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 обробки природної мови для сумаризації тексту наукових публікацій
(тема)

Виконав:
студент 2 курсу, групи СШІм-21-1
Морковський Кирило Дмитрович
(прізвище, ініціали)

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 спеціалізації)

Керівник доц. Узлов Д.Ю.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

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

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

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Морковському Кирилу Дмитровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів обробки природної мови для сумаризації тексту наукових публікацій.

затверджена наказом університету від 31 березня 20 23 р. № 306 Ст
2. Термін подання студентом роботи до екзаменаційної комісії 17 травня 20 23 р.
3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та відомих наукових проектів щодо розробки та дослідження методів сумаризації тексту, документація окремих готових рішень.

4. Перелік питань, що потрібно опрацювати в роботі Опис існуючих методів та засобів, Екстрактивна сумаризація, Представлення теми, Представлення індикатора, Абстрактна сумаризація, Методи на основі структури, Методи на основі семантики, Методи на основі глибокого навчання та нейронних мереж, Модель Трансформер, Методи оцінки результатів сумаризації, Датасети для оцінки методів автоматичної сумаризації, Оцінка та порівняння методів автоматичної сумаризації.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)_____

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Огляд і аналіз сучасного стану задачі	08.04.2023	виконано
2	Постановка задачі кваліфікаційної роботи	09.04.2023	виконано
3	Дослідження існуючих методів та засобів	16.04.2023	виконано
4	Оцінка та порівняння існуючих методів	24.04.2023	виконано
5	Оформлення пояснювальної записки	26.04.2023	виконано
6	Оформлення графічного матеріалу	27.04.2023	виконано
7	Розробка презентації	29.04.2023	виконано

Дата видачі завдання 3 квітня 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Узлов Д.Ю.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 70 с., 4 табл., 4 рис., 1 дод., 52 джерела.

ГЕНЕРАЦІЯ ТЕКСТУ, МАШИННЕ НАВЧАННЯ, ОБРОБКА ПРИРОДНОЇ МОВИ, ОБРОБКА ТЕКСТУ, РОЗУМІННЯ ПРИРОДНОЇ МОВИ, СУМАРИЗАЦІЯ, ТРАНСФОРМЕР, ШТУЧНИЙ ІНТЕЛЕКТ.

Об’єкт дослідження – сумаризація наукових публікацій.

Предмет дослідження – методи обробки природної мови для сумаризації тексту наукових публікацій.

Мета роботи – розібрати доступні існуючі методи та підходи обробки природної мови, за допомогою яких може бути здійснена сумаризація наукового тексту (публікації, статті, тощо) досить великого розміру. Провести їх порівняння в контексті теперішнього часу. Визначити найбільш оптимальний метод згідно з якістю вихідного тексту та рядом інших критеріїв.

Методи дослідження – огляд доступної літератури та інтернет ресурсів, оцінка роботи методів на реальних даних. Порівняння з точки зору прикладної ефективності.

Актуальність проблеми обумовлена тим, що у нас час інформаційного перевантаження важливо підтримувати актуальність інформації відносно читача, цінуючи його час та, яку функцію і виконують анотації. Кількість наукової літератури яка публікується кожен день ускладнює виявлення корисної інформації. Також частково актуальність обумовлена тим, що немає чітко визначеного кращого способу для сумаризації тексту великого обсягу.

Пояснювальна записка до кваліфікаційної роботи оформлена згідно ДСТУ 3008:2015 [1], перелік посилань оформлений згідно з ДСТУ 8302:2015 [2].

ABSTRACT

Explanatory note: 70 p., 4 fig., 4 tabl., 1 ann., 52 sources.

ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, NATURAL LANGUAGE PROCESSING, NATURAL LANGUAGE UNDERSTANDING, SUMMARIZATION, TEXT PROCESSING, TEXT GENERATION, TRANSFORMER.

Object of research – summarization of scientific publications.

The subject of research – Natural Language Processing methods for summarizing the text of scientific publications.

The purpose of the work is to analyze available existing methods and approaches of natural language processing, which can be used to summarize scientific text (publications, articles, etc.) of a sufficiently large size. Compare them in the context of the present time. Determine the most optimal method according to the quality of the source text and a number of other criteria.

Research methods – review of available literature and Internet resources, evaluation of methods on real data. Comparison in terms of applied efficiency.

The relevance of the problem is due to the fact that in our time of information overload it is important to maintain the relevance of information relative to the reader, valuing his time and the function that annotations perform. The amount of scientific literature that is published every day makes it difficult to find useful information. Also, the relevance is partly due to the fact that there is no clearly defined best way to summarize a large text.

The explanatory note to the qualification work is drawn up in accordance with DSTU 3008:2015 [1], the list of references is drawn up in accordance with DSTU 8302:2015 [2].

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

AMR – Abstract Meaning Representation, або Абстрактна Репрезентація Значення – це мова семантичного представлення. AMR має на меті абстрагування від синтаксичних представлень у тому сенсі, що схожим за значенням реченням слід призначати однакові AMR, навіть якщо вони не є ідентичними.

ATS – Automatic Text Summarization, або Автоматична Сумаризація Тексту – це процес автоматичного скорочення тексту обчислювальним шляхом для створення анотації, яка представляє найважливішу або релевантну інформацію в оригінальному тексті.

BERT – Bidirectional Encoder Representations from Transformers, або Двоспрямовані Кодувальні Представлення з Трансформерів – це сімейство моделей на базі трансформера, який використовує техніку маскування слів. Був представлений в 2018 році дослідниками Google.

Brill Tagger – це індуктивний метод позначення частини мови. Його можна коротко описати як «теггер на основі трансформації, керований помилками». Це форма навчання з наглядом, спрямована на мінімізацію помилок. І також це процес, заснований на трансформації, у тому сенсі, що тег призначається кожному слову та змінюється за допомогою набору попередньо визначених правил.

CNN – Convolutional neural network, або Згорткова нейронна мережа – це клас штучних нейронних мереж, які найчастіше застосовуються для аналізу візуальних зображень. CNN використовують математичну операцію, яка називається згорткою, замість загального множення матриць принаймні в одному зі своїх шарів.

Damping factor – Коефіцієнт демпфування – це ймовірність зловити «випадкового користувача» на веб-сторінці. Це дає можливість персоналізувати

результати та може зробити майже неможливим навмисне введення пошукової системи (у прикладі автора PageRank це Google) в оману для отримання вищого рейтингу сторінки. В контексті записки цей коефіцієнт використовується в LexRank.

DBS – Diverse Beam Search, або Пошук різноманітних променів – запропонований метод декодування вихідних даних моделей нейронних послідовностей, таких як RNN. Він має створювати послідовності, які значно відрізняються – з вимогами до часу виконання та пам’яті, які можна порівняти зі звичайним Променевим пошуком.

EMD – Earth Mover's Distance, чи Відстань землероба – є мірою відстані між двома розподілами ймовірностей в області D. У математиці ця міра відома як метрика Вассерштейна. Розподіли інтерпретуються як два різні способи нагромадження певної кількості землі в області D, EMD – це мінімальна вартість перетворення однієї купи в іншу; де вартість передбачається як кількість переміщеної землі, помножена на відстань, на яку вона переміщена.

FrameNet – це лексична база даних, яка надає семантичні кадри (та лексичні одиниці, які використовуються для визначення значення слова чи речення, класифікації речень та визначення релевантності між реченнями, а точність досягається за допомогою WordNet.

GPT-3 – Generative Pre-trained Transformer 3, або Генеративний попередньо навчений трансформер 3 – це авторегресійна мовна модель, яка була випущена у 2020 році. Ця модель є прямою еволюцією GPT-2, тепер із 175 мільярдами параметрів. Вона навчена на величезній кількості текстових даних (45 терабайт) та здатна виконувати широкий спектр завдань обробки природної мови, включаючи переклад мови, відповіді на запитання та завершення тексту.

GRU – Gated Recurrent Units, або Вентильні рекурентні вузли – це вентильний механізм в RNN, його застосовують для вирішення проблеми зникаючого градієнта в RNN.

Hierarchical clustering – Ієрархічна кластеризація – це метод кластерного аналізу, за допомогою якого будують ієрархію кластерів.

ILP – Integer Linear Programming, або Лінійне цілочисельне програмування – це завдання математичної оптимізації чи здійсненності, у якій деякі чи всі змінні мають бути цілими числами, а цільова функція та обмеження лінійні.

IR/IE – Information Extraction / Information Retrieval – Вилучення інформації/Пошук інформації. IE – це завдання автоматичного вилучення структурованої інформації з неструктурованих та/або напівструктурованих машинозчитуваних документів, тоді як IR – це пошук матеріалів неструктурованого характеру, які задовольняють інформаційні потреби у великих колекціях, за допомогою запитів.

JAMR – це семантичний аналізатор, генератор і вирівнювач для Абстрактної репрезентації значення, тобто для AMR-графів.

K-means – Кластеризація методом к–середніх – це метод векторного квантування, метою якого є розділення n спостережень на k кластерів, у яких кожне спостереження належить кластеру з найближчим середнім (центри кластера або центроїд кластера), що служить прототипом кластера.

LDA – Latent Dirichlet Allocation, або Латентне виділення Діріхле – це генеративна статистична модель, яка пояснює набір спостережень за допомогою неспостережуваних груп, і кожна група пояснює, чому деякі частини даних подібні. LDA є прикладом тематичної моделі.

Lexical unit – Лексична одиниця – у лексикографії це окреме слово, частина слова або ланцюжок слів, які утворюють основні елементи лексикону мови.

LexRank – це неконтрольований підхід на основі графів для ATS. Оцінка речень здійснюється за допомогою графів. Використовується для обчислення важливості речення на основі концепції центральності власного вектора в представленні речень на графі.

LSTM – Long Short-Term Memory, або Довга короткочасна пам'ять – це штучна нейронна мережа, яка використовується в галузі штучного інтелекту та глибокого навчання. На відміну від стандартних нейронних мереж прямого зв’язку, LSTM має зворотні зв’язки. Така рекурентна нейронна мережа (RNN) може обробляти не лише окремі точки даних, але й цілі послідовності даних (наприклад, мова чи відео).

MT – Machine Translation, або Машинний переклад – це підгалузь комп’ютерної лінгвістики, яка досліджує використання програмного забезпечення для перекладу тексту чи мови з однієї мови на іншу.

NLP – Natural Language Processing, або Обробка природної мови – це міждисциплінарний підрозділ лінгвістики, інформатики та штучного інтелекту, що займається взаємодією між комп’ютерами та людською мовою.

NLU – Natural Language Understanding, або Розуміння природної мови – це галузь штучного інтелекту, яка використовує комп’ютерне програмне забезпечення для розуміння введення у формі тексту чи мови.

PageRank – це алгоритм, який використовується Google для ранжування веб-сторінок у результатах пошукової системи.

PAS – Predicate-Argument Structure, або Предикатно-аргументна структура означає дієслова, підмет та об’єкт речення.

PENMAN – це формат серіалізації для спрямованих, кореневих графів, які використовуються для кодування семантичних залежностей, особливо в рамках AMR.

POS – Part of Speech, або Частина мови – у граматиці це категорія слів, які мають подібні граматичні властивості.

PropBank – це ресурс який містить тексти, анотовані інформацією про основні семантичні пропозиції.

PTLM – Pre-Trained Language Models, або Попередньо навчена мовна модель – це модель, яка спочатку була навчена на великому масиві текстових даних, щоб вивчити загальні мовні моделі та особливості. Потім модель

налаштовується на меншому наборі даних, щоб дізнатися про конкретні функції та закономірності, які стосуються поставленого завдання та предметної галузі.

ROUGE – Recall-Oriented Understudy for Gisting Evaluation, або Орієнтоване на Запам'ятовування Дослідження для Оцінки Суті – це набір показників, які використовуються для оцінки програмного забезпечення автоматичного підсумовування та машинного перекладу в обробці природної мови.

RSG – Rich Semantic Graphs, або Насичені семантичні графіки – це один із способів представлення семантичної інформації тексту у вигляді графіків. Іменники та дієслова тексту представляють вузли, а семантичне та топологічне відношення між ними відповідає краям. Іменники та дієслова тексту отримують за допомогою розбору або онтології. За потреби графік можна зменшити за допомогою евристичних правил, таких як заміна, видалення або комбінування вузлів.

Semantic frame – Семантичний фрейм (кадр) – це зв'язана структура понять, пов'язаних таким чином, що без знання всіх з них неможливо отримати повне знання жодного з них, і в цьому сенсі вони є типами гештальту.

Semantic role labeling – Маркування семантичної ролі – це процес, який призначає мітки словам або фразам у реченні, що вказує на їх семантичну роль у реченні, таку як агент, мета або результат.

Seq2seq – Sequence-to-sequence, або Послідовність до послідовності – це сімейство підходів машинного навчання, які використовуються для обробки природної мови. Програми включають переклад мови, підписи до зображень, розмовні моделі та підсумовування тексту.

Shallow Parsing – Chunking, Light parsing; або Поверхнево-синтаксичний аналіз – це аналіз речення, який спочатку ідентифікує складові частини речень (іменники, дієслова, прикметники тощо), а потім пов'язує їх з одиницями вищого порядку, які мають окремі граматичні значення (групи іменників або фрази, дієслова). групи тощо).

TF-IDF – Term frequency-inverse document frequency, або Частота терміну-зворотна частота документа – це показник із числової статистики, призначений для відображення важливості слова для документа в колекції або корпусі. Він часто використовується як ваговий коефіцієнт під час пошуку інформації, аналізу тексту та моделювання користувачів.

TG – Text Generation, або Генерація Тексту – це завдання генерування тексту який неможливо відрізнити від тексту, написаного людиною.

UMPG – Unsupervised Massive-scale Paraphrase Generation, або Неконтрольована генерація масових парафраз – підхід попереднього навчання, який генерує синтетичні парафрази уривків тексту та використовує їх для навчання моделі.

VSM – Vector Space Model, або Модель векторного простору – це алгебраїчна модель для представлення об’єктів як векторів ідентифікаторів.

WordNet – велика лексична база даних англійської мови. Іменники, дієслова, прикметники та прислівники згруповані в набори когнітивних синонімів, кожен з яких виражає окреме поняття. Вони пов'язані між собою за допомогою понятійно-семантичних та лексичних відношень.

ВСТУП

Направлення NLP (Natural Language Processing, або обробка природної мови) яке займається взаємодією між комп'ютерами та людською мовою, зокрема, як запрограмувати комп'ютери для обробки та аналізу великих обсягів даних природної мови. Має безліч варіантів прикладного використання, через це є однією з самих активних галузей машинного навчання. Великі компанії сприяють розвитку завдяки розповсюдженню різноманітних вже натренованих моделей, датасетів та документації до них.

Дуже популярної задачею NLP є сумаризація тексту, тобто автоматичне створення короткого змісту (заголовка, резюме, анотації) вихідного тексту. Наприклад такий підхід використовується в СМІ для створення заголовків статей, відео чи сюжетів для ТБ. Люди віддають перевагу коротким резюме з усіма важливими моментами, ніж читанню усієї статті та її конспектуванню самостійно. Це економить наш дорогоцінний час та показує, актуальна для нас ця тема чи ні. Реалізації цієї задачі пройшли довгий шлях від примітивних програм, які створюють резюме на основі вибраних з тексту речень (дуже часто без будь-якого редагування), до складних математичних функцій, RNN (Recurrent Neural Network, або Рекурентна нейронна мережа) рекурентних нейронних мереж, і врешті-решт до трансформерів.

Рекурентні нейронні мережі для задачі суммаризації тексту мали декілька важливих недоліків, тому згодом були цілком замінені на Трансформери.

Трансформер – це модель глибокого навчання, яка використовує механізм самоуваги, диференційовано зважуючи значимість кожної частини вхідних даних. Механізми уваги та самоуваги (Self-Attention) грають важливу роль у якості вихідного тексту. Трансформер став одним із найзначущих винаходів у NLP за останні шість років. Нещодавній вибух популярності GPT-3 привернув увагу ще більшої кількості людей до NLP та генерації тексту.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

Сумаризація (від англ. Summary – резюме) – це процес скорочення набору даних за допомогою обчислень для створення підмножини (анотації), яка представляє найважливішу або релевантну інформацію в оригінальному вмісті. Проблема, яка була актуальною з самого початку НЛП, це автоматичне вилучення анотації з великого корпусу тексту, або як її ще називають ATS.

ATS – Automatic Text Summarization, або Автоматична Сумаризація Тексту – це процес автоматичного скорочення тексту обчислювальним шляхом для створення анотації, яка представляє найважливішу або релевантну інформацію в оригінальному тексті. Анотація має бути якомога коротше, але також має охоплювати важливі елементи тексту. Це виявляється складним завданням для будь-якого алгоритму, оскільки існує фактично нескінченна кількість документів, які можуть існувати, і кожен із них може посилатися на унікальну концепцію. Комп’ютеру складно моделювати природну мову; відсутність або наявність окремого слова може змінити значення цілого речення або навіть цілого розділу.

Згідно до довжини тексту та його структури підхід до суммаризації може відрізнятися. В довгих текстах складніше виділяти важливі теми та уникати повторень. Тому виділяють декілька видів сумаризації за типом вхідних даних:

– суммаризація одного окремого документа;

– суммаризація декількох документів (multi-document summarization);

– суммаризація за запитом;

– інформаційна суммаризація: містить короткий виклад повної фактичної інформації, тобто без особистих поглядів та думок автора тексту.

Також сюди можно додати так звану Непрофесійну суммаризацію (Lay Summarization), що є коротким викладом дослідницького проекту або научного тексту, який написаний для представників громадськості, а не для дослідників або професіоналів. Тобто уникаючи складні терміни та професійний жаргон. Це

представляє досить складну задачу, тому як програмі необхідно мати увесь контекст проблеми, терміни, стійкі вирази, тощо. Уці елементи мови необхідно зрозуміти та перевести у досить просту форму мовлення. Програма має грамотно скласти текст у читабельному вигляді.

Згідно методів при вилученні тексту суммаризацію поділяють на декілька основних підходів:

– сумаризація на основі вилучення;
– сумаризація на абстрактній основі;
– сумаризація з допомогою.

При сумаризації на основі вилучення вміст витягується з вихідних даних, але речення тексту майже ніколи не редагуються чи змінюються. Приклади вилученого вмісту включають ключові фрази, які можна використовувати для «тегування» або індексування текстового документа, або ключові речення (включно з заголовками), які разом складають анотацію. Процес сумаризації тексту схож з процесом поверхневого читання, коли у першу чергу читається: зміст, заголовки та підзаголовки, перший та останній абзаци розділу; а вже потім робиться висновок чи читати детально увесь документ. Інші приклади виділення, які включають ключові послідовності тексту з точки зору клінічної значущості, наприклад: пацієнт, проблема, втручання та результат.

Сумаризація на абстрактній основі генерує новий текст, якого не було в оригінальному тексті. Абстрактні методи створюють внутрішнє семантичне представлення оригінального вмісту, що називають мовною моделлю, а потім використовують це представлення для створення резюме, яке ближче до того, що може висловити людина. Абстракція може трансформувати витягнутий вміст шляхом перефразування розділів вихідного документа, щоб ущільнити текст сильніше, ніж просте вилучення речень. Однак така трансформація набагато складніша з обчислювальної точки зору, залучаючи як обробку природної мови, так і часто глибоке розуміння домену оригінального тексту у випадках, коли оригінальний документ відноситься до спеціальної галузі знань.

Визначення теми, інтерпретація, створення резюме та оцінка створеного резюме є ключовими проблемами під час конспектування тексту. Для оцінки створеного резюме необхідно не тільки розуміти сенс тексту, а й трохи розбиратися у предметній галузі, щоб визначати коли резюме дійсно створене на основі тексту, а не лише виглядає ніби було створене на його основі. Це стає серйозною проблемою коли завдання включає тексти різних областей знань.

Існує два типи оцінювання результуючих анотацій:

– людське оцінювання;

– автоматичне оцінювання.

При людському оцінюванні бали призначаються експертами-людьми на основі того, наскільки добре резюме охоплює пункти тексту, відповідає на запити, уникає повторень інформації, відповідає граматичним та стилістичним критеріям.

Автоматичне оцінюванне припускає використання програмного забеспечення для оцінювання анотацій. Найбільш поширеною та відомою вважається ROUGE (Recall-Oriented Understudy for Gisting Evaluation), або Орієнтоване на Запам’ятовування Дослідження для Оцінки Суті – це набір показників, які використовуються для оцінки програмного забезпечення автоматичного підсумовування та машинного перекладу в обробці природної мови. Незважаючи на широке розповсюдження, показник ROUGE не є досконалим, але кращого способу оцінки автоматизованих анотацій ще не знайдено.

2 ОПИС ІСНУЮЧИХ МЕТОДІВ ТА ЗАСОБІВ

Завдання щодо узагальнення корпусу тексту завжди є цікавим через виклики, які воно створює. Визначення теми тексту, ранжування речень за важливістю, перефразування, використання ідіом тощо.

Усі ці завдання відносяться до NLU. Natural Language Understanding, або розуміння природної мови – це окремий технологічний напрямок в частині штучного інтелекту, який присвячений тому, як машина розуміє природну мову, якою людина звертається до неї. На зараз існує значний комерційний інтерес до цієї галузі через її застосування для автоматизованого обґрунтування, машинного перекладу, питально-відповідних систем, збору новин, категоризації тексту, голосової активації, архівування та широкомасштабного аналізу контенту.

Але просто розуміння тексту є однією частиною проблеми, а інша полягає в створенні текстової відповіді гідної якості. Це завдання TG – Text Generation, або Генерація Тексту – це завдання генерування тексту який неможливо відрізнити від тексту, написаного людиною.

У цьому розділі ми збираємося глибоко зануритися в методи, підходи та техніки резюмування тексту. Деякі з них, можливо, застаріли, але вони дають необхідний контекст для деяких сучасних проблем та рішень у цій галузі. В даний час усі методи реферування тексту поділяються на дві групи: екстрактивні та абстрактні.

2.1 Екстрактивна сумаризація

Методи екстракційної сумаризації створюють резюме шляхом вибору підмножини речень в оригінальному тексті. Екстрактивні підсумовувачі спочатку створюють проміжне представлення, яке має основне завдання виділити або вилучити найважливішу інформацію з тексту, який потрібно

підсумувати на основі представлень. Існує два основних типи представлень: представлення теми та індикатора.

2.1.1 Представлення теми

Представлення теми зосереджено на представленні тем із тексту. Існує кілька підходів до отримання цього представлення. До них належать: Частотні підходи, Підходи тематичних слів та Латентно-семантичний аналіз.

Частотні підходи походять від призначення вагових коефіцієнтів словам. Якщо слово пов'язане з темою, ми присвоюємо 1, у іншому випадку 0. Ваги можуть бути безперервними залежно від реалізації.

Імовірність слова (Word Probability) [3] це найпростіша форма використання частоти у вхідних даних як показника важливості та вираховується за формулою (2.1).

$$P(w) = \frac{f(w)}{N}, \tag{2.1}$$

де *f(w)* – це частота появи слова у тексті;

N – це кількість слів у вихідному тексті.

Існує альтернатива ймовірності слова, яка називається необробленою частотою (Raw Frequency), але на неї сильно впливає довжина документа. Обчислення частоти слова вносить поправки на довжину документа. Оскільки методи ймовірності слів залежать від списку стоп-слів, існує потреба в більш досконалих техніках.

Частота терміну-зворотна частота документа (Term frequency-inverse document frequency, TF-IDF) [3]. Цей метод був розроблений як удосконалення методу ймовірності слова. Метод призначає низьку вагу словам, які дуже часто зустрічаються в більшості документів, припускаючи що вони є стоп-словами або такими словами, як «The». В іншому випадку якщо слово з'являється в

документі унікально з високою частотою, йому надається висока вага. TF-IDF є добутком двох статистичних даних: частоти термінів та зворотної частоти документа. Існують різні способи визначення точних значень обох статистичних даних. Інший, згаданому раніше, варіант визначення ймовірності слова (2.2).

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}, \tag{2.2}$$

де $f_{t,d}$ – це необроблена кількість термінів у документі, тобто кількість разів, коли термін *t* зустрічається в документі *d*;

$\sum_{t' \in d} f_{t',d}$ – це загальна кількість термінів у документі d (підраховуючи кожне входження того самого терміну окремо).

Зворотна частота документа – це міра того, скільки інформації надає слово, тобто чи є воно поширеним чи рідкісним у всіх документах. Це логарифмічно масштабована обернена частка документів, які містять слово (2.3). Ми додаємо одиницю до знаменника, щоб виключити можливість ділення на нуль, яке може статися, якщо терміна немає в корпусі тексту.

$$idf(t,D) = log\,\frac{N}{1+|\{d \in D : t \in d\}|}, \tag{2.3}$$

де N – загальна кількість документів у корпусі N = | D |;

$1 + |\{d \in D : t \in d\}|$ – кількість документів, де з'являється термін *t*.

Тоді TF–IDF розраховується, як показано у формулі 2.4, як комбінація частоти слів (2.2) та зворотної частоти документа (2.3).

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D). \tag{2.4}$$

Оскільки відношення в логарифмічній функції *idf* завжди більше або дорівнює 1, тоді значення *idf* (та *tf–idf*) більше або дорівнює 0. Якщо термін

з’являється в більшій кількості документів, відношення в логарифмі наближається до 1, наближаючи *idf* та *tf–idf* до 0. Вагові коефіцієнти TF-IDF легко та швидко обчислити, а також є хорошими показниками для визначення важливості речень, і тому широко використовуються.

Підходи тематичних слів – вид підходів до представлення теми, метою яких є визначення слів, що описують тему вхідного документа. Такі методи обчислюють частоти слів та використовують порогове значення частоти, щоб знайти слово, яке потенційно може описати тему.

Для представлення теми може використовувався так званий метод Луна[3], який використовує частотні пороги для ідентифікації описових слів у документі. Описові слова в цьому підході виключають слова, які найчастіше зустрічаються в документі, які ймовірно є детермінантами, прийменниками або специфічними словами, а також такими, що зустрічаються лише кілька разів.

Сучасна статистична версія ідеї Луна застосовує тест співвідношення правдоподібності для визначення слів, які добре описують вхідні дані. Такі слова традиційно називаються «тематичними підписами». Використання тематичними підписів як представлення вхідних даних призвело до високої продуктивності у виборі важливого вмісту для сумаризації мульті-документних текстів.

Тематичні підписи (Topic signatures) [3] – це слова, які часто зустрічаються у вхідних даних, але рідко зустрічаються в інших текстах, тому їх обчислення вимагає підрахунку з великої колекції документів на додаток до вхідних даних для підсумовування. Однією з ключових сильних сторін підходу до перевірки логарифмічного відношення правдоподібності є те, що він забезпечує спосіб встановлення порогу для розділення всіх слів у вхідних даних на описові чи ні. Рішення приймається на основі тесту на статистичну значущість, що значною мірою усуває потребу в довільних порогових значеннях у оригінальному підході. Інформація про частоту появи слів у

великому фоновому корпусі необхідна для обчислення статистики, на основі якої визначаються тематичні підписи.

Імовірність вхідного *I* та фонового корпусу обчислюється за двома припущеннями:

– ймовірність слова у вхідних даних така ж, як у фоновому корпусі *B* – P(I) = P(B) = p;

– слово має іншу, вищу ймовірність, на вході, ніж у фоновому корпусі B – P(I) = p_I та P(B) = p_B та $p_I > p_B$.

Імовірність тексту щодо певного слова, яке нас цікавить, *w*, обчислюється за допомогою формули біноміального розподілу. Вхідний та фоновий корпуси розглядаються як послідовність слів. Поява кожного слова є схемою Бернуллі з ймовірністю *p* успіху, яка відбувається, коли $w_i = w$. Загальна ймовірність того, що слово *w* з'явиться *k* разів у *N* випробуваннях, визначається біноміальним розподілом (2.5).

$$b(N,k,p) = \left(\frac{N}{k}\right) p^k (1-p)^{N-k}. \quad (2.5)$$

Для першого припущення ймовірність *p* обчислюється разом із вхідними даними та фоновим корпусом. Для другого припущення p_1 обчислюється з вхідних даних, p_2 з фонового корпусу, а ймовірність усіх даних дорівнює добутку бінома для вхідних даних та для фонового корпусу. Більш конкретно, коефіцієнт правдоподібності визначається як (2.6).

$$\lambda = \frac{b(k,N,p)}{b(k_I,N_I,p_I)\cdot b(k_B,N_B,p_B)}, \quad (2.6)$$

де підрахунки з індексом *I* – обчислюються лише від вхідних даних до підсумовувача; підрахунки з індексом *B* – обчислюються над фоновим корпусом.

Статистика, яка дорівнює $-2 \log \lambda$, має відомий статистичний розподіл X^2, за допомогою якого можна визначити, які слова є тематичними підписами. Тематичні підписи це ті слова, які мають статистику ймовірності, більшу, ніж те, що можна було б очікувати випадково. Ймовірність випадкового отримання заданого значення можна знайти в таблиці розподілу X^2; наприклад: значення 10,83 можна отримати випадково з імовірністю 0,001.

Важливість речення обчислюється як кількість тематичних підписів, які воно містить, або як частка підписів у реченні. Обидві ці функції оцінювання речень базуються на тому самому представленні теми, бали, які вони призначають реченням, можуть бути досить різними. Підхід Луна, ймовірно, дасть вищу оцінку довшим реченням просто тому, що вони містять більше слів. Підхід з використанням перевірки відношенням правдоподібностей надає перевагу щільності тематичних слів.

Латентне виділення Діріхле (Latent Dirichlet allocation, LDA) [5] – це генеративна статистична модель, яка висуває гіпотезу про існування основного розподілу слів, тем та документів, які створили колекцію вхідного тексту. Використовуючи жаргон імовірнісної тематичної моделі, слова документа називаються «спостережуваними змінними», тоді як змінні структури теми називаються «прихованими змінними». Використовуючи ітераційний процес, модель оцінює апостеріорний розподіл прихованих змінних, враховуючи спостережувані змінні. Іншими словами, ця модель може виводити теми, спостерігаючи за розподілом слів у тексті. Однак величезна кількість тематичних структур, які можуть існувати, призводить до експоненціальної складності обчислень.

Вибірка Гіббса (Gibbs Sampling) [5] – у цьому методі будується ланцюг Маркова, що є послідовністю випадкових величин, кожна з яких залежить лише від попередньої, використовуючи вибірки з розподілу прихованих змінних. Присвоєння слів темам виконується ітераційно, доки ланцюг Маркова не зійдеться до цільового розподілу. На початку цієї процедури кожне слово

випадковим чином призначається темі, а в кожній наступній ітерації призначення слова-теми переоцінюються, що може призвести до проходження слів через кілька тем під час процесу. Вибірка Гіббса вирішує проблему експоненціальної складності обчислень у LDA.

Модель векторного простору (Vector Space Model, VSM) [5] передбачає проектування вхідних даних у n-вимірне векторне представлення, де семантична подібність точок визначається їхньою відстанню у векторному просторі, що був спроектований. Представлення векторів ознак широко використовуються в завданнях машинного навчання, напр. для класифікації, кластеризації тощо колекції вхідних елементів.

Сумка слів (Bag-of-words) [5] – це популярний спосіб представлення набору документів як векторів ознак, де речення можна представити як вектор ознак слів. Кожна векторна координата виражає статистику слова, наприклад, як згаданий раніше TF-IDF, значення даного слова у вихідних текстах. Завдяки зіставленню слова з його значенням TF-IDF слова отримують високу вагу, коли вони часто з’являються в документі, на який посилаються, але рідко в інших документах набору. Перевага цього підходу полягає в тому, що він пригнічує загальні слова, які зустрічаються в більшості документів, не містять жодної семантичної цінності для завдання. Було продемонстровано, що підхід може призвести до значних покращень у порівнянні з підходами з необробленими частотами в різноманітних завданнях пошуку інформації.

2.1.2 Представлення індикатора

Представлення індикатора (Indicator Representation) описує кожне речення як список важливих формальних характеристик (індикаторів), таких як довжина речення, позиція в документі або наявність певних фраз. Отже, тут важливість речення залежить не від слів, які воно містить, як ми бачили в представленнях теми, а безпосередньо від особливостей речення. Прикладами

представлень індикаторів є моделі на основі графів і моделі машинного навчання.

Методи на основі графа (Graph-based methods) являють собою один із підходів до представлення індикаторів. Ці методи використовують графи на основі речень для представлення документа або кластера документів.

LexRank – це підхід до підсумовування тексту, заснований на оцінці центральності речень на основі графіка [6]. Основна ідея полягає в тому, що речення «рекомендують» читачеві інші подібні речення. Таким чином, якщо одне речення дуже схоже на багато інших, воно, ймовірно, буде дуже важливим. Важливість цього речення також випливає з важливості речень, які його «рекомендують». Таким чином, щоб отримати високу оцінку та розмістити в резюме, речення має бути схоже на багато речень, які, у свою чергу, також схожі на багато інших речень. Це має інтуїтивно зрозумілий сенс та дозволяє застосовувати алгоритми до будь-якого довільного нового тексту.

У своїй статті автори LexRank обговорюють, як випадкові блукання по графах на основі речень можуть допомогти у підсумовуванні тексту. Наприклад, етапи підрахунку речень LexRank включають:

– представлення речень документа за допомогою неорієнтованого графа таким чином, що кожен вузол у графі представляє речення з вхідного тексту, а для кожної пари речень вага сполучного краю є семантичною подібністю між двома відповідними реченнями за допомогою косинусної подібності;

– використовуючи алгоритм ранжування, визначаються показник важливості для кожного речення.

LexRank включає формулу PageRank (2.7) [6], яка вперше була запропонована для обчислення престижу веб-сторінки та донині служить основним механізмом пошукової системи Google. Єдина відмінність від реалізації PageRank це те, що графік LexRank є неорієнтованим.

$$p(u) = \frac{d}{N} + (1 - d) \sum_{v \in adj[u]} \frac{p(v)}{degdeg\,(v)}. \quad (2.7)$$

де N – загальна кількість вузлів на графі;

d – «коефіцієнт демпфування» (damping factor), який зазвичай вибирається в інтервалі [0.1, 0.2].

Алгоритм із лістингу 2.1 підсумовує, як обчислити оцінки LexRank для заданого набору речень. Показники центральності ступеня також обчислюються (в масиві ступеня) як побічний продукт алгоритму.

Лістинг 2.1 – Алгоритм обчислення балів LexRank [6].

```
Input: An Array S of n sentences, cosine threshold t.
Output: An array L of LexRank.
Array CosineMatrixp[n][n];
Array Degree[n];
Array L[n];
for i ← 1 to n do
    for j ← 1 to n do
    CosineMatrix[i][j] = idf-modified-cosine(S[i], S[j]);
    if CosineMatrix[i][j] > t then
    CosineMatrix[i][j] = 1;
    Degree[i]++;
    end
    else
    CosineMatrix[i][j] = 0;
    end
    end
end
for i ← 1 to n do
    for j ← 1 to n do
    CosineMatrix[i][j]= CosineMatrix[i][j] / Degree[i];
    end
end
L = PowerMethod(CosineMatrix, n ,e);
return L;
```

Екстрактивна сумаризація працює шляхом вибору підмножини речень в оригінальних документах. Процес екстрактивної сумаризації можна розглядати як ідентифікацію найбільш центральних речень у кластері, які надають необхідну та достатню кількість інформації, пов’язаної з основною темою кластера. Центральність речення часто визначається з точки зору центральності слів, які воно містить. Поширеним способом оцінки центральності слова є перегляд центроїда кластера документа у векторному просторі. Центроїд кластера – це псевдодокумент, який складається зі слів, які мають показники tf-idf вище попередньо визначеного порогу, де tf – частота слова в кластері, а значення idf зазвичай обчислюються для набагато більшого датасету подібного жанру. У резюмуванні на основі центроїда речення, які містять більше слів із центроїда кластера, вважаються центральними. Це показник того, наскільки близько речення до центроїда кластера. Резюмування на основі центроїдів дало багатообіцяючі результати в минулому та посприяло створенню першої багатодокументної веб-системи реферування.

Можна виділити наступні ознаки речень, які важливі при процесі вибору важливих речень:

– довжина речення;

– позиція у вхідному документі – номер речення у тексті нормований за шкалою від 0 до 1;

– наявність дієслова – ця ознака базується на припущенні, що повне речення завжди містить дієслово;

– зворотні займенники – оцінка речення здійснюється за словами, присутніми в реченні. Під час цього процесу більшість систем спеціально нехтують стоп-словами, які зустрічаються в документі. Зворотними займенниками також нехтують як і стоп-словами. Але щоб отримати фактичну оцінку речення, слід також розглянути власні іменники, до яких ці займенники відносяться.

Також деякі показники описуються на рівні слів, такі як:

– періодичність терміну, про який вже йшлося раніше;

– довжина слова – менші слова зустрічаються частіше, ніж великі слова. Щоб звести нанівець цей ефект, довжина слова вважається ознакою;

– тег частини мови – теггер Брилл (Brill tagger) використовується для пошуку тегу частини мови слова. Теги ранжуються та призначаються ваги на основі інформації, яку вони вносять у речення;

– знайомість слова – знайомість, яка отримується від лексичної бази даних WordNet, вказує на те, наскільки загальним є слово в усіх документах. Це також свідчить про багатозначність слова. Менш знайомі слова отримують більшу вагу. Сигмоїдна функція використовується для обчислення ваги слова. Вага слова визначається як (2.8);

$$f(w) = \frac{1}{1 - e^{-8(\frac{1}{fam} - 0.5)}}. \quad (2.8)$$

– тег іменованої сутності (Name Entity) – Теги іменної сутності (чи об'єкти реального світу) ранжуються залежно від частоти їх появи;

– поява в заголовках або підзаголовках – слова, які зустрічаються в заголовках і підзаголовках, розглядаються як важливі та мають більшу вагу порівняно з іншими словами;

– стиль шрифту – шрифт, яким написано слово, також зберігається як функція. Також має значення, чи написано слово верхнім регістром, регістром заголовка чи нижнім регістром. Перевага словам віддається в тому ж порядку.

2.2 Абстрактна сумаризація

Абстрактний підхід представляє вхідний текст у проміжній формі, а потім генерує резюме зі словами та реченнями, які відрізняються від речень оригінального тексту. Складність обробки природної мови робить абстрактне

узагальнення складним завданням. Але зараз дослідження екстрактивних підсумків знаходяться в застої, оскільки вони досягли найвищої продуктивності, а отже увага більше зміщена в бік абстрактного резюмування та злиття екстрактивних та абстрактних технік.

Екстрактивна сумаризація можливо виграє у швидкості, але якість результатів могла б бути кращою. Справа в тому, що час не має значення, є потреба у високоякісних резюме, які є граматично правильними, читабельними, зв’язними, стислими та насиченими інформацією. Якість резюме компенсує витрачений час. Абстрактна сумаризація допомагає вирішити проблему висячої анафори і, таким чином, допомагає створити читабельні, стислі та зв’язні резюме.

Абстрактна сумаризація допомагає зменшити розмір речення, оскільки вона використовує злиття для об’єднання речень, таким чином, допомагає зменшити кількість повторів у резюме порівняно з екстрагованим резюме, де навіть нерелевантна частина речення також включається через те, що воно виділяє речення та впорядковує їх.

Експеримент із резюмування кількох документів для порівняння написаних людиною резюме, показав, що продуктивність чистих екстрактивних резюме дуже низька порівняно з абстрактними резюме. Останнім часом зростає інтерес до використання як екстрактивного, так і абстрактного типів резюме шляхом їх поєднання разом для підвищення якості кінцевого абстрактного резюме.

2.2.1 Методи на основі структури

Підходи на основі структури (Structure Based Methods), знаходять найважливішу інформацію з тексту, а потім використовують шаблони, правила, дерева, онтологію тощо для створення абстрактних резюме. Вони здебільшого

використовуються разом із підходами, заснованими на екстрактивному, семантичному або глибокому навчанні.

Мало хто з дослідників використовував шаблони разом із підходом на основі графів; деякі використовували правила та онтологію разом із семантичними графами; деякі використовували шаблони разом із семантичним підходом, тощо. Іноді ці підходи також використовуються як перший крок для попередньої обробки тексту, наприклад, деякі виділяють важливі ключові фрази з тексту за допомогою онтології, яку потім можна комбінувати з деякими іншими підходами для створення абстрактного резюме.

Методи на основі дерева (Tree Based Methods) – у цьому підході спочатку виділяється важливий текст, який потрібно розглянути для резюме. Потім подібні речення ідентифікуються з цього тексту за допомогою поверхнево-синтаксичного аналізу (Shallow Parsing). Подібні речення цього тексту потім заповнюються в деревоподібну структуру. Обробка дерев виконується шляхом лінеаризації (перетворення дерев на рядки), пошуку структури предикат-аргумент або об’єднання речень для створення остаточних абстрактних резюме.

Існують різні алгоритми, які допомагають нам виконувати це. Наприклад, алгоритм перетину теми вмісту (Content theme intersection) використовується для визначення загальних фраз за допомогою структури предикатів-аргументів, пошук базового дерева шляхом знаходження центроїда дерева залежностей, а потім збільшення цих базових дерев для отримання піддерев також є іншим способом щоб дати оцінку речення.

Дерево залежностей є найпопулярнішою структурою даних, яка використовується для представлення тексту у вигляді дерева. У 2005 році для створення інформативних резюме було застосовано генерацію «текст в текст». Цей метод походить від представлення речень за допомогою дерев залежностей для пошуку спільної інформації серед речень шляхом обробки дерев. Для отримання фінального речення метод перетинає перетини піддерев. Одним із

обмежень цього підходу є те, що він не може охопити зв’язки між реченнями, не знайшовши фразу, що перетинається між реченнями.

Один з таких методів це CoLIN [7], заснований на скороченні та лінеаризації дерева залежностей із збереженням семантичної інформації, інформаційного вмісту, граматичної правильності та зв’язності в резюме.

Інший метод використовує часткові дерева залежностей. Дерева залежностей будуються з синтаксичних дерев залежностей шляхом аналізу тексту. Під час аналізу рекомбінація та перехід виконуються на основі синтаксичної лінеаризації для створення багатодокументних абстрактних резюме. Дерева залежностей будуються за допомогою парсерів. Таким чином, продуктивність цих методів значною мірою залежить від доступних парсерів, що обмежує їх ефективність. Також вони більше зосереджуються на синтаксисі, ніж на семантиці.

Методи на основі шаблонів (Template Based Methods) – фрагменти тексту витягуються за допомогою ключових слів або підказок. Витягнуті фрагменти заповнюються в шаблони для формування остаточної анотації. Оскільки структура заздалегідь визначена, це допомагає створювати стислі та послідовні резюме. Вони спираються на глибокий синтаксичний та семантичний аналіз тексту. Цей метод добре працює, коли текст певним чином структурований. Але, оскільки правила та шаблони визначаються вручну, цей метод займає дуже багато часу, а також вимагає багато ручних зусиль.

Злиття кількох речень для створення абстрактних шаблонів вважається одним з таких методів. Для створення шаблонів використовуються фрази іменників разом із тегами POS (Part of Speech, або частини мови) та гіпернімами. Шаблони групуються та з них вилучаються кореневі дієслова. Шаблони зливаються разом за допомогою словарного графу (Word-graph).

Методи на основі онтології (Ontology Based Methods). Онтологію можна розглядати як сукупність сутностей та зв’язків між ними. Онтологія разом із набором окремих екземплярів класу становить базу знань. Класи є

найважливішою частиною онтології та представляють поняття. Онтологія визначає набір словників, а також містить узгодження термінів, що допомагає усунути будь-яку неоднозначність під час пошуку понять. Онтології, як правило, створюються експертами предметної області і здебільшого використовуються для вилучення понять та зв'язків із тексту. Потім ці концепції використовуються для створення резюме. Одним із прикладів предметно-орієнтованої онтології є WordNet, яка вже була частично згадана раніше. Онтології допомагають ділитися спільними знаннями між спільнотою інтересів. Вони допомагають відокремити предметну область і оперативні знання і, таким чином, допомагають легко вносити зміни, коли щось, пов'язане з доменом, змінюється. Вони дозволяють повторно використовувати знання. Онтології використовувалися в багатьох галузях, таких як електронне навчання, аналіз контенту, пов'язаного з користувачем, і аналіз зображень. Абстрактні системи на основі онтологій витягують інформацію з онтологій для створення анотації, що відповідає потребам користувача.

Часто словниковий запас в Інтернеті обмежений, онтологія допомагає правильно представити документ. Навіть семантичне представлення інформаційного вмісту можна значно покращити за допомогою онтології. Онтології допомагають досягти повноти теми та відсутності надмірностей у анотації.

Один із варіантів реалізації використовує спеціальний класифікатор, який приводить речення до формату визначеної онтології. Цей метод можна використовувати при конспектуванні статей новин. Онтологія допомагає фіксувати семантичну інформацію понять, щоб полегшити сумаризацію в системі.

Методи головної фрази та основної фрази (Lead and Body Phrase Methods) [7] – вибирається головне речення з тексту, потім до нього додається «важлива» відсутня інформація за допомогою вставки та заміни. Ці фрагменти важливої інформації називаються тригерами. Створення граматично правильних речень

за допомогою цього методу все ще є проблемою. Загалом головне речення змінюється шляхом додавання до нього інформації на зразок «хто», «де», «коли», «як» тощо.

Методи на основі графіків (Graph-Based Methods) – у цих методах для представлення тексту використовується словарний граф. Ці методи засновані на припущенні, що в тексті буде багато схожих речень, і ця подібність допоможе скомбінувати речення. Але не обов'язково, що подібні речення буде легко знайти. Використовуючи найкоротший шлях у графі слів можна отримати дуже граматично правильні та інформативні підсумки. Але оскільки речення ранжуються на основі інформативності, цьому методу бракує лінгвістичної якості. Одна з модифікацій цього підходу використовує, вже згаданий раніше, WordNet для відображення слів на графіку. Вміст інформації та граматична плавність вибираються як фактори для визначення найкращих шляхів графа, а потім використовуються узагальнення та агрегація для створення абстрактного резюме.

Альтернативний підхід до багатодокументної абстрактної сумаризації за допомогою графіків. Перш за все документи ранжуються за допомогою вищезгаданого LexRank. Потім подібні речення з важливих документів групуються разом. Найкоротші шляхи отримують між кластерами за допомогою словарного графа. І, нарешті, застосовується модель цілочисельного лінійного програмування (Integer Linear Programming, ILP), щоб знайти речення з максимальною інформативністю та читабельністю. ILP допомагає звести до мінімуму надмірність у резюме.

Методи на основі правил (Rule-Based Methods). Правила та категорії вводяться в систему для пошуку значущих кандидатів, які потім використовуються для створення резюме. Тут текстовий документ спочатку класифікується відповідно до термінів і понять, присутніх у ньому. Потім формулюються запитання відповідно до предметної області документа, а потім із документа витягуються відповіді. Запитання можуть бути такими як: де

відбулася подія X, коли відбулася подія X, хто проводив подію X, який був вплив події X тощо. На ці запитання зазвичай відповідають, знаходячи частину мови термінів та понять у тексті. Відповідаючи на ці запитання, вони вводяться в певний шаблон, який потім допомагає створити остаточне абстрактне резюме. Але знову ж таки, як і в підході на основі шаблонів, правила пишуться вручну, що призводить до втрати часу.

2.2.2 Методи на основі семантики

В методах на основі семантики (Semantic-Based Approach) спочатку семантичне представлення тексту отримується шляхом пошуку інформаційних елементів, структури предикатів-аргументів або створення семантичних графів. Потім це представлення передається в систему генерації природної мови, і за допомогою фраз іменників та дієслів створюється остаточне абстрактне резюме. Алгоритм перетину тем вмісту та графіки використовуються для визначення загальних фраз за допомогою структури предикат-аргумент.

Методи на основі інформаційних елементів (Information Item Based Methods, INIT) використовують інформаційні елементи, найменшу одиницю зв’язної інформації в тексті. Інформаційні елементи означають сутності, їхні характеристики та зв’язки або властивості між ними. Щоб знайти інформаційні елементи використовується семантичне рольове моделювання, усунення неоднозначності, аналіз спів-посилання, аналіз подібності та аналіз логіки предикатів. Також використовуються триплети підмета, дієслова та об’єкта разом із інформацією про час та місце для створення підсумкових речень. Загальна структура для методів на основі INIT містить чотири модулі, а саме:

– пошук елементів інформації, де триплети виділяються за допомогою парсера;

– генерація речень за допомогою генератора мови;

– відбір речень шляхом пошуку речень із найвищим рейтингом, який обчислюється на основі таких факторів, як частота документа тощо;

– генерація резюме за допомогою планування.

Підхід на основі предикатів-аргументів (Predicate-Argument Based Approach). Предикатно-аргументна структура (Predicate-Argument Structure, PAS) означає дієслова, підмет та об’єкт речення. Ця структура згенерована для речень, щоб представити їх семантично. Проводиться пошук семантично подібних структур, наприклад за допомогою вимірювань подібності. Семантично подібні структури об’єднуються разом за допомогою деяких методів, таких як K-середні (K-means) та алгоритму ієрархічної кластеризації (Hierarchical clustering). Функції витягуються зі структур предикатів-аргументів, а потім оцінюються. Щоб максимізувати показники помітності речень, використовуються такі підходи до оптимізації, як цілочисельне лінійне програмування (ILP). PAS з високою оцінкою вибираються та передаються в систему генерації мови для створення остаточного резюме. Але створювати з них нові речення є дуже складним завданням за допомогою цього методу. У цьому підході можна використовувати семантичне рольове моделювання (Semantic role labeling), щоб полегшити створення резюме [7].

Методи на основі семантичних графів (Semantic Graph-Based Methods). Це один із найпопулярніших способів представлення тексту на основі семантичних зв’язків між реченнями в тексті. Семантичні властивості включають онтологічні відношення та синтаксичні відношення між словами. Онтологічні відношення використовують властивість синонімії, гіпонімії, гіперонімії тощо. У той час як синтаксичні відношення використовують властивість зв’язку між словами на основі суб’єкт-об’єкт-дієслово, тобто представлені в термінах дерева залежностей і синтаксичного дерева. Тут документ представлено за допомогою семантичного графа. Іменники та дієслова представлені як вузли графа, а відносини між вузлами представлені ребром.

Інший запропонований підхід використовує FrameNet, де кожне речення розглядається як вершина, а семантичні відносини між реченнями представлені як ребра. FrameNet – це лексична база даних, яка надає семантичні кадри (semantic frame) та лексичні одиниці (lexical unit), які використовуються для визначення значення слова чи речення, класифікації речень та визначення релевантності між реченнями, а точність досягається за допомогою WordNet.

Насичені семантичні графіки (Rich Semantic Graphs, RSG) – це один із способів представлення семантичної інформації тексту у вигляді графіків. Іменники та дієслова тексту представляють вузли, а семантичне та топологічне відношення між ними відповідає краям. Іменники та дієслова тексту отримують за допомогою розбору або онтології. За потреби графік можна зменшити за допомогою евристичних правил, таких як заміна, видалення або комбінування вузлів.

Графи представлення абстрактного значення (Abstract Meaning Representation, AMR): Графи AMR – це позначені, вкорінені та спрямовані ациклічні графи речень. Вони забезпечують семантичне представлення речення. Вузли графів представляють концепції, а ребра представляють відношення між концепціями. Графи AMR представляють багато інформації про текст, як-от названі об’єкти, маркування семантичних ролей, структура предикат-аргумент та спів-посилання. Концепти – це переважно або англійські слова, або кадри PropBank.

PropBank – це ресурс який містить тексти, анотовані інформацією про основні семантичні ознаки. Незважаючи на те, що для аналізу доступна велика кількість синтаксичних аналізаторів, для аналізу використовується здебільшого парсер JAMR, це семантичний аналізатор, генератор та вирівнювач для AMR. Він виконує синтаксичний аналіз у два кроки:

– ідентифікація ключових понять за допомогою напівмарківської моделі;

– визначення зв’язків між поняттями шляхом пошуку максимально зв’язного остовного графа.

Оскільки подібні поняття об’єднуються разом, під час створення вихідного графа для речення кожна сутність представлена лише один раз. Таким чином, незалежно від того, скільки разів концепт з'явився в реченні, він представлений лише одним вузлом. І коли це застосовано до кількох речень, призведе до фінального графіка без надлишків. Графіки AMR використовуються в трьох представленнях, а саме у формі поєднання, щоб знайти подібність між більш ніж одним графіком AMR, у нотації PENMAN та як структура даних графіка. PENMAN – це формат серіалізації для спрямованих, кореневих графів, які використовуються для кодування семантичних залежностей, особливо в рамках AMR.

Мультимодальний семантичний метод. Більшість інформації в Інтернеті не є чисто текстовою. Разом із текстом вони здебільшого містять зображення, відео тощо. Загальну структуру мультимодальної сумаризації можна розділити на три етапи, а саме: побудова семантичної моделі концептів, рейтинг концептів на основі таких факторів, як повнота, та формування речень. Інформаційну щільність можна використовувати як метрику для обчислення вмісту, отриманого за допомогою цього підходу. Спочатку створюється семантична модель, використовуючи представлення знань. Потім створюються концепції та витягується семантична інформація з обмеженого конкретного класу зображень. Наступним кроком є визначення важливих понять за допомогою показника щільності інформації. Методи фразування використовуються для формування резюме.

Іншим способом є створення коротких резюме складних речень, визначаючи основну думку та сутності з тексту. Потім отримані резюме підставляються під структуру зображень документа. Онтологія використовується в цьому підході для пошуку понять, але в цих підходах не використовується автоматично створена онтологія, що призводить до втрати

часу. Крім того, немає автоматичного методу оцінки створених анотацій, що обмежує ефективність цих методів.

2.2.3 Методи на основі глибокого навчання та нейронних мереж

Глибоке навчання є частиною методів машинного навчання та передбачає навчання на основі даних. Цей підхід походить від використання кількох рівнів нелінійної обробки для вилучення ознак із тексту. Навчання може бути як контрольованим, так і неконтрольованим. В їх основі лежать штучні нейронні мережі. Глибинне навчання було успішно застосовано до різних завдань NLP. Синтаксичні аналізатори на основі рекурентних нейронних мереж (Recurrent Neural Network, RNN) досягли найвищої продуктивності в парсінгу залежностей та конституційному парсінгу. Моделі RNN допомагають передбачити складні відносини, які не можуть зробити прості структуровані або семантичні підходи поодинці. Але RNN мають досить вагомі недоліки:

– RNN приймає як вхідні дані речення та обробляє їх слова послідовно, по одному. Їм дуже важко працювати з великими послідовностями тексту, як-от довгі абзаци чи есе. До кінця абзацу вони забудуть, що сталося на початку;

– RNN дуже важко тренувати. Відомо, що вони чутливі до так званої проблеми зникнення/вибуху градієнта (Exploding gradient problem/Vanishing gradient problem). Також оскільки вони обробляли слова послідовно, RNN було важко розпаралелювати. Це означало, що ви не можете просто пришвидшити навчання, надавши їм більше обчислювальної потужності, що, у свою чергу, означало, що ви не зможете навчити їх на такій великій кількості даних.

RNN колись вважався найліпшим підходом для сумарізації тексту, навіть з його вагомими недоліками, але у наш час вже вважається застарілим, класичним підходом. На заміну RNN були запропоновані LSTM [8]. LSTM – Long Short-Term Memory, або Довга короткочасна пам'ять – це штучна нейронна мережа, яка використовується в галузі штучного інтелекту та

глибокого навчання. На відміну від стандартних нейронних мереж прямого зв’язку, LSTM має зворотні зв’язки. Така рекурентна нейронна мережа (RNN) може обробляти не лише окремі точки даних, але й цілі послідовності даних (наприклад, мова чи відео).

Також використовується більш спрощений варіант LSTM, який називається GRU (Gated Recurrent Units, або Вентильні рекурентні вузли) – це вентильний механізм в RNN, його застосовують для вирішення проблеми зникаючого градієнта в RNN [9]. Ворота забування вирішують, яку інформацію відкинути з попереднього стану, присвоюючи попередньому стану значення від 0 до 1. Цей механізм дозволяє відкидати нерелевантну інформацію, звільняючи місце для потенційно більш важливих даних.

Архітектура енкодер-декодер. Енкодер і декодер – це окремі нейронні мережі, які працюють разом як об’єднана нейронна мережа. Завдання енкодера – зрозуміти вхідні послідовності, тоді як завдання декодера – визначити послідовності та згенерувати відповідь.

Енкодер перетворює слова у векторне представлення, яке допомагає фіксувати контекст. Переважно як вкладання слів (Word embedding). Вкладання слів використовується для представлення слів в енкодерах, але багато хто також використовує модель “торби слів” (Bag-of-words model). Як правило, представлення є дійсним вектором, який кодує значення слова таким чином, що слова, які розташовані ближче у векторному просторі, мають бути схожими за значенням [10].

Декодери допомагають знайти наступне слово в анотації на основі попередніх слів. Коли і вхід, і вихід мають форму послідовності, як у випадку підсумовування тексту, проблеми навчання також називаються проблемами навчання Seq2Seq (Sequence to sequence, або Послідовність до послідовності) – це сімейство підходів машинного навчання, які використовуються для обробки природної мови. Програми включають переклад мови, підписи до зображень, розмовні моделі та підсумовування тексту.

Кілька поширених мереж, які використовуються в моделях енкодера-декодера для вирішення проблеми абстрактного підсумовування:

– Згорткова нейронна мережа (Convolutional neural network, CNN): Тут розмір вхідного даних завжди фіксований. Кожен вхід мережі не залежить від попередніх і майбутніх входів;

– RNN: у більшості практичних застосувань розмір вхідних даних зазвичай не є фіксованим, а входи моделі залежать один від одного. Більше того, кількість передбачень, необхідних для виведення, також не є фіксованою. Повторювані з'єднання додаються до нейронних мереж, що допомагає вловити залежність вхідних даних від попередніх або майбутніх вхідних даних;

– LSTM та GRU: RNN зчитує вхідні дані зліва направо й оновлює стан після кожного слова. Але коли ми досягаємо кінця, інформація про перші кілька слів втрачається. LSTM дає доступ до операцій забування, читання та запису, які допомагають зберегти контекст та вловити довгочасні залежності в тексті.

Однак використання цієї базової структури призводить до генерації тривіальних резюме з кількома проблемами, такими як: труднощі в роботі з довгостроковими залежностями та словами поза словником (OOV, Out of Vocabulary). Інші проблеми включають створення неякісних резюме, які втрачають деякі важливі деталі або включають неточні фактичні дані з повторюваними фразами. Таким чином, дослідники автоматичної сумаризації знайшли деякі методи та механізми подолання цих проблем та покращення якості результатів, такі як механізм уваги, механізм копіювання, механізм охоплення та механізм розширення знань.

Механізм уваги – це техніка, призначена для імітації когнітивної уваги людини. Ефект підсилює деякі частини вхідних даних, а інші пригнічує. Мотивація полягає в тому, що мережа повинна приділяти більше уваги невеликим, але важливим частинам даних, як це робить людина при читанні. Механізм уваги здебільшого використовується в моделях послідовності, де

інформація витягується з енкодера на основі показників уваги, і ця інформація використовується декодером. Увага допомагає зрозуміти, яку частину інформації нам потрібно зосередити на певній часовій позначці [11].

Променевий пошук (Beam Search). Під час тестування моделі Seq2Seq генерують повністю сформовані послідовності слів шляхом пошуку вихідних послідовностей за допомогою алгоритмів жадібного або променевого пошуку. У більшості моделей послідовності до послідовності, після визначення ймовірностей вихідних помітних маркерів на рівні уваги, алгоритм пошуку променя може бути використаний під час декодування для переваги гіпотез, відвідування помітних об’єктів, скорочення простору пошуку та зменшення обчислювальної складності [12].

Алгоритм жадібного пошуку має тенденцію безпосередньо вибирати слова з найбільшою ймовірністю на кожному кроці часу, що призводить до створення незв’язного речення з неточним порядком слів, тобто речень з поганою читабельністю. З іншого боку, алгоритм пошуку за променем забезпечує створення зв’язних та більш читабельних речень з правильним порядком слів. Алгоритм променевого пошуку на основі графа розглядає найвищі ймовірності наступного слова, які називаються розміром променя, а потім вибирає пропозицію, яка має найкращу сукупність ймовірностей.

Однак різноманітність та пошук локальних оптимумів є основними обмеженнями променевого пошуку. Пошук різноманітних променів (DBS, Diverse Beam Search) [13] вирішує першу проблему: він дає різноманітні результати шляхом оптимізації для цілі, розширеної різноманітністю. Було проведено численні дослідження ATS з використанням DBS для створення більш інформативних підсумків та підвищення абстрактної новизни ATS. Другу проблему можна вирішити за допомогою моделі прогнозування (мережі цінностей), заснованої на підходах RL [14].

RL – Reinforcement Learning, або Навчання з підкріпленням – це техніка машинного навчання на основі зворотного зв’язку, за якої агент вчиться

поводитися в середовищі, виконуючи дії та отримуючи результати дій. За кожну хорошу дію агент отримує позитивний відгук, а за кожну погану дію агент отримує негативний відгук або штраф.

Механізм копіювання. В абстрактних моделях звернення уваги ATS від послідовності до послідовності декодер може вибирати неточні фрази зі свого словника під час процесу перефразування. Іншою проблемою є нездатність обробляти рідкісні або нові слова, що призводить до генерації невідомого токена. Невідомий токен, або Unknown Token чи UNK – це спеціальний токен для представлення слів, яких немає в нашому словнику. Загалом це поганий знак, якщо ви бачите, що токенайзер виробляє багато таких токенів, оскільки він не зміг отримати розумне представлення слова, і ви втрачаєте інформацію в процесі. Мета під час створення словника полягає в тому, щоб зробити це таким чином, щоб токенайзер створював якомога менше невідомих токенов [15].

Дослідники запропонували вирішення цих проблем за допомогою механізму копіювання за допомогою вказівних мереж, які дозволяють моделі запозичувати слова з вихідного вхідного тексту за потреби. Це призводить до поєднання екстрактивного та абстрактного резюме. Були запропоновані різні моделі для вдосконалення механізму копіювання, включаючи використання механізму повторного читання [16] та гібридного генератора вказівників [17]. Однак цей механізм призводить до проблеми низької новизни, оскільки моделі, як правило, включають велику кількість фраз з оригінального тексту в створене резюме.

Механізм покриття. Незважаючи на те, що моделі з механізмом копіювання можуть ефективно вирішувати позасловникові слова та неточні деталі, поєднуючи цілі вилучення та абстрактності, вони не можуть вирішити проблему повторюваної інформації в резюме. Щоб подолати цю проблему, був розроблений механізм покриття, який відстежує інформацію, що вже була згенерована в резюме. Він допомагає коригувати увагу в майбутньому, щоб

уникнути повторення. Вектор покриття – це сума розподілу уваги за всіма попередніми токенами декодера. Механізм охоплення використовувався різними способами для вирішення абстракції вищого рівня шляхом узагальнення довгих текстів на рівні документа. Дослідники використали механізм покриття з іншої точки зору, використовуючи історію для відволікання, щоб уникнути повторення та покращити ефективність моделі [18].

Механізм розширеного знання (Knowledge-enhanced mechanism) був запропонований для того, щоб включити попередні знання для керування процесом генерації та покращення якості резюме. Цей механізм може включати різні види вказівної інформації, такої як ключові слова, токени довжини, виразні речення, описи фактів та семантичні залежності, представлені графіками та відношеннями. Приклади того, як цей механізм використовувався в абстрактних ATS, включають: включення ключових слів вхідного тексту для контролю вмісту резюме та вилучення описів фактів із вихідного тексту для зменшення генерації фальшивої інформації [19].

Двоетапна стратегія навчання (Two-stage learning strategy) може бути використана як спосіб подальшого підвищення продуктивності абстрактної ATS [10]. Вона передбачає використання кількох моделей. У цьому підході резюме-кандидатів, створені однією моделлю, використовуються як вхідні дані для іншої моделі, яка в кінцевому резюме генерує найкращий результат. Перша модель, як правило, є простішою моделлю, яка генерує початкові резюме-кандидати, тоді як друга модель є більш складною моделлю, яка вдосконалює резюме-кандидати у більш точні та послідовні резюме.

ELMo – це спосіб представлення слів у векторах або вбудовуваннях. Вбудовування слів допомагають досягти найсучасніших результатів у кількох завданнях NLP [20]. На відміну від традиційних вставок слів, таких як word2vec [21] та GLoVe [22], вектор ELMo, призначений лексемі або слову, насправді є

функцією всього речення, що містить це слово. Тому те саме слово може мати різні вектори слів у різних контекстах.

2.2.3.1 Модель Трансформер

Трансформер [11] – це тип архітектури нейронної мережі, яка була розроблена для завдань NLP. Не так давно вона стала однією з найпоширеніших архітектур у цій галузі.

Архітектура базується на концепції само-уваги (self-attention) [11], яка дозволяє моделі звертати увагу на різні частини вхідної послідовності під час обчислення представлення кожного слова чи лексеми. Ідея уваги натхненна людським розумінням об’єкта чи тексту шляхом зосередження лише на певних частинах. Наприклад, замість того, щоб звертати увагу на всі частини, дивлячись на зображення, людина зосереджується лише на конкретних деталях, щоб краще зрозуміти це зображення. Подібним чином можна дозволити моделі зосереджуватись лише на конкретних видах інформації, які вважаються найважливішими для досягнення кращого розуміння.

У традиційній архітектурі нейронної мережі, такій як RNN, вхідна послідовність обробляється послідовно, одне слово за раз. Трансформер може обробляти всю вхідну послідовність паралельно, що дозволяє йому бути набагато ефективнішим для довгих вхідних послідовностей.

Попередньо навчені мовні моделі (PTLM, Pre-Trained Language Models) навчають Трансформери на величезних корпусах та передають отримані знання для таких завдань, як ATS. Як наслідок, багаті семантичні та контекстуальні характеристики вбудованих слів, отриманих за допомогою PTLM, доводять, що вони можуть допомогти у підвищенні якості кінцевих анотацій. Широкий успіх Трансформерів та PTLM для різних завдань NLP, включаючи абстрактні автоматичні підсумовувачі, робить їх основою останніх досліджень NLP.

Трансформер складається з колекцій енкодерів та декодерів. Всередині кожного енкодера є два шари: шар само-уваги та нейронна мережа прямого поширення. Декодер має обидва ці рівні, але між ними знаходиться рівень уваги, який допомагає декодеру зосередитися на відповідних частинах вхідного речення. На рисунку 2.1 представлена схема архітектури Трасформера.

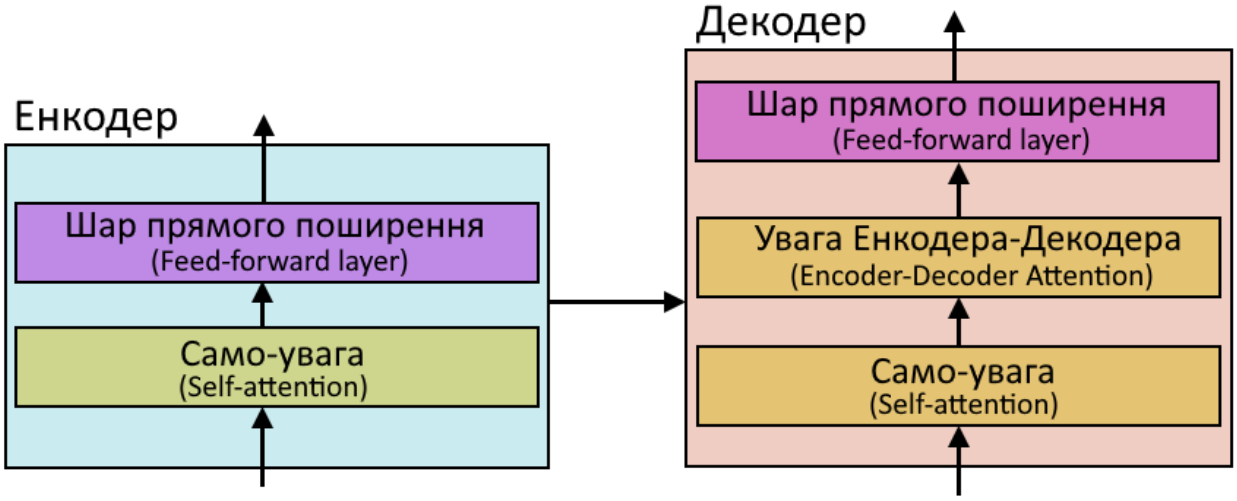

Рисунок 2.1 – Схема Трансформера

Щоб мовна модель могла передбачити значення тексту, вона має знати про контекстну подібність слів. Речення, що надходять до трансформера, перетворюються на вбудовування слів, що є низькорозмірним векторним представленням корпусу тексту, яке зберігає контекстну подібність слів. Крім того, що вбудовування слів піддається обробці за допомогою алгоритмів навчання, вони є ефективнішим та виразнішим представленням слів. Вбудовування відбувається лише в самому нижньому енкодері. Абстракція, яка є спільною для всіх енкодерів, полягає в тому, що вони отримують список векторів, кожен розміром 512. У нижньому енкодері це буде вбудовування слів, але в інших кодерах це буде вихід енкодера, який знаходиться прямо під ним. Розмір цього списку векторів – гіперпараметр (Hyperparameter), це параметр, значення якого керує процесом навчання та визначає значення параметрів

моделі, які вивчає алгоритм навчання. Слово в кожній позиції проходить через процес само-уваги. Потім кожен з них окремо проходить через нейронну мережу прямого зв’язку (рис. 2.2). Саме тут можно спостерігати одну із основних переваг трансформеру – обробка слів може виконуватися паралельно.

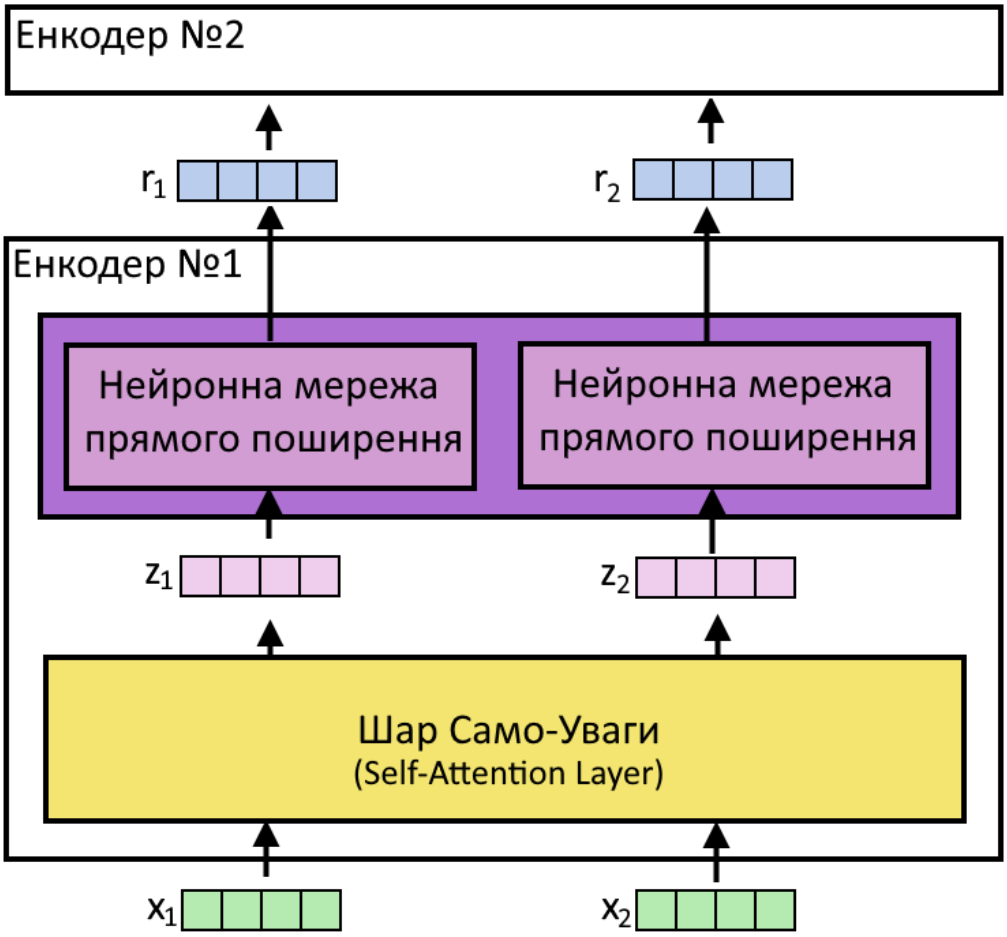

Рисунок 2.2 – Схема проходження слів через енкодери

Само-увага дозволяє моделі зрозуміти, коли ми непрямо посилаємося на об’єкти. Оскільки модель обробляє кожне слово (кожну позицію у вхідній послідовності), самоувага дозволяє їй переглядати інші позиції у вхідній послідовності для підказок, які можуть допомогти привести до кращого кодування цього слова.

Шар само-уваги бере вхідний текст та створює три вектори для кожного слова, ці вектори називають: запит (query), ключ (key) та значення (value).

Вектори створюються шляхом множення вбудовування на три матриці, які ми отримуємо в процесі навчання моделі. Ці вектори використовуються для обчислення оцінки кожного слова порівняно з іншими словами в реченні. Вхідні вектори мають розмірність 512, тоді як обчислені вектори мають розмірність 64. Вони навмисно менші, щоб зробити обчислення багатосторонньої уваги постійним. Оцінка обчислюється шляхом скалярного добутку вектора запиту на вектор ключа відповідного слова, яке ми оцінюємо. Щоб отримати більш стабільні градієнти, оцінки діляться на 8, це значення за замовчуванням, але воно може бути іншим. Потім використовується функція софтмаксу (softmax) для нормалізації оцінок, щоб усі вони були позитивними та складалися в 1. Потім кожен вектор множиться на оцінку софтмаксу. Це робиться для того, щоб зберегти значення слів, на яких ми хочемо зосередитися, і заглушити нерелевантні слова. Вихід із рівня само-уваги для поточного слова є сумою всіх зважених векторів значення. У реальній реалізації для швидшої обробки замість окремих векторів використовуються матриці. Повний процес обчислення само-уваги на прикладі векторів представлений на рисунку 2.3.

Рисунок 2.3 – Схема обчислення само-уваги на прикладі векторів

Наступним етапом розвитку самоуваги є «багатостороння» увага. Це покращує продуктивність шару двома основними способами:

– розширює здатність моделі зосереджуватися на різних позиціях. Вихід шару зазвичай містить трохи кожного іншого кодування, але в ньому може переважати саме слово. Якщо ми перекладаємо речення на зразок «Яблуко впало, тому що воно було занадто важким», було б корисно знати, до якого слова «воно» відноситься;

– надає шару уваги кілька «підпросторів представлення». З багатосторонньою увагою ми маємо кілька наборів вагових матриць запиту/ключу/значення. Наприклад Трансформер використовує 8 наборів для

кожного енкодера/декодера. Кожен із цих наборів ініціалізується випадковим чином. Потім, після навчання, кожен набір використовується для проектування вхідних вкладень в інший підпростір представлення.

Кожна з голов уваги має власну матрицю запит/ключ/значення, а також навчені зважені матриці з відповідними іменами. Голова уваги обчислює увагу так само, як описано вище, за винятком того, що вектори замінені матрицями. У підсумку ми отримуємо 8 різних матриць уваги від кожної з голов уваги. Усі ці матриці об'єднуються та множаться на іншу додаткову матрицю ваг W^o, яку також отримують шляхом навчання.

Щоб представити порядок слів у реченні, Трансформер додає вектор до кожного вбудовування. Ці вектори слідують певному шаблону, який вивчається моделлю, що допомагає їй визначити позицію кожного слова або відстань між різними ними у послідовності. Додавання цих значень до вбудовування забезпечує значущі відстані між векторами вбудовування після того, як вони проектуються у вектори запиту/ключу/значення та під час урахування скалярного добутку.

Потім вихідні дані верхнього енкодера перетворюються на набір векторів уваги ключ та значення. Вони використовуються кожним декодером на рівні «Увага енкодера-декодера», який допомагає декодеру зосередитися на відповідних місцях у вхідній послідовності. Наступні кроки повторюють процес, доки не буде досягнуто спеціального символу, який вказує, що декодер Трансформера завершив свій вихід. Вихідні дані кожного кроку подаються на нижній декодер на наступному часовому кроці, і декодери передають свої результати декодування вгору. Позиційні кодери додаються для кожного з входів декодера.

Рівням само-уваги в декодері дозволено звертати увагу лише на попередні позиції у вихідній послідовності. Це робиться шляхом маскування майбутніх позицій (встановлення їх значення в негативної нескінченності) перед кроком софтмаксу у розрахунку само-уваги. Шар «Увага енкодера-декодера» працює

так само, як багатостороння само-увага, за винятком того, що він створює свою матрицю запитів із шару під ним і бере матриці ключ та значення із вихідних даних стека енкодера.

Декодер виводить стек чисел з плаваючою точкою, які потім обробляються лінійним та софтмакс шарами. Лінійний шар – це проста повністю зв’язана нейронна мережа, яка проектує вектор, створений стеком декодерів, у набагато більший вектор, який називається логит вектором. Довжина цього вектора дорівнює кількості слів, які модель «знає». Кожен сегмент вектора відповідає одному слову та зберігає його оцінку. Софтмакс інтерпретує цю оцінку в ймовірності, які є позитивними та складаються в 1. Вибирається ячійка з найвищою ймовірністю, а пов’язане з нею слово виводиться як результат для цього часового кроку.

На рисунку 2.4 представлена схема архітектури Трансформер, як зображено в оригінальній публікації [11].

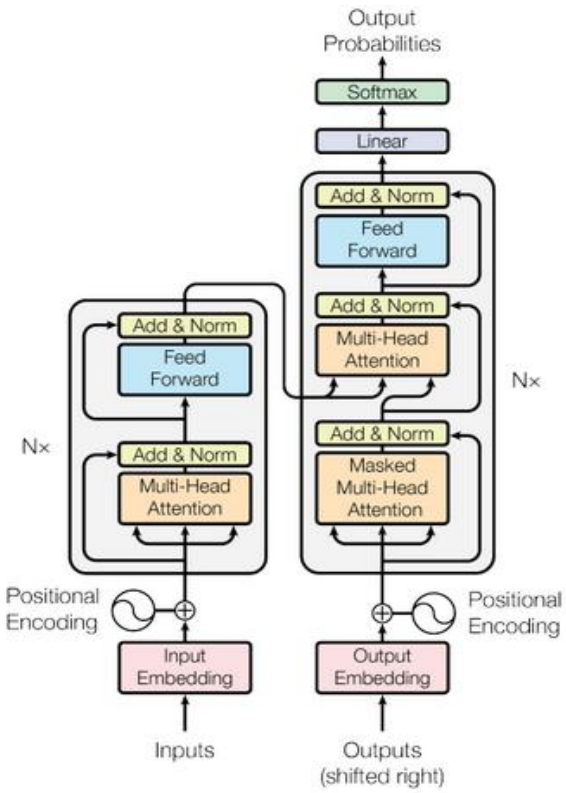

Рисунок 2.4 – Трансформер: модель архітектури

Однією з головних переваг архітектури Трансформер є її здатність моделювати довгострокові залежності у вхідній послідовності. Це досягається за рахунок використання само-уваги, що дозволяє моделі звертати увагу на будь-яку іншу частину послідовності, незалежно від її відстані від поточної позиції. Це робить Трансформер особливо ефективним для таких завдань, як переклад мови, де довгострокові залежності часто важливі.

Ще однією перевагою архітектури Трансформер є її розпаралелювання, що дозволяє навчати її набагато ефективніше, ніж традиційні архітектури нейронних мереж. Використання самоуваги також робить її менш схильною до проблеми зникнення градієнта, яка може виникнути в RNN.

Однак основні обмеження механізму само-уваги можна підсумувати як квадратичну пам’ять та обчислювальну складність, велику кількість операцій, необхідних під час роботи з довгими послідовностями, і те що контекст фіксованої довжини є необхідним для вивчення довгострокових залежностей.

Загалом, архітектура Трансформер довела свою високу ефективність для широкого спектру завдань обробки природної мови та стала однією зі стандартних моделей у цій галузі. Інші популярні моделі на основі архітектури Трансформер включають:

– BERT (Bidirectional Encoder Representations from Transformers, Двоспрямовані кодувальні представлення з трансформерів) – це попередньо навчена модель трансформера 2018 року для завдань розуміння природної мови, таких як аналіз настроїв та відповіді на запитання [23]. Він використовує завдання моделювання замаскованої мови для попереднього навчання моделі на великих обсягах текстових даних, що робить можливим тонке налаштування (fine-tuning) моделі для наступних завдань з відносно невеликою кількістю навчальних даних. BERT ефективний як для тонкого налаштування, так і для стратегій на основі ознак. Однак BERT має обмеження, такі як: нехтування залежностями замаскованих позицій і невідповідність тонкого налаштування

перед навчанням. Ці проблеми вирішуються іншими PTLM, такими як моделі XLNet [24] та Ernie-Gen [25]. Крім того, оскільки прогнози в BERT не виконуються авторегресійно, BERT краще підходить для завдань NLU, ніж для завдань генерації тексту (TG, Text Generation);

– GPT (Generative Pre-trained Transformer, Генеративний попередньо навчений трансформер) – це сімейство авторегресійних мовних моделей, які відрізняються своєю вражаючою продуктивністю в кількох еталонних завданнях NLP та здатністю створювати високоякісний текст, який часто важко відрізнити від написаного людиною. На даний момент виділяють три версії: GPT-2 [26], GPT-3 [27] та GPT-3.5, кожна з яких є еволюцією попередньої та демонструє навіть кращі показники. Остання навчена на величезній кількості текстових даних (45 терабайт) та здатна виконувати широкий спектр завдань обробки природної мови, включаючи переклад мови, відповіді на запитання та завершення тексту;

– представлений у 2019 році T5 (Text-to-Text Transfer Transformer, Трансформер для перетворення тексту-в-текст) – це модель на основі Transformer, яку можна навчити виконувати широкий спектр завдань із перетворення тексту в текст, зокрема машинний переклад, узагальнення та відповіді на запитання [29]. T5 відрізняється своєю гнучкістю та здатністю досягати найсучасніших результатів у широкому діапазоні завдань з мінімальними змінами архітектури для конкретних завдань;

– розроблений компанією Google у 2020 році Pegasus – це модель на основі Transformer для абстрактного підсумовування тексту. Він використовує підхід попереднього навчання під назвою «неконтрольована генерація масових парафраз» (UMPG, unsupervised massive-scale paraphrase generation), який генерує синтетичні парафрази уривків тексту та використовує їх для навчання моделі [30]. Pegasus відомий своєю здатністю генерувати високоякісні анотації, які є вільними та інформативними, модель досягла найсучасніших результатів на кількох контрольних наборах даних. Перевага Pegasus полягає в тому, що він

маскує кілька завершених речень замість менших безперервних текстів. Крім того, він вибирає речення на основі їх важливості, а не випадково, як інші моделі. Pegasus-X це розширена версія моделі, яка може похизуватися більшим об'ємом токенів, да и здебільшого розрахована на великі об'єми вхідних даних [31];

– модель BigBird – це трансформер на основі розрідженої уваги (sparse attention), який розширює моделі на основі Трансформера, такі як BERT, для набагато довших послідовностей [32]. Окрім розрідженої уваги, BigBird також застосовує глобальну увагу, а також довільну увагу до послідовності введення. Теоретично було показано, що застосування розрідженої, глобальної та довільної уваги наближає повну увагу, хоча обчислювально набагато ефективніше для довших послідовностей. Завдяки здатності працювати з довшим контекстом BigBird продемонстрував кращу продуктивність у виконанні різноманітних завдань NLP із довгими документами, таких як відповіді на запитання та підсумовування, порівняно з BERT або RoBERTa [33];

– Galactica (GAL) – це нова велика мовна модель, призначена для автоматичної організації науки. Galactica була навчена на великому та відібраному корпусу наукових знань людства. Це понад 48 мільйонів статей, підручників та конспектів лекцій, мільйони сполук і білків, наукові веб-сайти, енциклопедії тощо. На відміну від існуючих мовних моделей, які покладаються на парадигму без підготовленого сканування, корпус Galactica є високоякісним і добре курованим [34];

– Longformer – це модифікація оригінальної архітектури Трансформер. Традиційні трансформери не можуть обробляти довгі послідовності через їхню операцію само-уваги, яка квадратично масштабується з довжиною послідовності. Щоб вирішити цю проблему, Longformer використовує шаблон уваги, який лінійно масштабується залежно від довжини послідовності, що полегшує обробку документів із тисячами токенів або більше. Механізм уваги є

заміною стандартної само-уваги та поєднує локальну віконну увагу з глобальною увагою, мотивованою завданням [35];

– BART – це автоматичний кодер для усунення шумів для попереднього навчання моделей seq2seq. BART навчається шляхом спотворення тексту за допомогою довільної функції шуму та вивчення моделі для реконструкції оригінального тексту. Він використовує стандартну архітектуру нейронного машинного перекладу на основі Трансформер, яку, незважаючи на її простоту, можна розглядати як узагальнюючу BERT, GPT та багато інших пізніших схем попереднього навчання. BART особливо ефективний для генерування тексту після точного налаштування, але також добре працює для завдань на розуміння [36]. Відносно нещодавно запропонована модифікація BART з використанням LSG (Local Sparse Global) уваги демонструє досить непогані результати в завданнях сумаризації довгих документів [37].

Розріджена увага зменшує час обчислення та вимоги до пам’яті механізму уваги, обчислюючи обмежений вибір балів подібності з послідовності, а не з усіх можливих пар, що призводить до розрідженої матриці, а не повної матриці.

Однак навіть обчислення єдиної матриці уваги може стати непрактичним для дуже великих вхідних даних. У шаблонах розрідженої уваги кожна вихідна позиція обчислює лише ваги з підмножини вхідних позицій. Коли підмножина мала відносно повного набору вхідних даних, результуюче обчислення уваги стає доступним навіть для дуже довгих послідовностей із складністю алгоритму.

Існують дві основні стратегії для подальших завдань для застосування попередньо навчених мовних представлень: на основі ознак і тонкого налаштування. У стратегії на основі ознак фіксовані ознаки витягуються з попередньо навченої моделі, при цьому необхідна специфічна для завдання архитектура, а попередньо навчені представлення використовуються як додаткові ознаки. З іншого боку, у підході тонкого налаштування до попередньо навченої моделі часто додається простий рівень класифікації.

Подальша задача навчається шляхом спільного точного налаштування всіх попередньо навчених параметрів та введення мінімальних параметрів, специфічних для завдання.

Останнім часом тонке налаштування стало традиційною стратегією адаптації більшості PTLM. Однак точно налаштовані параметри не однакові для різних наступних завдань. У сучасних мовних моделях вихідні вектори нейронних енкодерів представляють семантику слова залежно від його контексту. Ці вектори називаються контекстними вкладеннями слів. В даний час одним із популярних напрямків досліджень є модифікація та/або адаптація Трансформерів та PTLM до різних задач розуміння та генерації.

Кількість гіперпараметрів визначає внутрішню складність моделі, яка зазвичай впливає на якість виведення, а також на швидкість і складність пам'яті. Навчальний датасет зазвичай визначає задачу та предметну область для якої модель може бути застосована за замовчуванням. Кожна з цих моделей може бути тонко налаштована під будь-яку предметну область, за бажанням та доступністю ресурсів.

Зараз проводиться тестування наступної ступени еволюції GPT, а саме GPT-4. Найбільшою різницею можно назвати те що, ця модель може оперувати не тільки текстом, але й відео та аудіо. GPT-4 вже може вирішувати нові та складні завдання, які охоплюють математику, кодування, бачення, медицину, право, психологію тощо, не потребуючи жодних спеціальних підказок [38]. Його можна розумно розглядати як ранню версію AGI (Artificial general intelligence, або Загальний штучний інтелект) – це гіпотетичний розумний агент, який може розуміти або вивчати будь-які інтелектуальні завдання, які можуть люди чи інші тварини. Дуже вірогідно, що саме GPT-4 стане тою системою штучного інтелекту, яку ми звикли бачити в поп-культурі.

У таблиці 2.1 представлені основні характеристики описаних вище моделей на основі Трансформера.

Таблиця 2.1 – Порівняльна таблиця для моделей на основі Трансформера

Наименування моделі	Кількість гіперпараметрів	Токени	Навчальні дані
BERT	345 млн.	512	BookCorpus, English Wikipedia
GPT-3 (GPT-3.5)	175 млрд.	4096	Common Crawl, WebText2, Books1, Books2, Wikipedia
T5	11 млрд.	16384	C4
Pegasus	568 млн.	4096	C4, Hugenews
Pegasus-X	568 млн.	4096	C4, Hugenews
BigBird	-	4096	Books, CC-News, Stories, Wikipedia
BART	400 млн.	1024	English language
LSG BART	145 млн.	16384	Arxiv, Pubmed
Longformer	464 млн.	4096	Books, English Wikipedia, Realnews, Stories
Galactica	120 млрд.	10240	Wikipedia, StackExchange, LibreText, Wikibooks, Open Textbooks, MIT OCW, Wikiversity, ProofWiki, Khan Academy, Papers with Code, IUPAC Goldbook та багато інших

3 ПОСТАНОВКА ЗАДАЧІ

При сумаризації тексту найголовніші критеріями є зміст резюме, його інформативність, відповідність тексту, оригінальність речень, відсутність повторень та неправдивої інформації.

Оскільки довжина статей може коливатися в великому діапазоні від 5 сторінок аж до 40, суммаризацію наукових робіт можна розділити на два завдання: суммаризація коротких документів та суммаризація довгих документів. Звісно, перед опрацюванням публікації, треба витягти текст, видалити всі зайві елементи.

Для порівняння ми припустимо, що всі публікації в оригіналі написані англійською мовою. Причина в тому, що не так багато доступних датасетів іншими мовами, переважно українською. Що дуже ускладнює ситуацію, тому що немає на чому тренувати модель. Звичайно, ви можете перекласти публікацію англійською, підсумувати, а потім знову перекласти резюме українською. Але є кілька проблем із таким підходом: глибина мови може бути втрачена через різницю в мовах, а служби МП не можуть гарантувати абсолютно правильний переклад у будь-який час. Навіть Google досі час від часу допускає помилки в довгих реченнях чи конкретних виразах у перекладі з англійської на українську.

Але варто зауважити, що більшість дослідників публікують свої роботи англійською мовою. Англійська мова – основна мова науки. Наприклад, у 2020 році 98% усіх опублікованих робіт було англійською мовою [39]. Отже, якщо вам не потрібен сервіс для сумаризації тексту вашою мовою для вашої місцевої спільноти дослідників, тоді вам насправді не потрібен датасет. Однак це створює гарне поле для досліджень, якщо хтось хоче створити анотований датасет для сумаризації та точно налаштувати на ньому PTLM.

3.1 Методи оцінки результатів сумаризації

Визначення того, чи резюме хороше, є дуже суб’єктивним. Людям можуть подобатися резюме з нижчою оцінкою лише тому, що їх легше читати або вони просто абстрактно більш привабливі. Погані та застарілі моделі можуть генерувати підсумок, який отримує оцінку людини «достатньо добре», а високо оцінені резюме можуть бути відхилені читачем.

Оскільки виміряти правильність підсумків важко, в більшості досліджень абстрактних ATS використовують ROUGE (Recall-Oriented Understudy for Gisting Evaluation) як стандартний показник оцінки. ROUGE включає набір балів для оцінювання абстрактних завдань ATS та MT. Метрика ROUGE призначена для вимірювання лексичної подібності, тобто накладання n-грамів між підсумковим резюме та посиланням, як правило, резюме, яке було написане людиною [40].

ROUGE-F1 є найпопулярнішим стандартом, який використовується для вимірювання трьох показників ROUGE, а саме ROUGE-1, ROUGE-2 і ROUGE-L. ROUGE-1 вимірює перекриття уніграм, тобто кожного окремого слова, між результуючим та еталонним резюме. У той час як ROUGE-2 вимірює перекриття біграм, тобто кожні два послідовних слова, між результуючим і достовірним резюме. ROUGE-L вимірює найдовшу спільну послідовність між результуючим та еталонним резюме. ROUGE-LSUM вимірює ефективність моделей для суммаризації в контексті довгих документів. Він враховує важливість інформації в різних місцях у довгому документі, а також важливість інформації в самому резюме.

Проте метрика ROUGE має три основні обмеження: її упередженість щодо лексичної подібності, низька увага до плавності та читабельності згенерованих абстрактних резюме та його жорстка передумова використання базових резюме для отримання балів. Таким чином, дослідники розробили нові показники для підвищення новизни, узгодженості фактів та якості на основі

відповідей на запитання та людських оцінок, використовуючи підходи винагороди RL, не вимагаючи коротких резюме.

Вищі бали ROUGE насправді не гарантують вищої якості та кращої читабельності, а навпаки, вони часто отримують нижчі людські оцінки. Це пояснюється тим, що ROUGE не в змозі вловити подібності на таких високих рівнях, як семантичні подібності, а зосереджується лише на локальних подібностях.

Крім того, показник ROUGE не в змозі захопити новизну у високоякісних резюме. Це пов'язано з тим, що резюме з високим рівнем новизни, швидше за все, включатимуть синоніми, крім виразів, включених до достовірних резюме, що призводить до зниження балів перекриття та, отже, до нижчих балів ROUGE.

Для інших показників Meteor [41] та BLEU [42] є двома добре відомими показниками, які в основному призначені для вимірювання результатів машинного перекладу (MT, Machine translation). Однак ці два показники також можуть вимірювати результати ATS.

Оскільки абстрактні моделі перефразовують текст, вони можуть мати низькі оцінки, а високі оцінки можуть не призвести до хороших анотацій у реальних умовах [43].

Для абстрактних ATS були запропоновані більш надійні семантичні метрики, які краще корелюють з людськими оцінками, такі як VERT [44], MoverScore [45] та BERTScore [46]. VERT порівнює підсумки, згенеровані моделлю, та достовірні резюме, поєднуючи показники подібності на рівні документа та несхожості на рівні слів, щоб виміряти семантичну подібність отриманого резюме. MoverScore поєднує контекстуалізовані вбудовування слів з Earth Mover's Distance (EMD), який вимірює семантичну відстань між текстами. Зовсім недавно запропонований BERTScore використовує контекстні вбудовування для обчислення семантичної подібності. MoverScore та

BERTScore починають набирати популярності в оцінюванні абстрактних моделей дослідження ATS.

3.2 Датасети для оцінки методів автоматичної сумаризації

Для завдання суммаризації текста наукових публікацій, а отже наукових текстів, нам найбільше підходять два датасети – ArXiv [48] та PubMed [49]. Обидва датасети складаються мають дві ознаки:

– article: основна частина документа, абзаци розділені символом «/n»;
– abstract: анотація документа, абзаци розділені символом «/n».

Вони нерівно поділені на три частини: навчання, валідація та тестування, за яких навчання це найбільша частина, а валідація та тестування приблизно одного розміру.

Протягом майже 30 років ArXiv служив громадськості та дослідницьким спільнотам, надаючи відкритий доступ до наукових статей, від величезних галузей фізики до багатьох субдисциплін інформатики та всього між ними, включаючи математику, статистику, електротехніку, кількісну біологію, та економіку. Цей багатий масив інформації пропонує значну, але іноді приголомшливу глибину.

Щоб зробити arXiv більш доступним, Корнельський університет представляє безкоштовний відкритий машиночитаємий датасет arXiv: репозиторій з 1,7 мільйона статей із відповідними ознаками, такими як заголовки статей, автори, категорії, анотації, повні тексти PDF-файлів тощо. Відібрана частина датасету для сумаризації поділена як представлено у таблиці 3.1.

Таблиця 3.1 – Розподіл датасету ArXiv

№	Розподіл датасету	Кількість екземплярів	Середня кількість токенів (стаття / резюме)
1	Навчання	203,037	6038 / 299
2	Валідація	6,436	5894 / 172
3	Тестування	6,440	5905 / 174

PubMed містить цитати та реферати біомедичної літератури з кількох літературних ресурсів Національної медичної бібліотеки (NLM) Сполучених Штатів Америки. Відібрана частина датасету для сумаризації поділена як представлено у таблиці 3.2.

Таблиця 3.2 – Розподіл датасету PubMed

№	Розподіл датасету	Кількість екземплярів	Середня кількість токенів (стаття / резюме)
1	Навчання	119,924	3043 / 215
2	Валідація	6,633	3111 / 216
3	Тестування	6,658	3092 / 219

Hugging Face представляє обидва ці датасети як один датасет scientific_papers [50], що загалом дає 322 961 екземплярів. Усі три датасети є досить популярними та активно використовуються. Також існує Multi-XScience [51] датасет який призначений для багато-документної сумаризації, але він також набагато менший – лише 30 000 екземплярів для навчання.

Multi-XScience представляє складне завдання підсумовування декількох документів: написання розділу статті, пов’язаного з роботою, на основі її анотації та статей, на які вона посилається [51].

3.3 Оцінка та порівняння методів сумаризації

Екстрактивні методи самі по собі не спроможні виробляти тексти достойної якості та не є конкурентоспроможними в порівнянні с методами на основі глибинного навчання. Але деякі з мовних моделей можуть використовувати екстрактивні методи в комбінації з нейронними мережами.

Оцінка результатів мовних моделей і генерування тексту є складною темою, оскільки науковець, клієнт та модель по-різному розуміють хороше резюме.

Навіть якщо модель непридатна для задачі резюмування довгого тексту через обмеження вхідних токенів, є способи для таких випадків. Спочатку ви можете спробувати сегментувати текст та підсумувати ці сегменти окремо, а потім або об’єднати ці менші анотації в одну, або підсумувати їх знову. Другий варіант, який показав гідний результат, це вирізання середньої частини тексту. Іншими словами, використовувати першу пару абзаців та останню пару абзаців як вхідну послідовність. Так, такі підходи загалом збільшують час обробки тексту, але це мізерні частки часу.

Деякі моделі сімейства GPT можуть робити деякі граматичні помилки, але BART показує чудові результати в створенні правильних речень. Граматичні помилки інших моделей може бути особливо важко усунути під час постобробки.

Недоліком більш складних моделей на основі GPT-3.5 є те, що вони недоступні для тонкого налаштування. Але вони все одно створюють анотації доцільної якості, навіть якщо контекст виходить за межі знань моделі. BART може бути дешевшим варіантом, ніж моделі на основі GPT-3, з точки зору витрат на бізнес.

Таблиця 3.3 – Оцінки ROUGE для моделей на основі датасету Arxiv [48]

Модель	Навчальний Датасет	ROUGE				Meteor	Loss
		1	2	L	LSUM		
BART	CNN, Pubmed, Arxiv	41.36	15.18	23.86	37.09	-	2.340
LSG BART	Arxiv	48.74	20.88	28.50	**44.23**	-	-
Pegasus	C4, HugeNews	42.2	15.8	37.3	29.2	-	-
Pegasus-X	-	**50.0**	21.8	**44.6**	36.5	-	-
BigBird Pegasus		46.8	19.6	28.0	29.5	0.282	2.774
BigBird Pegasus Large	Arxiv	36.02	13.41	21.96	29.64	0.282	2.774
	Pubmed	40.89	18.11	26.17	34.27	0.351	2.171
Long T5 Large	-	48.3	21.6	44.1	35.8	-	-
Long T5 XL	-	48.4	**21.9**	44.3	36.1	-	-
Longformer	-	46.6	19.6	41.8	33.7	-	-

Модифікований BART з використанням LSG демонструє вищі показники, ніж звичайний BART, але може бути дорожчим варіантом.

Згідно с табл. 3.3 Pegasus-X демонструє найкращі показники ROUGE, але, як вже було зазначено раніше, вищі показники ROUGE не гарантують кращої анотації та якості тексту. Сильна сторона Pegasus все ж таки короткі документи, але це значить що його не можна використовувати для інших завдань.

Galactica доступна для завдання сумаризації за замовчуванням, а також завдяки великому корпусу наукової літератури згенеровані тексти мають дуже високу фактичну відповідність, тобто генерують меньше неправдивої інформації. Основною перевагою моделі є можливість тренуватися на ній протягом кількох епох без перенавчання, де продуктивність висхідного та низхідного потоків покращується за допомогою повторюваних маркерів. Galactica це більше, ніж мовна модель, вона може виконувати мультимодальні завдання, включаючи хімічні формули, математичні рівняння та послідовності білків.

SCROLLS бенчмарк [52] ставить Long T5 XL на перше місце лідерборду для задачі сумаризації довгих документів на датасетах GovRep (Government report, або Урядовий звіт), ScrSum (Screenplay Summarization, або Суммаризація Сценаріїв) та QMSum (Query-based Multi-domain Meeting Summarization, або Суммаризація багатодоменної зустрічі на основі запитів). Але у вищезгаданій таблиці, при оцінці с датасетом Arxiv, посідає лише третю сходинку. Long T5 Large у тому ж самому бенчмарку посідає друге місце, а базовий BART на восьмому.

ВИСНОВКИ

У цій роботі було проведено огляд більшості відомих методів, які використовуються в задачах автоматичної сумаризації тексту. А також їх можливості та особливості відносно використання на довгих документах, наукового чи відносно-наукового змісту. За замовчуванням ми припускаємо, що усі були розглянуті тексти написані англійською мовою, тому що англійська це основна мова науки.

Методи екстрактивної сумаризації, хоч і були представлені, не можуть конкурувати за якістю текстів з абстрактними методами на основі глубокого навчання. Але подекуди використовуються у комбінації з абстрактними методами, наприклад для імпорту термінів із тексту, що обробляється.

Останні кілька років можно спостерігати великий вибух інтересу у NLP, здебільшого завдяки публікації «Attention is all you need» 2017 року. Ця публікація посприяла створенню архітектури Трансформер та великої кількості моделей на її основі, для усіх задач NLP. А потім ще один вибух, вже громадського, інтересу завдяки GPT-3 та GPTchat.

Усі мовні моделі на базі Трансформера демонструють високу якість текста, хоч і можуть подекуди робити помилки та генерувати неправдиву інформацію. Galactica має дуже високий потенціал до мінімізації створеної дезінформації завдяки куророваному датасету з наукових ресурсів. Дві моделі Long T5 демонструють найвищі показники ROUGE в задачі довгої сумаризації, але це не гарантує якість тексту. GPT може бути легшим для інтегрування в уже існуючі системи, а BART може бути дешевшою альтернативою.

Можливи темами подальших досліджень може бути створення нових датасетів для суммаризації довгих текстів, бо деяка частка дослідників жаліється на якість існуючих датасетів. Також є деяка необхідність в датасетах українською мовою, тому що таких фактично немає.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008:2015. Звіти у сфері науки і техніки структура та правила оформлення. [Чинний від 2017-07-01]. Вид. офіц. Київ : УкрНДНЦ, 2016. 31 с. (Інформація та документація).

2. ДСТУ 8302:2015. Бібліографічні посилання. Загальні положення та правила складання. [Чинний від 2016-04-03]. Вид. офіц. Київ : УкрНДНЦ, 2016. 20 с. (Інформація та документація).

3. Text Summarization Techniques: A Brief Survey / M. Allahyari et al. International Journal of Advanced Computer Science and Applications. 2017. Vol. 8, no. 10. P. 397–405.

4. Gialitsis N., Pittaras N., Stamatopoulos P. A topic-based sentence representation for extractive text summarization. MultiLing 2019: Summarization Across Languages, Genres and Sources, Varna, 6 September 2019. 2019. P. 36–34.

5. Assessing sentence scoring techniques for extractive text summarization / R. Ferreira et al. Expert Systems with Applications. 2013. Vol. 40, № 14. P. 5755–5764.

6. Erkan G., Radev D. R. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research. 2004. Vol. 22. P. 457–479.

7. Automatic text summarization: A comprehensive survey / W. S. El-Kassas et al. Expert Systems with Applications. 2021. Vol. 165. 165. 26 p.

8. Hochreiter S., Schmidhuber J. Long Short-Term Memory. Neural Computation. 1997. Vol. 9, no. 8. P. 1735–1780.

9. Gupta S., Gupta S. K. Abstractive summarization: An overview of the state of the art. Expert Systems with Applications. 2019. Vol. 121. P. 49–65.

10. Suleiman D., Awajan A. Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges. Mathematical Problems in Engineering. 2020. Vol. 2020. P. 1–29.

11. Attention is all you need / A. Vaswani et al. Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), New York, 4–9 December 2017. New York, 2017. P. 6000–6010.

12. Furcy D., Koenig S. Limited discrepancy beam search. Proceedings of the 19th international joint conference on Artificial intelligence. 2005. P. 125–131.

13. Diverse beam search: decoding diverse solutions from neural sequence models / A. K. Vijayakumar et al. 2018. 16 p.

14. Deep reinforcement and transfer learning for abstractive text summarization: A review / A. Alomari et al. Computer Speech & Language. 2022. Vol. 71. 43 p.

15. Incorporating copying mechanism in sequence-to-sequence learning / J. Gu et al. 2016. 10 p.

16. Efficient summarization with read-again and copy mechanism / W. Zeng et al. Iclr 2017. 2017. 11 p.

17. See A., Liu P. J., Manning C. D. Get to the point: summarization with pointer-generator networks. 2017. 20 p.

18. Modeling coverage for neural machine translation / Z. Tu et al. Proceedings of the 54th annual meeting of the association for computational linguistics. 2016. P. 76–85.

19. Knowledge enhanced contextual word representations / M. E. Peters et al. Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP). 2019. P. 43–54.

20. Mastronardo C., Tamburini F. Enhancing a text summarization system with ELMo. 2019. 11 p.

21. Efficient estimation of word representations in vector space / T. Mikolov et al. Proceedings of Workshop at ICLR. 2013. 13 p.

22. Krantz J., Kalita J. GloVe: global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014. P. 1532–1543.

23. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin et al. Proceedings of NAACL-HLT 2019. 2019. P. 4171–4186.

24. XLNet: generalized autoregressive pretraining for language understanding / Z. Yang et al. Proceedings of 33rd conference on neural information processing systems (neurips 2019). 2019. P. 5753–5763.

25. ERNIE-GEN: an enhanced multi-flow pre-training and fine-tuning framework for natural language generation / D. Xiao et al. Proceedings of the 29th international joint conference on artificial intelligence. 2021. P. 3997–4003.

26. Language models are unsupervised multitask learners / A. Radford et al. 2019. 24 p.

27. Language models are few-shot learners / T. B. Brown et al. NeurIPS 2020. 2020. 75 p.

28. Improving Language Understanding by Generative Pre-Training / R. Alec et al. 2018. 12 p.

29. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer / C. Raffel et al. Journal of Machine Learning Research. 2020. № 21. P. 1–67.

30. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization / J. Zhang et al. Proceedings of 2020 International Conference on Machine Learning. 2020. № 119. P. 11328–11339.

31. Phang J., Zhao Y., Liu P. J. Investigating efficiently extending transformers for long input summarization. 2022. 128 c.

32. Big Bird: Transformers for Longer Sequences / M. Zaheer et al. Proceedings of 34th Conference on Neural Information Processing Systems (NeurIPS 2020). 2021. № 33. P. 17283–17297.

33. RoBERTa: A robustly optimized BERT pretraining approach / Y. Liu et al. 2019. 13 p.

34. Galactica: a large language model for science / R. Taylor et al. 2022. 58 p.

35. Beltagy I., Peters M. E., Cohan A. Longformer: The Long-Document Transformer. 2020. 17 p.

36. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension / M. Lewis et al. 2019. 10 p.

37. Condevaux C., Harispe S. LSG Attention: Extrapolation of pretrained Transformers to long sequences. 2022. 12 p.

38. Sparks of artificial general intelligence: early experiments with GPT-4 / S. Bubeck et al. 2023. 155 p.

39. Ramírez-Castañeda V. Disadvantages in preparing and publishing scientific papers caused by the dominance of the English language in science: The case of Colombian researchers in biological sciences. PLoS one. 2020. 15 p.

40. Chin-Yew L. ROUGE: A Package for Automatic Evaluation of Summaries. Text Summarization Branches Out, Barcelona, 25–26 July 2004. 2004. P. 74–81.

41. Banerjee S., Lavie A. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization. 2005. P. 65–72.

42. BLEU: a method for automatic evaluation of machine translation / K. Papineni et al. Proceedings of the 40th annual meeting of the association for computational linguistics (ACL). 2002. P. 311–318.

43. Post M. A call for clarity in reporting BLEU scores. Proceedings of the third conference on machine translation: research papers. 2018. P. 186–191.

44. Krantz J., Kalita J. Abstractive summarization using attentive neural techniques. 2018. 9 p.

45. BERTScore: evaluating text generation with BERT / T. Zhang et al. Proceedings of the international conference on learning representations (ICLR) 2020. 2020. 43 p.

46. MoverScore: text generation evaluating with contextualized embeddings and earth mover distance / W. Zhao et al. Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP). 2019. P. 563–578.

47. A discourse-aware attention model for abstractive summarization of long documents / A. Cohan et al. Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: human language technologies, volume 2 (short papers). 2018. P. 615–621.

48. Villanova A. Ccdv/pubmed-summarization datasets at hugging face. Hugging Face – The AI community building the future. URL: https://huggingface.co/datasets/ccdv/pubmed-summarization (date of access: 10.04.2023).

49. Villanova A. Ccdv/arxiv-summarization datasets at hugging face. Hugging Face – The AI community building the future. URL: https://huggingface.co/datasets/ccdv/arxiv-summarization (date of access: 10.04.2023).

50. Villanova A. Scientific_papers dataset at hugging face. Hugging Face – The AI community building the future. URL: https://huggingface.co/datasets/scientific_papers (date of access: 10.04.2023).

51. Yao Lu, Yue Dong, Laurent Charlin. Multi-XScience: a large-scale dataset for extreme multi-document summarization of scientific articles. 2020. 7 c.

52. Leaderboard | SCROLLS Benchmark. Scrolls Benchmark. URL: https://www.scrolls-benchmark.com/leaderboard (date of access: 11.04.2023).

ДОДАТОК А

Відомість кваліфікаційної роботи

Позначення	Найменування	Дод. відомості
	Текстові документи	
1.	Пояснювальна записка	70 с.
	Інші документи	
2.	Презентаційні матеріали	10 плакатів

.	.	Прізвище та ініціали.	Підп.	Дата	
Розробив		Морковський К.Д.			Дослідження методів обробки природної мови для сумаризації тексту наукових публікацій
Перевірив		Узлов Д.Ю.			
Н.контр.		Малєєва І.А.			
Затв.		Філатов В.О.			

Шифр групи	Код напр./спец.
СШІм-21-1	122
ХНУРЕ кафедра ШІ	