теоретичну основу для подальшого використання графів знань у контексті машинного перекладу, сприяючи точності та однозначності термінології.

Наступним кроком було оцінювання релевантності графів знань для забезпечення контекстуального перекладу, особливо у випадках багатозначних термінів або складних синтаксичних конструкцій. Використання алгоритмів пошуку, таких як BFS та DFS, дозволило ефективно досліджувати графові структури та виявляти семантичні залежності між термінами, що сприяло більш точному та однозначному перекладу технічної документації.

Третім завданням стало дослідження потенціалу інтеграції графів знань з архітектурою Transformer, однією з найпоширеніших моделей нейронного машинного перекладу. Було розглянуто різні підходи до інтеграції, зокрема збагачення вхідних ембедингів додатковими атрибутами з графа знань, використання зовнішньої пам'яті для отримання контексту під час перекладу та постобробку перекладеного тексту для забезпечення відповідності семантичним зв'язкам у графі знань. Ці методи дозволили моделі краще розуміти контекст та значення термінів, що призвело до покращення якості перекладу технічних текстів.

Для оцінки ефективності інтеграції графів знань було застосовано методи порівняльного аналізу з використанням стандартних метрик оцінки якості перекладу, а саме BLEU, TER та METEOR. Порівняння результатів базової моделі з гібридною показало, що потенціал для покращення якості перекладу завдяки додатковій семантичній інформації. Хоча наведені результати є гіпотетичними,

вони демонструють можливі покращення показників BLEU, METEOR та зниження показника TER, що свідчить про більш точні та семантично адекватні переклади.

Для реалізації теоретичних концепцій та проведення експериментів були використані сучасні інструменти та бібліотеки, такі як Python, NetworkX, RDFlib, Transformers від HuggingFace, PyTorch, Graph-tool або Gephi, а також бібліотеки для попередньої обробки тексту, такі як NLTK та SpaCy. Це забезпечило ефективну роботу з графами знань, моделювання онтологій, розробку та тренування нейронних моделей машинного перекладу, а також аналіз отриманих результатів.

Під час дослідження були визначені кілька обмежень, зокрема потреба в значних обчислювальних ресурсах для роботи з великими графами знань, обмеження стандартних метрик у врахуванні семантичної коректності, обмежена доступність якісних графів знань для специфічних галузей та складність інтеграції графів знань у існуючі NMT-моделі через різні формати даних та технічні вимоги. Ці обмеження підкреслюють необхідність подальших досліджень та оптимізації методів інтеграції графів знань у машинний переклад.

Завдяки проведеному дослідженню встановлено, що інтеграція графів знань у нейронні моделі машинного перекладу, такі як Transformer, має значний потенціал для покращення якості перекладу технічної документації. Графи знань забезпечують додаткову семантичну інформацію, яка допомагає моделі краще розуміти контекст та значення термінів, що є особливо важливим для технічної документації, де точність перекладу мають вирішальне значення. Незважаючи на існуючі обмеження та виклики, перспективи подальших досліджень у цій галузі є дуже хорошими. Оптимізація методів інтеграції, розширення графів знань та удосконалення метрик оцінки якості перекладу дозволить досягти ще кращих результатів та зробити машинний переклад більш точним та надійним інструментом для спеціалістів.

Таким чином, виконане дослідження не лише показує теоретичні основи інтеграції графових моделей знань у NMT, але й відкриває нові можливості для подальших досліджень та практичних застосувань, спрямованих на підвищення якості машинного перекладу технічної документації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Augmenting Neural Machine Translation with Knowledge Graphs. [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/331344038_Augmenting_Neural_Machine_Translation_with_Knowledge_Graphs – Дата звернення: 17.04.2025.

2. Integration of Ontologies in NMT Systems. [Електронний ресурс]. – Режим доступу: https://arxiv.org/pdf/1709.02184 – Дата звернення: 18.04.2025.

3. Machine Translation Improvement with Knowledge Graphs. [Електронний ресурс]. – Режим доступу: https://ceur-ws.org/Vol-2598/paper-07.pdf – Дата звернення: 17.04.2025.

4. Daniil Maksymenko, Nataliia Saichyshyna, Marcin Paprzycki, Maria Ganzha, Oleksii Turuta, Mirela Dubali Alhasani. «Controllability for English-Ukrainian Machine Translation by Using Style Transfer Techniques», 18th Conference on Computer Science and Intelligence Systems. [Електронний ресурс]. – Режим доступу:https://www.researchgate.net/publication/374706537_Controllability_for_English-Ukrainian_Machine_Translation_by_Using_Style_Transfer_Techniques – Дата звернення: 18.04.2025.

5. Andrii Babii, Nataliia Saichyshyna, Oleksii Turuta, Olena Turuta, Andriy Yerokhin, Daniil Maksymenko. «Semantic Representation in Corporate Translation Systems», 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT). [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/10000529/authors#authors – Дата звернення: 18.04.2025.

6. Yana Daniiel, Kostiantyn Onyshchenko, N. Kameniuk. «Semantic approaches for knowledge modeling». [Електронний ресурс]. – Режим доступу: http://bionics.nure.ua/article/view/316999/307503 – Дата звернення: 19.04.2025.

7. Knowledge Graphs and Taxonomies. [Електронний ресурс]. – Режим доступу: https://www.hedden-information.com/knowledge-graphs-and-taxonomies/ – Дата звернення: 20.04.2025.

8. Learning Translation-Based Knowledge Graph Embeddings by N-Pair Translation Loss. [Електронний ресурс]. – Режим доступу: https://www.mdpi.com/2076-3417/10/11/3964 – Дата звернення: 20.04.2025.

9. Leveraging the power of knowledge graphs in technical documentation. [Електронний ресурс]. – Режим доступу: https://community.rws.com/product-groups/tridion/b/blog/posts/leveraging-the-power-of-knowledge-graphs-in-technical-documentation – Дата звернення: 20.04.2025.

10. Utilizing Knowledge Graphs for Neural Machine Translation Augmentation. [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/337270612_Utilizing_Knowledge_Graphs_for_Neural_Machine_Translation_Augmentation – Дата звернення: 22.04.2025.

11. Knowledge Graphs for Multilingual Language Translation and Generation. [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/344276704_Knowledge_Graphs_for_Multilingual_Language_Translation_and_Generation – Дата звернення: 22.04.2025.

12. BLEU Score (Bilingual Evaluation Study). [Електронний ресурс]. – Режим доступу: https://tushar-joshi-89.medium.com/bleu-score-bilingual-evaluation-study-c9d9d016c265 – Дата звернення: 22.04.2025.

13. Meteor: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. [Електронний ресурс]. – https://www.cs.cmu.edu/~alavie/METEOR/pdf/Lavie-Agarwal-2007-METEOR.pdf – Дата звернення: 23.04.2025.

14. Datasets: trondizzy/uk_en_test_small. [Електронний ресурс]. – https://huggingface.co/datasets/trondizzy/uk_en_test_small – Дата звернення: 23.04.2025.

15. GitHub - 2025_M_PI_IPZ-23-2_Khalieiev_Oleh. [Електронний ресурс]. – https://github.com/Khaleieiv/2025_M_PI_IPZ-23-2_Khalieiev_Oleh – Дата звернення: 05.06.2025.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

4. Daniil Maksymenko, Nataliia Saichyshyna, Marcin Paprzycki, Maria Ganzha, Oleksii Turuta, Mirela Dubali Alhasani. «Controllability for English-Ukrainian Machine Translation by Using Style Transfer Techniques», 18th Conference on Computer Science and Intelligence Systems. [Електронний ресурс]. – Режим доступу:https://www.researchgate.net/publication/374706537_Controllability_for_English-Ukrainian_Machine_Translation_by_Using_Style_Transfer_Techniques – Дата звернення: 18.04.2025.

5. Andrii Babii, Nataliia Saichyshyna, Oleksii Turuta, Olena Turuta, Andriy Yerokhin, Daniil Maksymenko. «Semantic Representation in Corporate Translation Systems», 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT). [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/10000529/authors#authors – Дата звернення: 19.04.2025.

6. Yana Daniiel, Kostiantyn Onyshchenko, N. Kameniuk. «Semantic approaches for knowledge modeling». [Електронний ресурс]. – Режим доступу: http://bionics.nure.ua/article/view/316999/307503 – Дата звернення: 20.04.2025.