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 автоматичного резюмування текстів новин

(тема)

Виконав:
студент 2 курсу, групи СШІм-22-2
Смолярчук С.В.
(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. Турута О.П.
(посада, прізвище, ініціали)

Допускається до захисту

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

2024 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
(повна назва)

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

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Смолярчуку Сергію Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів автоматичного резюмування текстів новин

затверджена наказом університету від 1 квітня 20 24 р. № 260Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13 червня 20 24 р.

3. Вихідні дані до роботи Теоретичне дослідження рішення і опис технологій, робота з потоком даних, математична основа рішення, задача в предметній галузі, практична частина роботи з вибраним алгоритмом, вибір алгоритму машинного навчання, пошук об'єктів на фото, попередня обробка та візуалізація даних для роботи, застосовані технології, програмна реалізація застосунку, процес роботи з потоковим відео та вибір ЄВМ, проектування та тестування

4. Перелік питань, що потрібно опрацювати в роботі
1) Огляд існуючих систем моніторингу та розпізнавання образів
2) Вивчення математичної основи рішення
3) Розробка системи аналізу тексту
4) Тестування ефективності розроблених рішень шляхом багаторазового повтору
5) Оцінка розробленої системи
6) Вивчення прийнятності

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	01.04.2024	виконано
2	Аналіз предметної галузі та постановка задачі	04.04.2024 – 15.04.2024	виконано
3	Аналіз методів щодо вирішення поставленої	15.04.2024 – 20.04.2024	виконано
	задачі		
4	Вибір методу для вирішення задачі	20.04.2024 – 25.04.2024	виконано
5	Розробка базового прототипу	01.05.2024 – 15.05.2024	виконано
6	Підготовка пояснювальної записки	16.05.2024 – 5.06.2024	виконано
7	Захист перед ЕК	13.06.2024	виконано

Дата видачі завдання 1 квітня 20 24 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Турута О.П.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 71 с., 16 рис., 1 табл., 1 дод., 17 джерел.

ГЕНЕРАТИВНІ МОДЕЛІ, ДОВГОТРИВАЛА КОРОТКОЧАСНА ПАМ'ЯТЬ, МЕХАНІЗМ УВАГИ, МОДЕЛЬ КОДЕРА-ДЕКОДЕРА, ОЦІНКА КОРИСТУВАЧА.

Об'єкт дослідження – дослідження методів штучного інтелекту для задачі обробки природомовної інформації.

Предмет дослідження – практична реалізація одного з алгоритмів штучного інтелекту для розв'язання задачі сумаризації тексту для обраної предметної галузі резюмування блоку новин.

Мета роботи – застосування методів штучного інтелекту для розв'язання задачі сумаризації тексту новин.

Методи дослідження – моделювання та експериментальне дослідження в особливих умовах, що створюються особисто для вивчення певних характеристик явлення в задачі розпізнавання тексту.

ABSTRACT

Master’s thesis contains: 71 pp., 16 fig., 1 tabl., 1 ann, 20 references.

ATTENTION MECHANISM, ENCODER-DECODER MODEL, GENERATIVE MODELS, LONG-TERM SHORT-TERM MEMORY, USER EVALUATION.

The object of research is to study artificial intelligence methods for solving the problem of processing natural language information.

The subject of the study is the practical implementation of one of the artificial intelligence algorithms for solving the problem of text summarization for the selected subject area of news summarization.

The purpose of the study is to apply the artificial intelligence methodology to solve the problem of news text summarization.

Research methods – modeling and experimental research in special conditions created personally to study certain features of the phenomenon in the task of text recognition.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЗНМ – згорткова нейронна мережа;

КВ – крос валідація;

ШІ – штучний інтелект;

DL – Deep Learning – глибинне навчання;

DNN – Deep Neural Network – глибока нейронна мережа;

FL – Feature Learning – навчання ознак;

IDF – Inverse Document Frequency – зворотна частота документообігу;

TF – Term Frequency – частота терміну.

ВСТУП

Цю роботу присвячено розв'язанню задачі сумаризації тексту із застосуванням нейронної мережі на прикладі генерації новинних заголовків.

Інформаційне перевантаження, що виникло з поширенням інтернету, зумовило необхідність використання автоматичного опрацювання природної мови та текстів, написаних нею. Особливо це помітно в новинному жанрі, тексти в якому виробляються з великою швидкістю.

Серед інших методів опрацювання тексту вирізняється сумаризація, оскільки вручну таке опрацювання потребує значних зусиль, а автоматизація цього процесу дає змогу заощадити колосальні людські ресурси. Нейронні мережі дали змогу якісно поліпшити результати такої автоматизації, але розв'язаним це завдання визнати не можна, що й зумовлює актуальність цієї роботи. Окремим випадком сумаризації є генерація новинних заголовків із тексту, оскільки заголовок являє собою вичавку зі змісту вихідного тексту новини, коротке повідомлення на її основі.

Об'єктом дослідження є автоматичне реферування новинних текстів.

Предмет дослідження – моделі автоматичного реферування новинних текстів.

Мета цієї роботи – вивчити модель Transfomer до вирішення завдання автоматичного реферування новинних текстів.

Для досягнення мети було поставлено такі завдання:

– на основі вивчення наявних моделей автоматичного реферування текстів обґрунтувати застосування моделі, що ефективно працює;

– застосувати архітектуру Transformer до завдання сумаризації новинних статей.

Теоретичними джерелами дослідження послугували роботи, присвячені як проблемам комп'ютерної лінгвістики загалом, так і автоматичної сумаризації зокрема. Це роботи Д. Журафскі [14] з опрацювання природної мови, Х. П. Луна [22], Т. Едмундсона [10] і Д. Радєва [32] з реферування, Ч. Й. Ліна [19] з оцінювання якості автоматичних рефератів. Крім них, теоретичним підґрунтям послугували й сучасні роботи із застосування нейронних мереж до задачі сумаризації – роботи К. Лопірева [21], А. Васвані [45] і Д. Бахданау [2].

Ці дослідження допомогли засвоїти передові методи та технології у цій області та використовувати їх у власних експериментах.

Матеріалом для дослідження були використані текстові дані з новинних корпусів відомих інформаційних агентств, зокрема RBK.ua та Ukraina Today. Ці корпуси містять значну кількість новинних статей, що дозволило отримати репрезентативні дані для проведення експериментів та оцінки ефективності розроблених моделей.

Теоретична значущість дослідження полягає в унікальному внеску в розвиток методів генерації та сумаризації новинних статей. Отримані результати та використані методи можуть стати основою для подальших досліджень у цій області та застосовуватися для вирішення схожих завдань з обробки природної мови.

Практична значущість дослідження виявляється в можливості впровадження розроблених методів у проєкти, присвячені автоматичній генерації новин та реферуванню текстів. Розроблена система не лише вирішує завдання автоматичної сумаризації, а й може бути використана для реферування будь-яких текстів, що відкриває нові можливості для автоматизації процесів обробки інформації та підвищення продуктивності відповідних проектів.

1 СТАНОВЛЕННЯ ТА СУЧАСНІ МЕТОДИ АВТОМАТИЧНОЇ СУМАРИЗАЦІЇ

1.1 Визначення сумаризації

У своїй роботі 2002 року Радев [32] визначає реферат як текст, вироблений з одного або кількох текстів, що містить важливу інформацію з цих оригінальних текстів і довжина якого не перевищує половини оригінального тексту, а зазвичай значно менша. Виходячи з цього визначення, що містить у собі найважливіші аспекти автоматичної сумаризації, можна прийняти, що автоматична генерація заголовка новини є окремим випадком генерації реферату тексту, оскільки формально вона відповідає даному вище визначенню.

Одна з класифікацій [32], застосованих до методів сумаризації, поділяє їх на екстрактивні та абстрактні методи. Перші ґрунтуються на виокремленні з текстів найважливіших речень, тоді як другі мають важливу відмінність – вони включають у себе метод генерації тексту, оскільки генерують для реферату нові речення. Крім них виокремлюють змішування (fusion) і стискання (compression). І змішання, і компресію можна вважати окремим випадком екстрактивного методу, оскільки змішання являє собою комбінацію витягнутих частин речень, а компресія – видалення непотрібних частин тексту.

Описуючи далі методи сумаризації тексту, ми припускаємо, що всі вони можуть бути використані для аналізу новинних текстів.

Ймовірнісні тематичні моделі здійснюють «м'яку» кластеризацію, дозволяючи документу або терміну відноситися відразу до декількох тем з різними ймовірностями. Імовірнісні тематичні моделі описують кожну тему дискретним розподілом на множині термінів, кожен документ дискретним розподілом на множині тем.

1.2 Ранні алгоритми реферування

Перші роботи з автоматичного складання рефератів можуть здатися далекими від задачі генерації заголовка – у них розглянуто тільки завдання екстрактивного реферування технічних документів. На відміну від абстрактної сумаризації, яка включає в себе генерацію тексту, вона передбачає, що реферат буде складено шляхом виключення зайвої або менш важливої інформації. Однак багато ідей, запропонованих у ранніх статтях, дотепер успішно використовуються не тільки для розв'язання задачі автоматичного реферування, а й у комп'ютерній лінгвістиці загалом.

У 1958 році дослідник з IBM Ганс Петер Лун опублікував статтю, в якій було описано дослідження сумаризації технічних документів, проведене в компанії. У своїй роботі Лун пропонує вважати мірою важливості слова частоту його вживання, а на підставі цієї міри обчислювати важливість кожного речення, що, своєю чергою, дасть змогу обрати найбільш значущі речення для реферату. Особливий інтерес у цій статті, яка згодом стала однією з найбільш цитованих, становить низка речень і зауважень. Стоп-слова, будучи часто повторюваними і необхідними для розуміння тексту, не є необхідними для розуміння конкретного тексту; їх можна і потрібно виключити з тексту перед підрахунком важливості слів за методом Луна. Слова, що залишилися, мають бути приведені до початкової форми і лематизовані. Далі список змістовних слів сортується від найбільш значущих до менш значущих. Для кожного речення на основі кількості входжень лем зі списку та лінійної відстані між ними обчислюється свій індекс значущості, за яким далі сортуються речення. Нарешті, речення з найвищим індексом обираються для реферату.

У роботі Баксендейла [3], опублікованій того ж року і теж заснованій на дослідженнях IBM, описано результати аналізу 200 текстів, проведеного з метою з'ясувати, як впливає позиція речення на його значущість. У 85%

випадків найбільш значущим виявлялося перше речення, у 7% – останнє, з чого було зроблено висновок про те, що позиція речення в тексті є важливою ознакою значущості речення. Положення слова і речення в тексті дотепер використовується в алгоритмах сумаризації – у системи, засновані на машинному навчанні, позиція включена як ознака.

Пізніше, 1969 року, Едмундсон описав створення сумаризатора і поставлений експеримент з екстрактивного реферування, структура якого пізніше стала типовою для подібних досліджень. Автор розробив протокол для створення рефератів вручну, який було застосовано до 400 документів. Великим внеском у подальші дослідження стали нові ознаки, використані в дослідженні, крім описаних раніше частоти та позиції – наявність ключових слів і структура документа; ваги кожній ознаці призначалися вручну. У результаті експерименту було виявлено, що 44% рефератів, створених сумаризатором Едмундсона, збігаються з тими, що були написані вручну.

1.3 Ранні методи машинного навчання

У міру розвитку методів машинного навчання, вони набули широкого застосування в розв'язанні задач комп'ютерної лінгвістики, зокрема й щодо завдання екстракції реферату з тексту. Незважаючи на те, що саме по собі машинне навчання було новою технологією, яка вже істотно змінювала процес вибору речень для реферату, тривало вивчення впливу вибору ознак на якість екстракції. Однак єдиної думки з приводу залежності ознак одна від одної не було, тому такі дослідження проводили в рамках застосування як наївного Байєсівського класифікатора, так і складніших моделей, наприклад, дерева рішень.

У 1995 році Купієць [16] описав метод, який спирається на метод Едмундсона. За своєю суттю метод схожий скоріше на класифікацію – система класифікує речення на ті, що будуть включені в реферат, і ті, що

будуть виключені, використовуючи найнаївніший Байєсівський класифікатор. Нехай s – це речення, S – множина речень реферату, і F_1 , ..., F_k – ознаки.

Припускаючи, що ознаки незалежні, можемо записати таку формулу:

$$P(s \in S|F_1, F_2, .. F_k) = \frac{\prod_{i=1}^{k} P(F_i|s \in S) * P(s \in S)}{\prod_{i=1}^{k} P(F_i)}. \quad (1.1)$$

Ознаки, на яких було навчено сумаризатор, відповідали ознакам у згаданій вище роботі Едмундсона, але додатково до них було використано довжину речення та наявність слів, що починаються з великої літери. Після обчислення балів, для реферату було обрано n речень із найвищими балами. Для оцінки якості роботи системи було використано корпус технічних документів із рефератами, створеними людиною. Кожен із рефератів був розмічений таким чином: кожне речення було позначено залежно від того, чи відповідає воно реченню з тексту. Окремо позначалися речення, що відповідали одразу кільком реченням. Реферати, створені автоматично, були оцінені в порівнянні з розміченим матеріалом. Аналіз ознак виявив, що система, навчена на ознаках позиції, довжини речення, а також ознаках із моделі Едмундсона, показала найкращі результати.

Як і Купієць, Еон [1] використовував наївний Байєсівський класифікатор, проте більш насичений ознаками. Описана у відповідній роботі 1999 року система під назвою DimSum використовувала ознаки, які сьогодні відомі як tf та idf – частота терміна (term frequency) та зворотна частота документа (inverse document frequency), отримана з корпусу текстів на схожу тему, щоб вивести список слів, що позначають ключові концепти в тексті. Знайдені статистично найуживаніші словосполучення використовувалися нарівні з окремими словами, тобто за своєю суттю були окремим токеном. Суттєвим просуванням була розмітка іменованих

сутностей, що дало змогу зв'язати між собою різні назви для однієї сутності, синоніми та розшифровки абревіатур.

Серйозніший аналіз ознаки позиції слова в тексті був проведений дослідниками Ліном і Гові й описаний у роботі 1997 року [17]. Автори виходять із припущення, що тексти слідують за передбачуваною структурою і, відповідно, найбільш значущими реченнями в тексті є ті, що знаходяться в певних частинах тексту, наприклад, у підзаголовках, анотаціях і так далі. Однак, зауважують автори, структура тексту різниться для різних жанрів, тому необхідно переглянути метод вибору найбільш значущих речень на підставі речення в тексті з роботи Баксендейла. Ця робота робить значний внесок у дослідження застосування позиційного методу до різних жанрів і дискурсів.

Для вивчення було використано корпус TIPSTER, що складається з текстів про комп'ютери. Для кожного тексту подано невеликий реферат, написаний людиною, а також кілька ключових слів. Для кожного документа автори виміряли загальний вміст ключових слів у кожному реченні, щоб виробити так звану Optimal Position Policy (OPP), схему розташування найважливіших речень для даного жанру. Для оцінки якості такого підходу було проведено два експерименти. У першому експерименті процедуру з вироблення OPP було проведено на нових текстах цього ж жанру, унаслідок чого було отримано схему, аналогічну обчисленій вручну. У другому експерименті було підраховано перетин рефератів, створених людьми, з результатом роботи автоматичного сумаризатора; високий ступінь перетину показав ефективність методу.

У своїй подальшій роботі 1999 року Лін пішов [18] від припущення, що ознаки слід вважати незалежними. Для доведення взаємозалежності Лін використовував іншу модель машинного навчання, дерево рішень, і провів випробування з різними наборами ознак – як з комбінацією всіх введених ознак, так і тільки з ознакою позиції. Крім ознак, включених до більш ранніх

робіт, було введено велику кількість нових ознак, описаних у роботі Еона 1999 року: це, здебільшого, ознаки типу Boolean, що містять одиницю, якщо речення містить число, прикметник, лапки, день тижня, місяць тощо.

Експерименти продемонстрували, що для деяких тем наївний класифікатор працює краще, але для всього датасету, складеного з текстів TRIPSTER-SUMMAC, саме дерево рішень показало найкращі результати. Лін припускає, що так сталося через незалежність окремих ознак. Аналіз результатів показав, що оцінка, дана на основі кількості згадок ключових концептів у тексті (IR Signature), виявилася особливо значущою, що підтвердило ранні припущення Луна.

На противагу попереднім підходам, що ґрунтувалися переважно на аналізі ознак, Конрой та О'Лірі висловили [8] припущення, що слід використовувати приховану марковську модель, оскільки саме послідовна модель може допомогти знайти локальні зв'язки між реченнями. Використаними ознаками стали тільки позиція речення в документі, кількість термінів у реченні та вірогідність речення, ґрунтуючись на термінах усього тексту (рисунок 1.1).

Структура моделі містить 2s+1 станів: s summary-станів і s+1 nonsummary-станів.

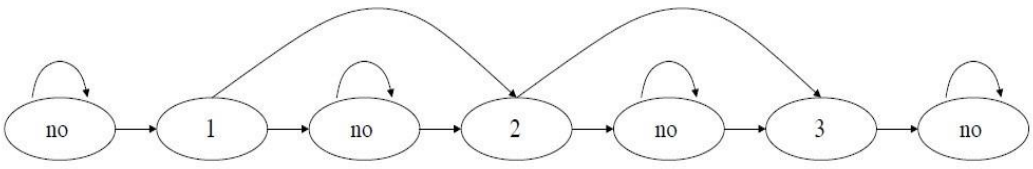


Рисунок 1.1 – Схема прихованої марковської моделі для трьох пропозицій

Використовуючи вибірку TREC [28], автори обчислили ймовірність переходу зі стану в стан, сформувавши матрицю переходів; кожному стану і в рядку відповідає функція виведення:

$$b_i = P(0|i), \quad (1.2)$$

де O – вектор ознак (для спрощення прийнято припущення, що ознаки розподілені нормально). Експеримент виявив успішність підходу.

Незважаючи на порівняно невеликі потужності обчислень, проводилися спроби використання нейронних мереж – у 2007 році Своуром [44] було здійснено тренування моделі нейронної мережі, що здійснювала ранжування пропозицій на основі безлічі введених ознак. Були введені ознаки, які раніше не використовувалися: схожість речення з реченням із корпусу ROUGE-1, що містив реферати. Ранг речення в тексті підвищувала і наявність термінів і n-грам, що зустрічаються в пошуковому двигуні від Microsoft і Вікіпедії. Подальше оцінювання якості роботи показало ефективність методу.

Ще раз перелічимо особливості, переваги та недоліки методів, проводячи зв'язок від сумаризатора Х. П. Луна до нейронних мереж.

Усі перелічені вище роботи є роботами з екстрактивної сумаризації, що якісно відрізняється від абстрактивної, оскільки не містить завдання генерації тексту, а лише класифікує речення за важливістю, як максимум – комбінує частини речень, не втручаючись у їхню структуру. Припущення про те, що важливість того чи іншого речення залежить від його змісту, приводить до думки про необхідність вироблення ознак на основі частоти вживання ключових концепцій у реченні. Подальші роботи додають до цієї моделі інші ознаки: на основі статистичних досліджень виявляється важливість позиції речення в тексті. Паралельно відбувається процес вироблення стандартів оцінювання рефератів, однак брак обчислювальних потужностей зумовлює невеликий розмір корпусів для оцінювання – усі реферати в таких корпусах проходять оцінювання вручну. З появою більших обчислювальних можливостей набувають поширення методи машинного навчання. Це дає змогу ввести більшу кількість ознак в обчислення.

Використовуються прості статистичні моделі, як-от найнаївніший Байєсівський метод, проте висловлюються й припущення про взаємозалежність ознак. Для доведення цієї думки було використано складнішу модель – дерево рішень, експерименти з якою призвели до висновку про часткову незалежність ознак та підтвердили припущення Луна про важливість ознаки, що ґрунтується на наявності ключових концептів у тексті. Важливість концепції послідовності та пошуку послідовного зв'язку речень у тексті виявило використання прихованої Марковської моделі, що підводить до думки про можливість використання нейронних мереж, оскільки ті добре працюють із послідовностями, що за роки знайшло відображення в концепції моделі Seq2seq. Перші роботи з використання нейронних мереж запропонували ознаки, засновані на оцінці важливості термінів залежно від знаходження їхніх визначень у Вікіпедії. Цю ознаку можна вважати дуже простим прообразом майбутнього переднавчання складніших сучасних моделей на корпусі Вікіпедії для вироблення мовних уявлень.

Таким чином, на прикладі згаданих вище автоматичних сумаризаторів ми можемо спостерігати еволюцію методів і підходів до реферування в період до широкого поширення нейронних мереж.

1.4 Сучасні методи машинного навчання

Сьогодні машинне навчання насамперед асоціюється з нейронними мережами: обчислювальні потужності, що зросли, дали змогу ускладнити архітектуру моделей машинного навчання, а кількість параметрів збільшити до десятків і сотень тисяч. Бум нейромереж, що стався, призвів до того, що екстрактивна сумаризація майже цілком змінилася на абстрактну, тобто, до завдання реферування було включено завдання генерації тексту природною

мовою. Однак деякі роботи повертаються до екстракції і навіть комбінують підходи, як, наприклад, розглянута далі модель Pointer-Generator.

1.4.1 Seq2seq

У 2015 році чи не першим дослідником, який застосував нейронну мережу до задачі абстрактивної сумаризації, став Раш [36]. Попередні роботи з абстрактивного реферування спиралися передусім на лінгвістичні обмеження та синтаксичні трансформації, тоді як ця робота пропонує підхід виключно математичний.

Концепція Seq2seq розвиває висловлену раніше думку про необхідність сприйняття тексту насамперед як послідовності даних. Не всі нейронні мережі здатні сприймати контекст і обробляти інформацію з урахуванням положення слів відносно одна одної. В основі архітектури Sequence to sequence лежить використання рекурентних блоків, що містять цикл, який дає змогу зберігати пам'ять про попередні частини тексту під час опрацювання нових.

Дослідники провели експерименти з трьома різними кодувальниками, серед яких кодувальник, що базується на увазі, показав найкращі результати. Проведені експерименти були обмежені генерацією заголовків на основі першого речення в тексті.

Ця модель, утім, була значно поліпшена Наллапаті у 2016 році [27]. У своїй роботі він використовує рекурентну нейронну мережу, в якій не тільки кодувальник, а й декодувальник заснований на механізмі уваги. Важливим доповненням до архітектури стала велика кількість ознак – текст було розмежовано за частинами мови, було виділено іменовані сутності, а також було застосовано частоту слова і зворотну частоту документа. Незнайомі нейромережі слова, тобто слова, яких немає в датасеті, на якому навчена модель, копіюються за допомогою генератора покажчиків. Крім цього

чималого внеску в подальші дослідження, дослідники створили новий датасет із новин CNN і Daily Mail, що містить понад 286 тисяч тренувальних пар заголовок-реферат, а також додаткові набори для тестів і валідації. Крім звичайного корпусу було створено «анонімний» корпус, де з метою розмітки іменованих сутностей вони були замінені на токени з індвидуальним ідентифікатором.

1.4.2 Механізми доповнення Seq2seq

Модель виду кодувальник-декодувальник доволі проста, і тому велика кількість досліджень використовує різноманітні модифікації та доповнення. Імплементувати ці доповнення та зміни можна на одному з трьох етапів – на вході, на виході та у прихованому шарі.

На вході поліпшення відбуваються насамперед за допомогою якісної роботи дослідника з текстовими даними – збору об'ємного репрезентативного та різноманітного корпусу, очищення даних, вибору відповідного алгоритму векторизації тексту. За допомогою правильно обраного алгоритму векторизації тексту можна уникнути великої кількості проблем, як-от неможливість опрацювання рідкісних термінів і слів поза словником моделі та брак оперативної пам'яті під час навчання. Так, алгоритми векторизації на основі підрядків добре опрацьовують невідомі моделі словоформи, а алгоритми на кшталт Word2Vec, що прийшли на заміну One-Hot кодуванню, займають значно менше місця [26].

Механізм копіювання також можна вважати успішним у завданні обробки out-of-vocabulary токенів. Такий механізм, якщо розглядати його на прикладі завдання сумаризації, просто переносить токен із тексту в реферат.

Проблему перенавчання моделі вирішує dropout – цей механізм нібито проріджує нейрони, що особливо важливо для рекурентних нейронних мереж, оскільки змінює структуру нейромережі на кожному кроці, що

робить нейрони, що залишилися, менш адаптованими до конкретних даних [41]. Для розв'язання труднощів із симетричністю мережі, що виникають під час застосування дропаута, використовується спеціальний його різновид – recurrent dropout.

Одним із найважливіших механізмів, що з'явилися для поліпшення роботи архітектури Seq2seq, є механізм уваги, створений з метою розв'язання проблеми втрати контексту моделлю під час опрацювання довгих речень. На відміну від звичайного підходу, за якого довге речення зводиться тільки до одного прихованого стану, механізм уваги дає змогу співвідносити вектор контексту з усім, що надходить на вхід. Існує не один варіант механізму уваги: multi-head self-attention, global, local, hard, soft тощо. На основі механізму уваги побудовано широко популярну архітектуру Transformer, яка є стандартом для сумаризації в наші дні.

1.4.3 Сімейство моделей Transformer

Серед інших підходів значно виділяються моделі, створені на основі архітектури Transformer, запропонованої 2017 року лабораторією Google. Істотна їхня відмінність від запропонованих раніше моделей Seq2seq, доповнених механізмом уваги, що забезпечило їм більшу популярність, – відмова від рекурентних і згортальних блоків, що дає змогу запускати дані на вхід не в певному порядку. Це, своєю чергою, означає, що навчатися модель може значно швидше завдяки розпаралелюванню обробки послідовностей.

Одна з широко розповсюджених модифікацій моделі Transformer BERT (Bidirectional Encoder Representations from Transformers), також створена в Google і презентована 2018 року [9]. Окрім заявленої в назві двоспрямованості, яка дає змогу інформації проходити назад до входу, що покращує якість роботи мережі, ця модель чутлива до контексту. Таким

чином, така модель добре працює з омографами, створюючи для них значно відмінні вектори.

BERT, як і інші моделі архітектури Transformer, використовується насамперед у попередньо навченому вигляді, а згодом проходить донавчання на специфічних для завдання вибірках. Саме такий підхід до використання цієї моделі було запропоновано в роботі 2018 року авторами лабораторії OpenAI. У роботі виокремлено важливі проблеми звичайного підходу до навчання нейронних мереж: це неможливість перевикористання моделей, навчених для певної задачі, і потреба у великій кількості даних, що найчастіше вимагає великих ресурсів. GPT, представлена в цій роботі, була попередньо навчена на BooksCorpus, і згодом донавчена для специфічних завдань; для 9 із 12 завдань результат перевершив поточні state-of-art роботи.

1.4.4 OpenAI і сімейство моделей GPT

GPT GPT-2, що прийшла на зміну GPT, була натренована на 40 гігабайтах текстів з інтернету (8 мільйонів сторінок) і містила 1,5 мільярда параметрів, що приблизно в 10 разів перевершує показники GPT. Пріоритетом при створенні датасету була не тільки кількість даних, а й їхня якість: наприклад, активно використовували посилання на сторінки з Reddit, дописи з якими отримували понад три очки «карми» (аналог лайка на сайті), що автори пропонують як показник того, що користувачі скоріше знаходять посилання корисним і важливим, а отже, і текст у ньому досить якісний.

Завдяки мовним уявленням, побудованим на багатій і різноманітній вибірці, і навчанню, що не передбачає специфічної задачі, модель винятково добре адаптується до стилю тексту на вході, не потребуючи донавчання для отримання досить якісних результатів. Так, на сайті OpenAI наводиться, серед інших, приклад генерації тексту новини при подачі на вхід двох

речень; за своєю суттю таке завдання є зворотним до сумаризації, рисунок 1.2.

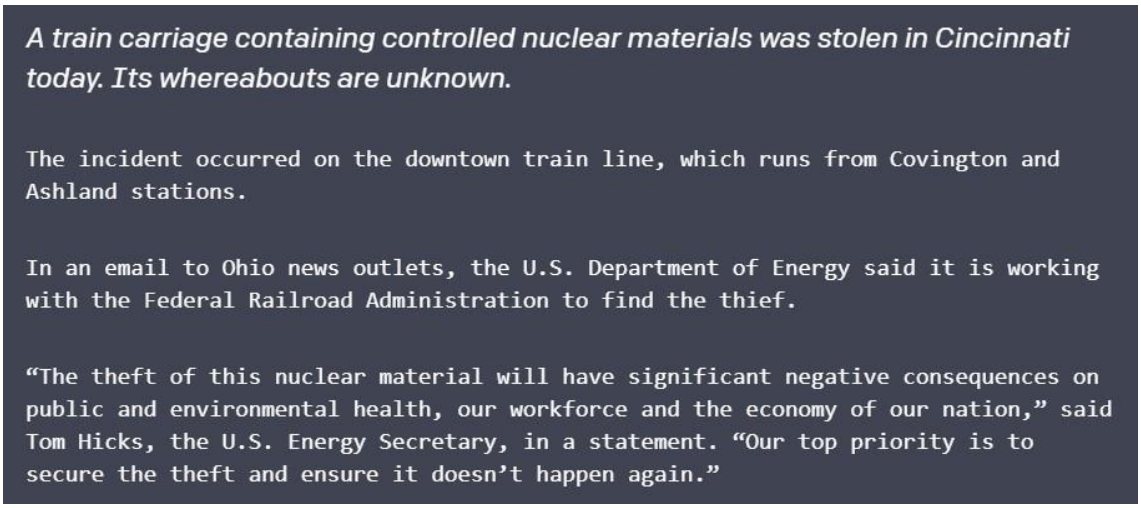


Рисунок 1.2 – Приклад тексту, згенерованого GPT-2

При цьому, варто зазначити, хоча ця вибірка дає змогу покрити більшу частину популярних іменованих сутностей, як і всі аналогічні моделі, GPT- 2 часто зазнає труднощів при введенні на вхід тексту про невідому іменовану сутність або специфічне явище. У завданні генерації заголовків новин це є особливо важливою проблемою, оскільки найпоширеніші в новинах імена і назви, зібрані в період збору корпусу і навчання моделі, можуть зовсім не зустрічатися в період її використання.

GPT-3, випущена 2020 року, значно перевершує будь-яку випущену раніше модель за кількістю параметрів – 175 мільярдів. Такий обсяг дає змогу застосувати концепцію few-shots learning, яка передбачає, що модель може, подібно до людини, навчатися виконувати те чи інше завдання виключно взаємодіючи з людиною, яка пояснює інструкцію до завдання за допомогою тексту. Тому під час проведення експериментів та оцінювання не проводилося донавчання для специфікації завдання. Такий підхід дає

змогу зробити модель універсальнішою і позбавляє розробників необхідності шукати додаткові розмічені дані [4].

Незважаючи на очевидний успіх, розробники помічають, що навіть у такої моделі є свої обмеження і недоліки. На найбільш вузьких завданнях якість роботи такої моделі не перевершує якість роботи більш простих, але донавчених систем. Модель навчена тільки англійською мовою, що не скасовує можливості виробництва аналогічних моделей для окремих мов, але є перешкодою для максимально широкого поширення. Як і в багатьох системах, побудованих на основі глибокого навчання, рішення GPT-3 погано піддаються інтерпретації, що віддаляє нейронні мережі від їхньої первісної ідеї – побудови системи на подобу людського мозку.

Окрім дослідження практичних властивостей моделі та її можливостей, автори здійснили детальний аналіз, що стосується низки непрямих проблем, пов'язаних із використанням моделі, таких як упередженість щодо гендеру, раси та релігії, шкідливе використання та обман з її допомогою. Хоча автори не пропонують конкретних рішень, це одна з перших робіт, що порушує ці неочевидні питання.

1.4.5 Екстрактивні підходи

Робота 2019 року Лью демонструє видатний результат у завданні екстрактивної сумаризації, для чого було використано переднавчену модель BERT [20]. Для того, щоб отримати реферат, до моделі було додано додатковий шар. У роботі Лью порівнює різні архітектури для цього додаткового шару – простий класифікатор, рекурентну нейронну мережу та двошаровий inter-sentence transformer, і остання дає найкращий результат.

Комбінацією екстрактивного та абстрактивного підходів до задачі сумаризації можна назвати модель Pointer-Generator, що використовує покажчик для копіювання конкретних слів, і показник coverage, що

допомагає нейронній мережі відстежувати, що вже було включено до реферату [37]. Такий підхід обходить багато моделей при вимірюванні метрикою ROUGE, однак має свої недоліки, притаманні всім екстрактивним підходам, – нечутливість до контексту та неможливість виникнення синонімів і переформулювання речень.

1.4.6 Моделі для текстів українською мовою мовою

Серед моделей, створених для генерації текстів українською мовою, можна відзначити моделі, створені в процесі змагання з генерації заголовків, що відбувалося в рамках конференції Діалог у 2019 році [24]. У підсумковій роботі, що описує результати змагання, було описано результати робіт чотирьох команд. Три роботи представляли процес створення роботи «з нуля», без використання попередньо вивчених моделей, кожна використовувала оригінальні архітектурні рішення, четверта використовувала механізм поліпшення введення без істотних архітектурних доповнень. Усі моделі були описані у відповідних публікаціях.

Перша з моделей заснована на Phrase-Based Attentional Transformer, цю архітектуру було представлено 2018 року [40], і вона базується на згортках ядер для механізму уваги:

$$QueryK(Q,K,V) = \left(\frac{Conv_n(KW_k, QW_q)}{\sqrt{d_n * n}}\right) Conv_n(V, W_n). \qquad (1.3)$$

Автори роботи стверджують, що досягли state-of-the-art результату на вибірці «Україна Сьогодні», заснованій на матеріалах цього новинного агентства.

Наступна модель використовує у своїй роботі механізм копіювання CopyNet. Ідею цього механізму описано вище, нижче наведено схему цієї

архітектури з дослідження [13], присвяченого цьому механізму. Важливо зазначити, що для виведення скопійованих слів у моделі є додатковий висновок. Для навчання було використано вибірку, зібрану з ресурсу ICTV.ua [21], (рисунок 1.3).

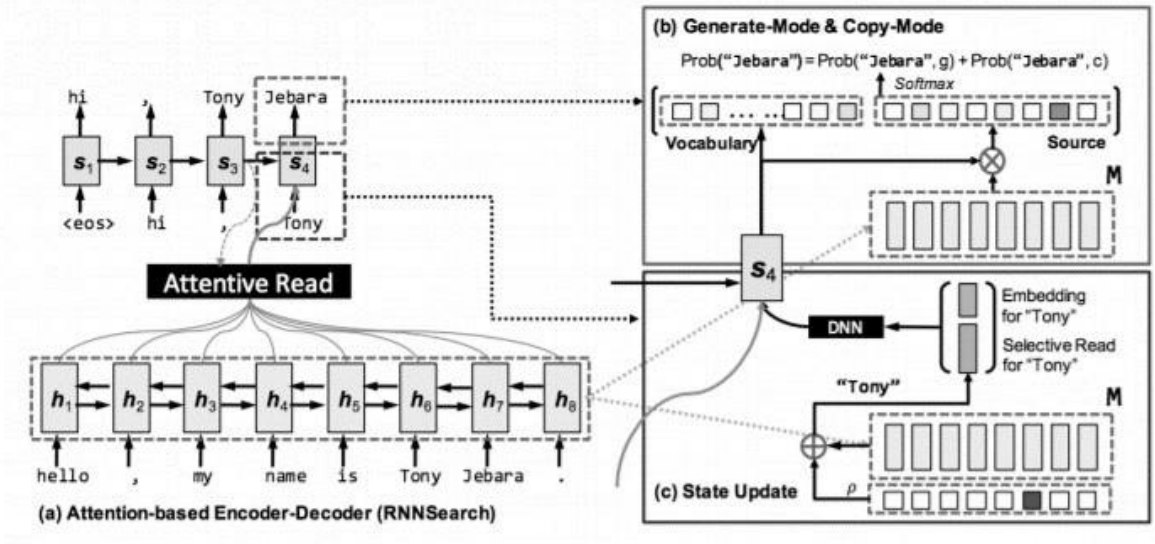


Рисунок 1.3 – Схема архітектури нейронної мережі з використанням механізму копіювання

Було представлено й модель, що використовує схожий з CopyNet механізм – Pointer Generator. Однак ця імплементація моделі була доповнена використанням грамем і особливими структурами-ембеддінгами для леми і флексії. Використовуючи слово на введенні, модель передбачає для нього грамему. Незважаючи на те, що отриманий результат показав, що класична модель показує кращі метрики, підхід визнаний авторами як такий, що виправдав очікування [42].

На відміну від трьох досліджень, описаних вище, четверте не використовує оригінальну архітектуру, а спирається на переднавчені векторно-семантичні моделі [6]. Автори використовували технологію Byte-

Pair Encoding, яка працює на основі підрядків, що особливо ефективно для переднавчання. Як матеріал використовується текст, зібраний із Вікіпедії.

Результати змагання наведено в таблиці 1.1, вони отримані в процесі тестування моделі на прихованому датасеті. Саме на ці результати ми орієнтуємося в процесі виконання цієї роботи.

Таблиця 1.1 – Результати оцінювання якості сумаризації у змаганні в рамках конференції Діалог-2019

Архітектура	Матеріал	Оцінювання якості сумаризації
Фраза-орієнтована на уважність Трансформер	"Україна Сьогодні"	20.267
CopyNet	RBK.ua	23.142
Генератор покажчиків	RBK.ua	20.293
BPE	Вікіпедія	20.268

Оцінювання якості роботи проводили за такою формулою:

$$score(r,h) = \frac{1}{3N}\sum_{i=1}^{n}(ROGUE-1(r_i,h_1) + ROGUE-2\,(r_i,h_1) + ROGUE-L(r_i,h_1), \quad (1.4)$$

де r і h – реферати, складені людиною та автоматичним сумаризатором відповідно.

Хоча тематичне моделювання традиційно описувалося і застосовувалося в обробці природної мови, воно знайшло своє застосування і в інших областях, наприклад, таких як біоінформатика.

1.5 Оцінювання якості роботи алгоритмів сумаризації

Незважаючи на те, що ранні роботи із сумаризації оцінювали за допомогою рефератів, складених вручну, такі реферати, очевидно, не є об'єктивним заходом для оцінювання якості. Навіть у разі екстрактивної сумаризації не може бути одного ідеального реферату, створеного людиною; рівень згоди між укладачами рефератів низький. Крім цього, цей метод складання вибірок для оцінювання надзвичайно витратний – Лін у праці 2004 року зазначає [19], що ручне оцінювання вибірки рефератів масштабу конференції DUC забрало б понад 3000 годин людської роботи.

1.5.1 ROUGE

У 2004 році Лін представив набір метрик під назвою Recall-Oriented Understudy for Gisting Evaluation (ROUGE), які стали стандартом для оцінювання якості результатів сумаризації.

Як стає зрозуміло з назви, ці метрики засновані на повноті; припустимо, що $R = \{r\ ,\ .\ ,\ r\ \}$ – набір рефератів-зразків, а s – реферат, згенерований деякою системою. Нехай $n(d)$ – бінарний вектор, що представляє n-грами, які містяться в документі d, таким чином – якщо i-а n-грама міститься в документі d, $\varphi\ \ d$ дорівнює 1. Тоді метрику ROUGE-N можна обчислити таким чином:

$$ROGUE - N = \frac{\sum_{r \in R} \langle \varphi_n(r), \varphi_n(s) \rangle}{\sum_{r \in R} \langle \varphi_n(r), \varphi_n(r) \rangle}. \quad (1.5)$$

Інша метрика з набору застосовує концепцію найдовшого підрядка. Попередня міра ґрунтувалася на припущенні, що кількість n-грам, що збігаються, відображає схожість автоматичного і ручного рефератів. Ця

приймає за таку мірку загальну послідовність – чим вона довша, тим більше схожі між собою реферати.

Незважаючи на назву й орієнтованість на recall, існує й використовується варіант ROUGE-precision, що розраховується схожим на ROUGE-n чином, однак, на відміну від останньої, у знаменнику не кількість n-грамів у рефераті, створеному людиною, а кількість n-грамів у моделі.

Крім цього, існує ROUGE-L, який визначається як F-міра, заснована на найдовшому підрядку:

$$ROGUE - L = \frac{(1+\beta^2)R_{LCS}P_{LCS}}{R_{LCS}+\beta^2 P_{LCS}}, \quad (1.6)$$

де R_{LCS} визначається як:

$$R_{LCS}(S) = \frac{\sum_{i=1}^{u} LCS(r_i,s)}{\sum_{i=1}^{u} r_i}, \quad (1.7)$$

а P як:

$$P_{LCS}(S) = \frac{\sum_{i=1}^{u} LCS(r_i,s)}{|s|}. \quad (1.8)$$

Незважаючи на очевидні проблеми обох метрик, наприклад, неможливість враховувати синоніми в процесі оцінювання схожості, ROUGE-L частково розв'язує проблему віддавання переваги ROUGE-N коротшим згенерованим послідовностям.

1.5.2 BLEU

Попри те, що ROUGE був вигаданий спеціально для сумаризації та є основною метрикою для оцінювання якості, разом із ним часто

використовують вигадану раніше для оцінювання якості перекладів метрику BLEU (bilingual evaluation understudy), що ґрунтується на precision [29]. Причиною цього є низка переваг. Так, один із важливих компонентів BLEU – brevity penalty. Ця змінна, коефіцієнт у формулі, дорівнює 1, якщо переклад (або, у разі сумаризації, реферат), дорівнює за довжиною або довший за переклад, створений людиною (reference translation), і тим менший, чим коротший переклад.

Окрім цього, згідно з дослідженням 2015 року [11], автори якого порівняли BLEU та 192 варіанти ROUGE, саме BLEU найкраще корелює з людською оцінкою, хоча й не перевершує найкращі з варіантів ROUGE значно.

Зрештою, можливе використання і ROUGE, і BLEU, як як самостійних метрик, так і як змінних в іншій формулі, наприклад, F1-score.

Завдання генерації заголовків еквівалентне завданню сумаризації тексту, що дає змогу розглядати роботи з реферування як такі, що передують цьому дослідженню. Різноманітність методів і отримані в роботах результати дають змогу зробити висновок, що проблема пошуку оптимальної технології вилучення реферату, як і раніше, залишається актуальною.

Класифікації поділяють сумаризацію на екстрактивну й абстрактну. Система, описана в цьому дослідженні, здійснює абстрактну сумаризацію, що означає, що в завдання входить також і генерація тексту природною мовою.

Роботи, описані в розділах, підрозділяють сумаризатори на засновані на евристиках і засновані на машинному навчанні, а останні включають у себе нейронні мережі. До числа нейронних мереж входить модель Encoder-Decoder, вона ж Seq2Seq. Ця робота описує нейронну мережу, побудовану на основі цієї архітектури. Механізм уваги, яким доповнено нейромережу, розв'язує проблеми обробки довгих речень.

2 НЕЙРОННІ МЕРЕЖІ СТОСОВНО ЗАВДАННЯ СУМАРИЗАЦІЇ НОВИННИХ СТАТЕЙ

2.1 Особливості роботи нейронних мереж із текстовими даними

У розділі 1 було подано огляд автоматичних сумаризаторів від ранніх алгоритмів до сучасних моделей нейронних мереж складних архітектур. Однак ми розглядали різновиди автоматичних сумаризаторів як класичну «чорну скриньку», зазначаючи тільки ті деталі будови, що були нововведенням для свого часу і відрізняли архітектури одна від одної. У цьому розділі ми зосередимося на еволюції конкретної обраної нами моделі та докладніше розглянемо процес опрацювання інформації від тексту статті на вході в модель до тексту заголовка на виході.

Для того, щоб розкрити «чорну скриньку», слід дати визначення нейронної мережі, яке використовуватиметься далі для опису її роботи. Штучна нейронна мережа – це універсальний апроксиматор [46]. Інакше кажучи, це математична модель, що розв'язує задачу знаходження деякої закономірності, що виражається функцією, якій підкоряються дані.

Ідея опрацювання природної мови за допомогою математичних моделей деякий час була неочевидна, оскільки завдання нелінійної оптимізації передбачає обчислення і, відповідно, набір числових даних. Однак нейронні мережі сьогодні працюють з будь-якими типами даних – виконують роботу з розпізнавання зображень, поліпшення якості відео, нарешті, з розглянутої нами сумаризації тексту. Це стає можливим при знаходженні способу представлення оброблюваних нейромережею даних у чисельному вигляді, а саме у векторному, матричному або тензорному.

Наразі існує велика кількість способів переведення текстової послідовності в числову, однак до цього необхідно провести попередню обробку тексту і полегшити подальшу роботу нейронної мережі,

визначивши, яких даних ми можемо позбутися, не втративши цінних властивостей усього тексту. Це попереднє опрацювання не залежить від обраного способу подальшої векторизації та відрізняється залежно від розв'язуваної задачі, а точніше від тексту, що має бути отриманий на виході.

Так, для задачі абстрактивної сумаризації не має сенсу проводити лематизацію тексту і видалення стоп-слів, оскільки необхідно забезпечити зв'язність тексту, а вона досягається, зокрема, і використанням словоформ і службових слів. Для простих завдань, як-от класифікація тексту за темами або сентимент-аналіз, лематизація навпаки необхідна – скорочення кількості токенів у словнику моделі економить місце під час навчання моделі. Видалення пунктуації проводиться для практично всіх завдань, включно з сумаризацією, оскільки її відсутність не є критичною для розуміння тексту заголовка. Крім лематизації та видалення пунктуації, до процесу попереднього опрацювання тексту також входить приведення слів до нижнього регістру, необхідне за відсутності лематизації, оскільки без подібного опрацювання те саме слово на початку речення і в іншому місці може бути розпізнане як два різні токени.

У межах деяких завдань відбувається введення додаткових токенів, як-от «start» і «eos» (end of sequence), що позначають початок і кінець тексту на вході, «oov» (out of vocabulary), що позначає слово, яке не зустрічається в словнику.

Важливо зазначити, що крім специфічних автоматизованих кроків на кшталт токенізації, до попереднього опрацювання тексту також входять загальні види опрацювання тексту: його очищення від випадкових викидів, наприклад, зайвих символів, зіпсованих даних, переведення в кодування, яке можна прочитати обраним токенізатором, і т. д.

Перехід з простору термінів в простір знайдених тематик допомагає вирішувати синонімію і полісемію термінів.

2.1.1 Екстрактивна сумаризація

Екстрактивні системи створюють сумаризації шляхом визначення (і подальшого об'єднання) найважливіших речень у документі. Нейронні моделі розглядають екстрактивну сумаризацію як проблему класифікації речень: нейронний кодувальник створює уявлення речень, а класифікатор пророкує, які речення слід вибрати як сумаризації. SUMMARUNNER [42] – один із перших нейронних підходів, що використовують кодувальник на основі рекурентних нейронних мереж. REFRESH [43] – це система, заснована на навчанні з підкріпленням, навчена шляхом глобальної оптимізації метрики ROUGE. У пізніших роботах досягається більш висока продуктивність зі складнішими модульними структурами. LATENT [44] розглядає екстрактивне узагальнення як проблему виведення прихованих змінних; замість того, щоб максимізувати ймовірність речень, що підсумовують, їхня прихована модель безпосередньо максимізує ймовірність ручних сумаризацій.

SUMO [45] використовує поняття структурованої уваги для створення уявлення документа у вигляді дерева залежностей з кількома коренями при прогнозуванні сумаризацій. NEUSUM [46] оцінює і відбирає пропозиції спільно і на момент виходу був однією з найбільш якісних моделей екстрактивної сумаризації.

2.1.2 Абстрактна сумаризація

Нейронні підходи до абстрактної сумаризації концептуалізують завдання як завдання отримання однієї послідовності слів з іншої (seq2seq), де кодувальник відображає послідовність токенів у вихідному документі $x = [x_1 ,....,x_n]$ у послідовність безперервних подань $z = [z_1 ,...,z_n]$, і декодер потім генерує цільове зведення $y = [y_1 ,...,y_m]$ токен за токеном

авторегресивним способом, у такий спосіб моделюючи умовну ймовірність: $p(y_1 ,...,y_m |x_1 ,....,x_n)$. Nallapati та ін. (2016) стали одними з перших, хто застосував модель із використанням архітектури нейронного кодувальника-декодера для сумаризації текстів [47]. Надалі цю модель було вдосконалено за допомогою мережі генераторів покажчиків (PTGEN) [48], що дає змогу копіювати слова з вихідного тексту, і механізму покриття (COV), який відстежує слова, що були підсумовані. У методі Deep Communicating Agents (DCA) [49] пропонується система, в якій кілька агентів (кодувальників) представляють документ разом з ієрархічним механізмом уваги (через агентів) для декодування. Ця модель використовує наскрізне навчання з навчанням із підкріпленням. Paulus та ін. (2018) також представляють глибоку модель з підкріпленням (DRM) [50] для абстрактних сумаризацій, що вирішує проблему охоплення за допомогою механізму внутрішньої уваги, коли декодер обробляє раніше згенеровані слова. Gehrmann та ін. (2018) слідуючи висхідному підходу (BOTTOMUP), спочатку визначають, які фрази у вихідному документі повинні бути частиною сумаризації, а потім, механізм копіювання застосовується тільки до заздалегідь обраних фраз під час декодування [51]. Narayan та ін. (2018) пропонують абстрактну модель, яка особливо добре підходить для екстремального випадку – сумаризацій з одного речення, що базується на згорткових нейронних мережах, додатково обумовлених розподілом тем (TCONVS2S) [52].

2.1.3 Застосування мовних моделей

Крім названих раніше стандартних нейромережевих підходів, починаючи з 2018 року пропонували різні варіанти застосування універсальних мовних моделей як у завданні екстрактивних, так і абстрактних сумаризацій. Одним із прикладів може стати робота Yang Liu та

Mirella Lapata, в якій автори розглядають можливість застосування моделі BERT у завданні сумаризації текстових даних [53].

Хоча BERT використовували для розв'язання цілого спектра різних завдань НЛП, за допомогою методів донавчання і налаштування моделі, його застосування в задачах сумаризації неможливе безпосередньо. Оскільки BERT навчається як замаскована мовна модель, вихідні вектори прив'язані до токенів, а не до речень, тоді як під час екстрактивної сумаризації більшість моделей маніпулюють уявленнями на рівні речень. Хоча ембедінги сегментації представляють різні речення в BERT, вони можуть бути застосовані тільки до вхідних пар речень, тоді як під час сумаризації виникає необхідність кодування і маніпулювання вхідними даними, які стосуються кількох речень. На рисунку 2.1 показано пропоновану авторами оригінальної статті архітектуру BERT для сумаризацій (BERTSUM).

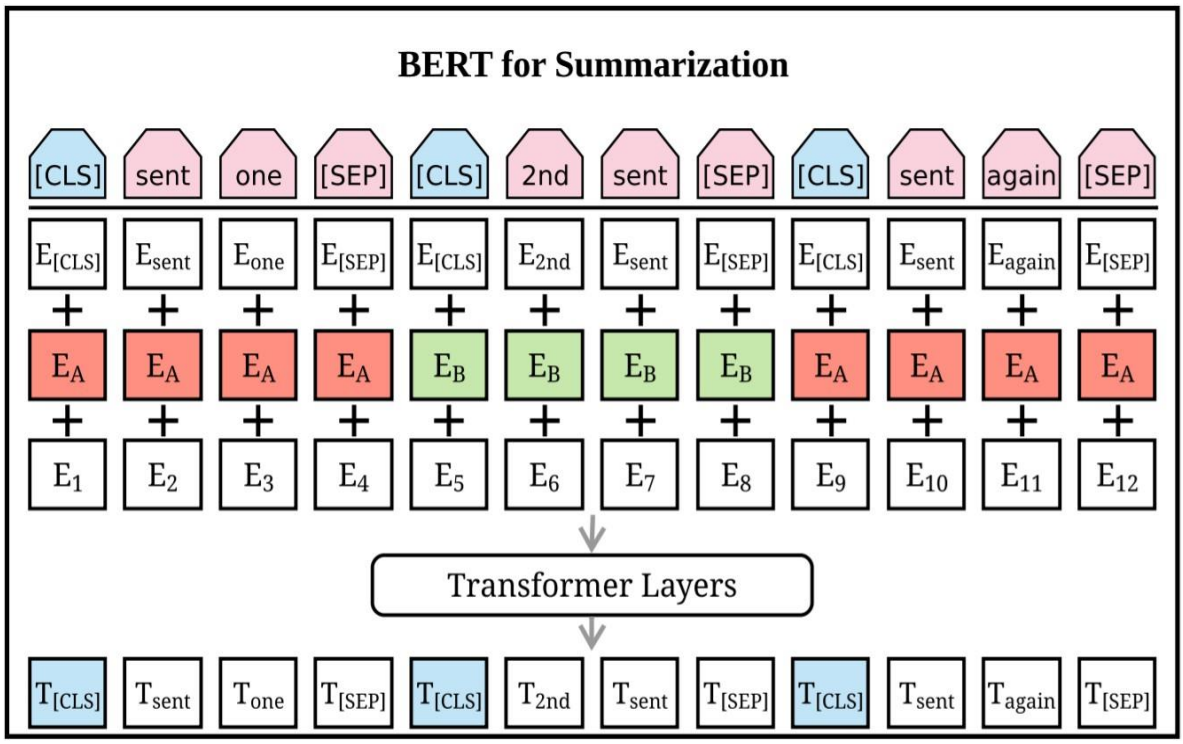


Рисунок 2.1 – Архітектура модифікації BertSum

На цій схемі верхній рядок – вхідні речення з роздільниками. Наступний шар являє собою суму трьох різних подань слів (аналогічно стандартній архітектурі BERT). Цю суму векторів використовують як вхідні кодування для шарів двонаправленого перетворювача, генеруючи контекстні вектори для кожного токена. BERTSUM розширює BERT, вставляючи кілька символів [CLS] для вивчення уявлень речень і використовуючи ембедінги інтервальної сегментації (показані червоним і зеленим кольором) для розрізнення кількох речень.

Нехай d позначає документ, що містить [$sent_1$,...,$sent_m$], де $sent_i$ – це і-те речення в документі. Екстрактивна сумаризація визначає таку задачу як задачу присвоєння класу $y_i \in 0,1$ кожному $sent_i$, що показує, чи входить об'єкт до сумаризації за $y_i = 1$, чи не входить за $y_i = 0$. Передбачається, що зібрані таким чином речення представляють найважливіші частини вихідного документа. За допомогою BERTSUM вектор t_i , який є представленням і-го символу [CLS] верхнього рівня, може використовуватися як представлення для $sent_i$. Потім кілька шарів Transformer між реченнями складаються поверх вихідних даних BERT, щоб захопити функції рівня документа для вилучення сумарності:

$$\widetilde{h^l} = LN\left(h^{l-1} + MHAtt(h^{l-1})\right), \tag{2.1}$$

$$h^l = LN\left(\widetilde{h^l} + FNN\left(\widetilde{h^l}\right)\right), \tag{2.2}$$

де $h^0 = PosEmb(T)$, T – вихід векторів речень, а функція $PosEmb(*)$ додає в кодування синусоїдну позиційну частину, показуючи положення кожного речення. Фінальний шар цієї мережі являє собою наступний сигмоїдний класифікатор.

$$\widehat{y_i} = \sigma\left(W_0 h_i^l + b_0\right), \quad (2.3)$$

де h_i це векторне представлення для пропозиції $sent_i$ з верхнього шару Transformer моделі.

Абстрактна сумаризація BERT – на відміну від екстрактивної сумаризації в цьому завданні можливе використання стандартної архітектури кодувальника-декодувальника.

Кодувальник – це попередньо навчена модель BERTSUM, а декодувальник – це 6-шаровий перетворювач, ініціалізований випадковим чином. Цілком можливо, що існує невідповідність між кодером і декодером, оскільки перший попередньо навчений, а другий повинен навчатися з нуля. Це може зробити тюнінг моделі нестабільним. Так, наприклад, кодувальник може перенавчитися, а декодувальник навчитися недостатньо, або навпаки. Щоб обійти цю проблему автори оригінальної статті пропонують двоетапний підхід до тонкої надбудови (fine-tuning), за якого спочатку налаштовують кодувальник за допомогою задачі екстрактивної сумаризації, а потім налаштовують для задачі абстрактної сумаризації. Li та ін. у своїй роботі показують, як використання екстрактивних завдань може підвищити ефективність абстрактних сумаризацій [14]. Зважаючи на відносну простоту цього підходу, відсутність змін в основі архітектури моделі та загального поліпшення результатів тюнінгу моделі, цей підхід є стандартним під час навчання цієї моделі.

2.2 Векторизація текстових даних

Після попередньої обробки тексту стає можливим його переведення в числове представлення. Існує велика кількість підходів до векторизації, що використовуються залежно від параметрів завдання.

Для відносно простих завдань, у яких текст не генерується, наприклад, як для згаданих вище сентимент-аналізу та класифікації за темами, можливе використання відносно простих методів векторизації, які не потребують витратних обчислень, оскільки обчислюється вектор речення, а не слова. Одним із таких методів є TF-IDF – для того, щоб представити слово в числовому вигляді за допомогою такого методу, достатньо підрахувати частоту терміна в тексті (term frequency) і поділити її на зворотну частоту терміна в усьому корпусі (inverse document frequency). Вектор тексту дорівнюватиме довжині спільного для корпусу вокабуляра, на позиції слова у вокабулярі стоятиме його значення TF-IDF. Ще простішим методом є Bag of words – замість значення TF-IDF буде тільки TF, частота терміна в тексті. Ці методи передбачають, що порядок слів неважливий, а як було згадано вище, для нейронної мережі, що генерує текст, важлива наявність зв'язності в тексті. Для забезпечення перекладу тексту у векторний вигляд для складніших завдань використовують інші методи.

У разі сумаризації використовують методи, що дають змогу отримати вектори слів або підрядків, у такому разі послідовність слів у тексті просто задають послідовністю векторів, які подають на вхід. Найпростішим із подібних методів можна назвати One-Hot Encoding – у векторі слова довжини розміру вокабуляра на позиції поточного слова стоїть одиниця, інші вектори дорівнюють 0. Такі вектори займають надто багато місця, оскільки величина вокабуляра може досягати десятків тисяч, тому практично не використовуються. Більш економним є підхід Word2Vec. Такий метод дає змогу не просто перевести слова у вектор, а й дає можливість відобразити деякі семантичні властивості тексту. Так, вектори зі схожим значенням перебуватимуть на близькій відстані у створеному векторному відображенні тексту. Розмірність вектора слова при застосуванні такого або аналогічного йому способу кодування слів буде невелика – близько 500 [31].

У цій роботі використовується векторизація на підрядковому рівні, а саме метод SentencePiece. Цей метод реалізовано на основі двох інших підрядкових підходів до векторизації – Byte-Pair Encoding і Unigram Language Model [15]. Підрядковий підхід передбачає, що слова, які рідко використовуються в корпусі, мають бути декомпоновані в підрядки, які використовуються частіше. Ця властивість дає змогу значно економити місце під словник моделі. Незважаючи на особливу ефективність для аглютинативних мов, цей метод підходить і для російської мови – підрядки, що формуються таким методом, можна грубо співвіднести з частинами слова. Так, наприклад, у словах «токенізатор» і «лемматизатор», які нечасто трапляються в новинах, під час використання BPE майже напевно буде виділено загальний підрядок «##изатор». Символи «##» у цьому випадку позначають, що цей підрядок несамостійний і має йти відразу після іншого підрядка. SentencePiece має і свої особливості – на відміну від BPE і ULM, які розраховані насамперед на індоєвропейські мови, він універсальніший, оскільки сприймає послідовність на вході, а не розраховує на вже розділений на токени текст – це дає змогу сприймати і мови, які не використовують пропуски, завдяки тому, що вони приймаються за символ нарівні з іншими. Для алгоритмів, розрахованих на роботу з текстом російською мовою, це означає меншу, порівняно з Word2Vec та іншими векторносемантичними моделями, інтерпретованість токенізованого тексту, але швидшу роботу і можливість використовувати алгоритм для інших мов.

2.3 Архітектура Seq2Seq

Повторимо, що завдання генерації заголовка статті еквівалентне завданню сумаризації її тексту, а сумаризація, своєю чергою, є завданням перекладу послідовності в послідовність, тексту статті або деякого новинного тексту в заголовок статті. Саме для роботи з послідовностями

призначена запропонована 2014 року [43] архітектура нейронної мережі Seq2seq (sequence to sequence), що стала натхненням для інших моделей на кшталт Transformer, архітектура нейронної мережі Seq2seq (sequence to sequence).

Головними компонентами архітектури Seq2Seq є вже згадані кодувальник і декодувальник, а робота моделі зводиться до двох процесів – сумаризації та генерації тексту.

Під час фази навчання кодувальник, що складається з блоків рекурентної нейронної мережі, «читає» послідовність, запропоновану йому для навчання. На кожному кроці відбувається запам'ятовування цієї послідовності в процесі ітеративної побудови подання інформації. Під час переходу з одного блоку в інший використовуються прихований стан і поточні вхідні дані, отримані в попередньому блоці. Таким чином, декодувальник під час фази навчання ніби тренується передбачати вихідну послідовність, передбачаючи ту саму послідовність, яка надходить до нього на вхід, із запізненням на один крок у часі. Токени початку і кінця послідовності, згадані раніше в розділі, присвяченому попередній обробці тексту, використовуються для того, щоб дати сигнал декодеру про початок передбачення.

Після навчання модель тестується на новому матеріалі, виведення якого невідоме моделі. Для цього декодувальнику надходить токен початку, після чого кілька разів відбувається процес вибору найімовірнішого слова передбачуваного тексту, поки не буде згенеровано токен кінця послідовності. Потенційна проблема такої архітектури полягає в тому, що нейронна мережа повинна змогти стиснути інформацію так, щоб вона помістилася у вектор фіксованої довжини, а це може ускладнити обробку довгих речень.

Завдяки своїй простоті архітектура Seq2seq відкрита до доповнення механізмами, техніками та прийомами. Серед них найважливішим для

розуміння архітектури Transformer, яку ми використовуємо, є механізм уваги.

2.4 Механізм уваги

На відміну від звичайної Seq2Seq, модель із механізмом уваги не вимагає кодувати всю послідовність, що надходить на введення. За рахунок додаткового кроку між кодувальником і декодувальником, останній ніби фокусується на найважливіших словах речення. Увагу можна інтерпретувати як вектор важливості ваг: для того, щоб передбачити наступний елемент послідовності, ми за допомогою уваги визначаємо, наскільки він співвідноситься з іншими елементами послідовності. Потім сума значень ваг, оцінена увагою, допомагає нам наблизитися до цього елемента.

Відбувається це таким чином. Набір прихованих станів – векторів закодованих слів – передають у декодувальник, що відповідає за механізм уваги, і там кожен прихований стан оцінюють за формулою уваги, і йому призначають ваги, після чого обчислюють вектор контексту [2]:

$$a_{ts} = \frac{\exp(score(h_t, \overline{h_s}))}{\sum_{s'}^{s} \exp(score(h_t, \overline{h_s}))}, \tag{2.4}$$

де s – індекс кодувальника, t – індекс декодувальника, ats – ваги уваги, hs – послідовність, що виводиться з декодера, ht – стан декодера, що відповідає послідовності, ct – вектор контексту, at – фінальний вивід, що поєднує контекст і стан декодера.

Механізм уваги, представлений вище, є так званою «м'якою» (soft) увагою. Існує також і «жорстка» (hard) увага [23], що робить фокус ще вужчим – замість оцінювання всіх прихованих станів обирається тільки невелика частина. Таким чином, у завданні сумаризації вона не

розподіляється рівномірно по всьому реченню, а зосереджена на окремих його словах, що, наприклад, особливо корисно в задачах генерації тексту. Обчислення моделі з жорсткою увагою є швидшим, ніж із м'якою, особливо на великих обсягах даних, однак модель потребує складних механізмів навчання, оскільки перестає бути диференційованою за використання такого механізму.

Компромісним варіантом виступає локальна увага. На відміну від м'якої, яка в цій концепції називається глобальною, і подібна до жорсткої уваги, local attention оцінює ваги не всієї послідовності, що надійшла на вхід, а обирає лише частину з них, що дає змогу економити обчислювальні ресурси. Разом з цим, за рахунок того, що вікно, яке обирається для фокусу, ширше, модель стає диференційованою, і стає можливим уникнути складнообчислюваного навчання.

2.5 Методи аналізу тональності тексту

Тональність можна визначити як погляд або думку, що міститься в деякому тексті. Sentiment Analysis при цьому визначається як це процес ідентифікації та категоризації думок, виражених у фрагменті текстового контенту, зокрема, для визначення ставлення автора до певної теми, продукту або проблеми [55]. Sentiment Analysis, також відомий як opinion mining, широко використовується в багатьох галузях, як-от аналіз продуктів, послуг, соціальних і політичних проблем, для аналізу поведінки або думки користувачів за певними темами. Метою аналізу тональності є виявлення позитивних, негативних або нейтральних настроїв у наборі текстів або документів [57].

2.5.1 Традиційні підходи

Традиційно це завдання вирішували двома основними підходами: словниковим і за допомогою методів машинного навчання [58].

Підходи, засновані на словниках, являють собою статистичні методи, що використовують попередньо зібрані словники тональності, які містять різні слова і полярність, що їм відповідає, для визначення заданого слова як позитивного або негативного. Stone та ін. [59] уперше окреслили завдання аналізу тональності з використанням методу словників ще 1966 року. Пізніше були запропоновані різні словникові набори, такі як WordNet, WordNet-Affect, SenticNet, MPQA і SentiWordNet [60]. Ці підходи не вимагають набору навчальних даних. Однак побудова повних словників для великих обсягів неструктурованих даних, що генеруються користувачами, є складним завданням.

Підходи на основі машинного навчання допомагають вирішити проблему. Такі підходи засновані на алгоритмах класифікації слів за відповідними мітками тональності. Основна перевага підходів на основі машинного навчання – їхня здатність до репрезентативного навчання. Pang та ін. [61] вперше застосували ці методи для аналізу тональності. Алгоритмам машинного навчання потрібен навчальний набір даних, який допомагає автоматизувати класифікатор, і тестовий набір даних, що використовується для перевірки працездатності класифікатора. Тому підходи машинного навчання кращі для аналізу настроїв через їхню здатність працювати з великими обсягами даних порівняно з підходами, заснованими на словниках [62].

Попри свою простоту та інтуїтивно зрозумілу систему роботи, традиційні алгоритми мають безліч проблем, особливо помітних під час роботи з короткими текстами, отриманими із соціальних медіа. Саме тому

сучасні моделі дедалі частіше створюються на основі різних архітектур глибокого навчання, зокрема класифікації на основі мовних моделей.

2.5.2 Підходи на основі нейромереж

За останні роки дослідниками було запропоновано безліч підходів на основі глибокого навчання, більшість з яких, так чи інакше, було застосовано для аналізу тональності. Серед таких підходів можна назвати переднавчені мережі без вчителя (UPNs), згорткові мережі (CNNs), рекурентні мережі (RNNs), рекурсивні мережі (RvNNs) та описані раніше трансформерні мережі. У різний час ці підходи показували найкращі результати, проте тільки з початком використання трансформерних мереж та ідей трансферного навчання, підходи на основі глибокого навчання практично повністю витіснили традиційні методи. Нові підходи розв'язали складні проблеми, як-от адаптація предметної області, можливість працювати з контекстом і моделювати довгострокові залежності.

2.5.3 Архітектура Transformer

Основна відмінність архітектури Transformer від архітектури Seq2seq – відмова від рекурентних блоків, таких як RNN і LSTM, завдяки тому, що ця мережа цілком побудована на механізмі multi-head self-attention. Трансформер отримує на вході набір пар ключ-значення (key-value), де розмірність обох компонентів дорівнює довжині вхідної послідовності. Для сумаризації, і ключі, і значення – це приховані стани в кодувальнику. У декодері ключ і значення поєднуються в запит (query), і наступний висновок утворений розміткою цього запиту і пари ключ-значення [45]:

$$Attention(Q, K, V) = softmax\left(\frac{QK^{\top}}{\sqrt{n}}\right)V. \qquad (2.5)$$

Замість того, щоб порахувати увагу лише один раз, її прораховують паралельно кілька разів, після чого незалежні оцінки конкатенують і лінійно трансформують, що дає змогу уникнути надмірного впливу подальшого усереднення. Архітектурно виглядає імплементація цього механізму, описана на малюнку нижче таким чином [45].

Використані моделі (BERT [14], BART [5] і T5 [6]) є моделями, заснованими на архітектурі типу «Трансформер». Загальну архітектуру представлено на рисунку 2.2.

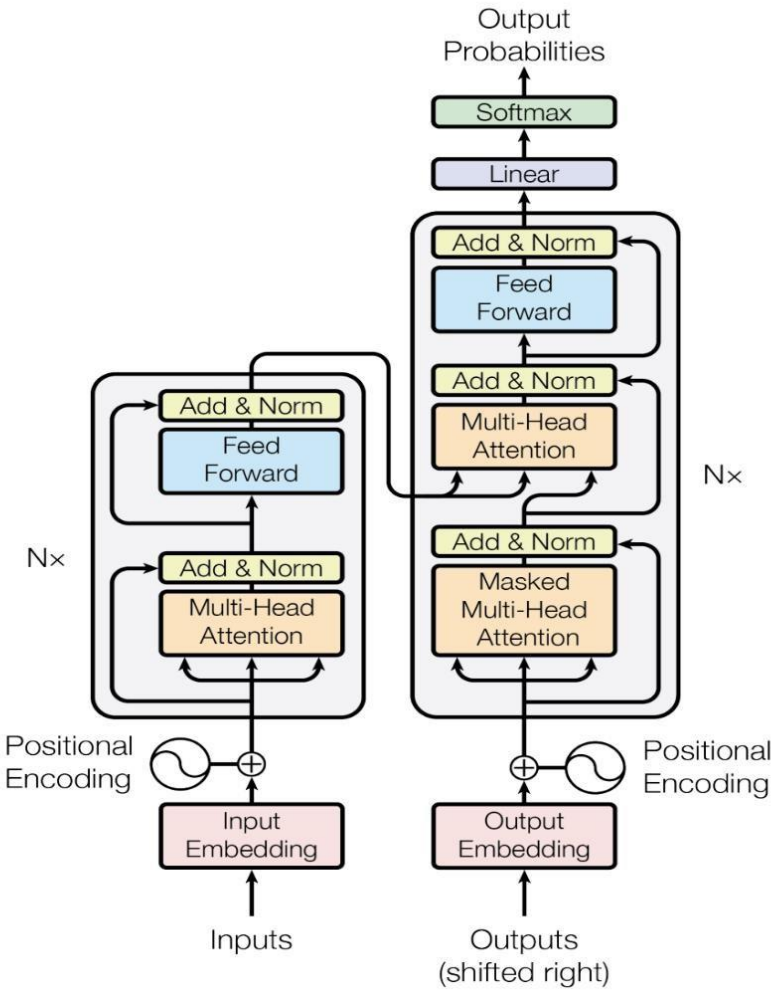


Рисунок 2.2 – Архітектура моделі трансформер

Основна відмінність цих моделей від більш ранніх полягає у відході від використання рекурентних нейронних мереж, замінюючи їх механізмом уваги [8].

Функцію внутрішньої уваги $Attention(Q, K, V)$ можна описати як зіставлення запиту і набору пар ключ-значення з вихідними даними, де запит, ключі, значення і вихідні дані є векторами. Результат обчислюється як зважена сума значень, де вагу, присвоєну кожному значенню, обчислює функція сумісності запиту з відповідним ключем (рисунок 2.3).

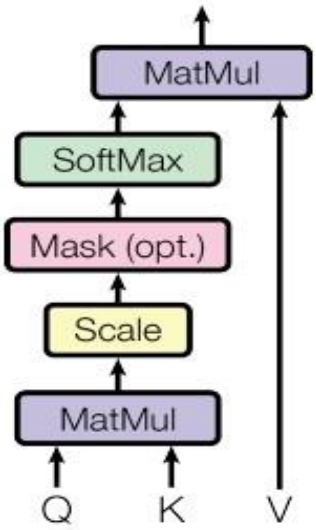


Рисунок 2.3 – Схема внутрішньої уваги

В оригінальній статті внутрішня увага покращується за допомогою додавання механізму, що називається множинною увагою $MultiHead\ (Q, K, V)$, Однак, як буде показано надалі, ці завдання вирішуються мовними моделями шляхом отримання останнього прихованого стану моделі. Такий підхід дає змогу отримувати якісні подання даних, оптимізованих для специфічних завдань (рисунок 2.4).

Основні переваги цього методу:

– простота реалізації – реалізований майже у всіх фреймворках роботи з природною мовою;

– швидкість роботи найпростіші операції підрахунку слів;

– розряджені матриці дають змогу використовувати спеціальні формати зберігання даних.

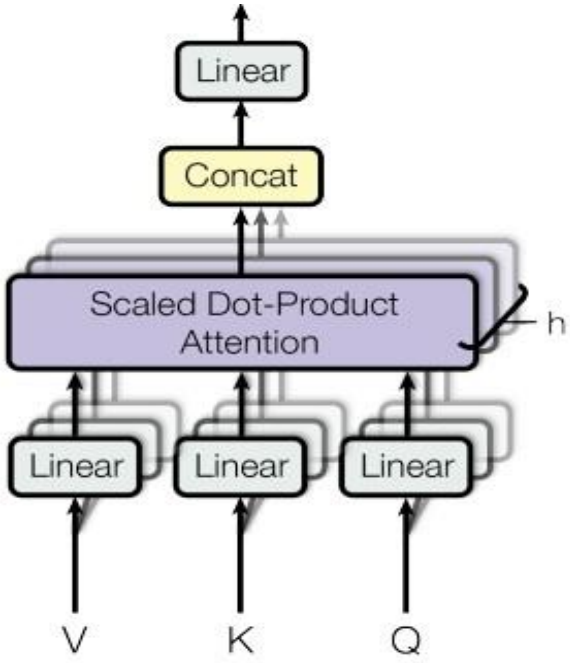


Рисунок 2.4 – Схема роботи множинної уваги

$$MultiHead(Q,K,V) = Concat(head_1, \dots, head_n)W^0, \quad (2.6)$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right), \quad (2.7)$$

де W – матриці ваг.

Ще одним із головних нововведень у моделі є позиційне кодування. Оскільки модель не містить рекурентних, для того щоб модель могла використовувати порядок послідовності, ми повинні ввести деяку інформацію про відносне або абсолютне положення токенів у реченні. Ідея

полягає в конкатенації або підсумовуванні (що більш часте явище на практиці) вхідних векторів з якимось вектором, який зберігає інформацію про позицію слів у реченні.

Цю залежність зображено на рисунку 2.5.

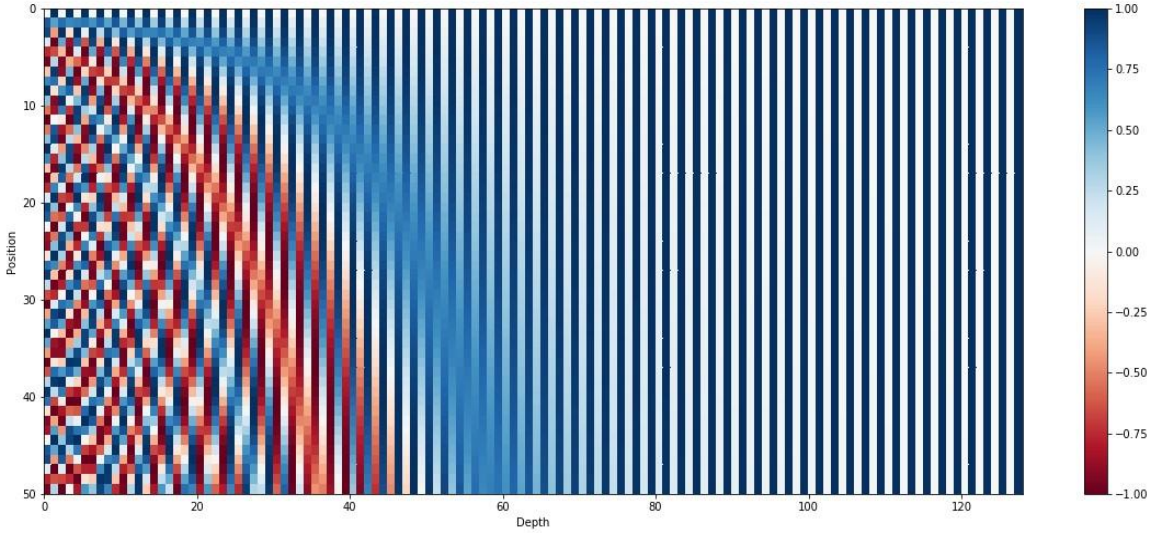


Рисунок 2.5 – Залежність позиції від частоти компоненти вектора

Наразі ці моделі є State-of-the-art моделями, що показують найкращі результати в різних sequence-to-sequence завданнях.

2.6 Глибоке навчання та fine-tuning

Як ми вже зазначили вище, завдання абстрактної сумаризації містить у собі завдання генерації тексту. Саме тому розглянуті в цьому розділі архітектури Seq2seq і Transformer – це архітектури нейронних мереж для глибокого навчання. Це означає, що нейромережа навчається не розв'язанню конкретного завдання, а мовним уявленням, тобто створює мовну модель, виходячи з якої і відбувається, наприклад, передбачення тексту.

З поширенням глибоких моделей популярності набув підхід до розв'язання задач опрацювання природної мови, за якого переднавчену мовним уявленням модель донавчають залежно від завдання. Він має велику кількість переваг порівняно з методом, що використовувався раніше, серед них, наприклад, забезпечення перевикористання моделей – це дає змогу економити час і ресурси. Економить час і швидка імплементація моделей, що дає змогу, однак, здійснити досить індивідуальне налаштування. Утім, сучасні моделі, навчені на великому корпусі, не вимагають навіть завдання параметрів і здатні показувати адекватні результати без тонкого налаштування.

Процес переднавчання моделі являє собою «згодовування» їй великої кількості нерозмічених даних для формування мовних уявлень.

Великий обсяг даних, необхідний для формування адекватної моделі, накладає певні обмеження для навчання, що, своєю чергою, створює умови для подальшої роботи моделі. Так, під час збирання корпусу необхідно забезпечувати репрезентативність з метою дотримання нейтральності мови моделі та збирати тексти різних стилів і типів. Великий обсяг даних також гарантовано зробить процес навчання моделі трудомістким, тому зазвичай обирають найпростіші та найнадійніші алгоритми.

Fine-tuning, за рахунок меншого оброблюваного обсягу даних, є менш витратним процесом, ніж переднавчання, і тому більш відкритим до модифікацій. Існує досить велика кількість різновидів донавчання залежно від завдання:

– послідовна адаптація – у донавчання додається передпідготовка моделі на датасеті для аналогічного завдання. Це допомагає розв'язати проблему невеликого обсягу вибірки для донавчання і загалом покращує якість роботи моделі навіть без переднавчання на нерозміченому датасеті [30];

– мультизадачне донавчання – цей вид розв'язує ті самі проблеми, що й послідовна адаптація, проте часто передбачає, що верхні шари моделі специфічні для обраних завдань [35];

– навчання з частковим залученням вчителя – додаючи в нерозмічені дані перешкоди на кшталт шуму або модифікуючи вже наявні дані, тобто здійснюючи аугментацію даних [7];

– ансамблі моделей – під ансамблями можна мати на увазі як різні моделі, так і одну й ту саму модель із різними параметрами або одну й ту саму модель, навчену для аналогічних цільовому завдань.

2.7 Модель mT5

Використовувана нами переднавчена модель mT5 [48] (Text-To-Text Transfer Transformer) побудована на кшталт T5, яка, своєю чергою, побудована на основі класичної архітектури Transformer [34] і призначена безпосередньо для завдань обробки текстів природною мовою. Її розробники підкреслюють, що не пропонують нововведень архітектури, однак до створення було застосовано новий підхід: усі завдання, на вході та виході яких має бути текст, об'єднано в одне мета-завдання, яке й розв'язує ця модель. Це стандартна практика для генеративних задач, проте нововведення для задач класифікації, у яких T5 сприймає класи теж як послідовність (наприклад, у задачі сентимент-аналізу виводяться рядки «positive» і «negative»).

mT5 – це багатомовний варіант моделі T5, і, відповідно, найважливішим елементом переднавчання став корпус mC4, багатомовний аналог англомовного корпусу C4. Він містить 107 мов, зокрема російську мову в кирилиці та латиниці. Кожна мова представлена в корпусі щонайменше 10 000 сторінками, кожна з яких довша за 200 символів.

Автори вказують, що один із найефективніших способів використання цієї моделі – завдання заповнення пустот на противагу завданню генерації, з яким, наприклад, успішніше впорається GPT-3, проте передбачається, що донавчання може забезпечити успішні результати.

T5 [6] (Text-to-Text Transformer) – чергова модель типу Трансформер. Аналогічно до попередньої моделі BART [5], T5 [6] має як блоки анкодера, так і блоки декодера. Архітектура моделі наведена на рисунку 2.6.

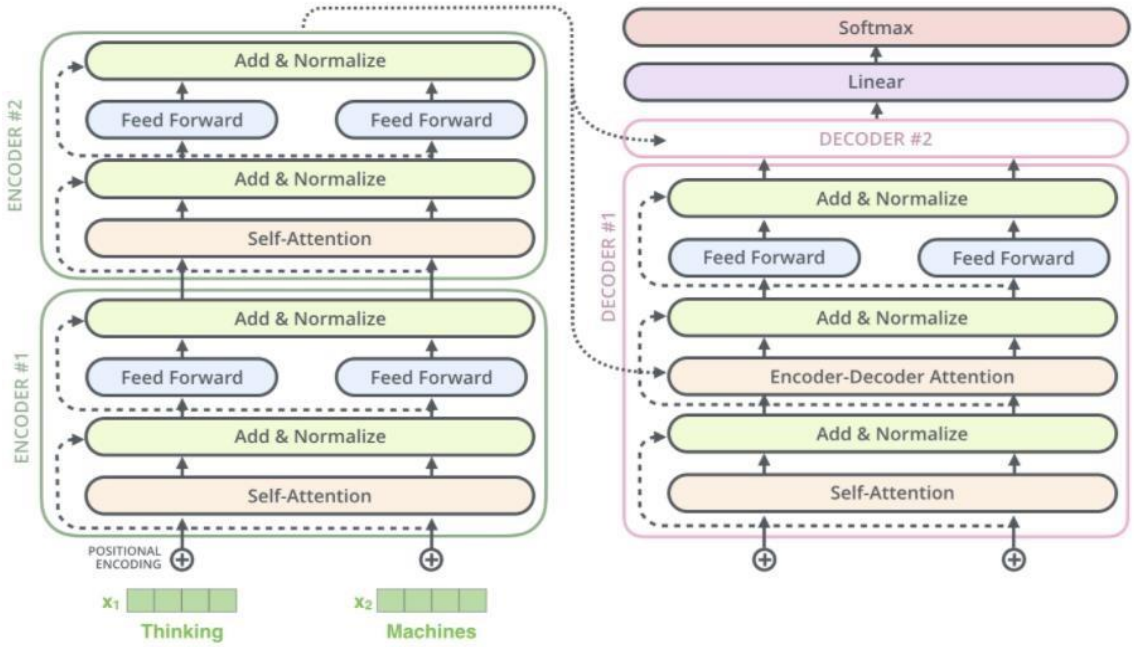


Рисунок 2.6 – Архітектура моделі T5

Text-to-text моделі, такі як T5, відкривають широкі можливості для розв'язання різних завдань обробки природної мови. На відміну від традиційних моделей, які працюють лише з визначеним набором завдань, T5 може генерувати текст відповідно до вхідних текстових даних. Це дозволяє використовувати модель для сумаризації текстів, машинного перекладу, регресії, класифікації та навіть задач «запитання-відповідь».

Одним з ключових аспектів успіху нейронних мереж є їх здатність апроксимувати складні функції, знаходячи відповідні закономірності в даних. Для цього моделі проводять обчислення, що дозволяє їм розв'язувати різноманітні завдання в області обробки природної мови.

У контексті сумаризації тексту, використання моделі T5 потребує представлення текстових даних у чисельному вигляді. Один з ефективних методів цього – векторизація, зокрема векторизатор SentencePiece, який працює на рівні підрядків. Це дозволяє ефективно вирішувати проблему відсутності слова в словнику моделі.

Перехід від Seq2seq архітектури до Transformer став ключовим кроком у розвитку моделей для обробки природної мови. Механізм уваги, зокрема multi-head self-attention, дозволяє моделям ефективно взаємодіяти з різними частинами вхідних даних.

Також, щоб зменшити витрати на навчання моделей, застосовується підхід fine-tuning, де моделі спочатку навчаються загальним мовним уявленням, а потім донавчаються для конкретного завдання. Прекрасним прикладом таких переднавчених моделей є багатомовна mT5, яку можна використовувати для сумаризації новинних заголовків українською мовою та інших завдань в обробці тексту, приклад на (рисунку 2.7).

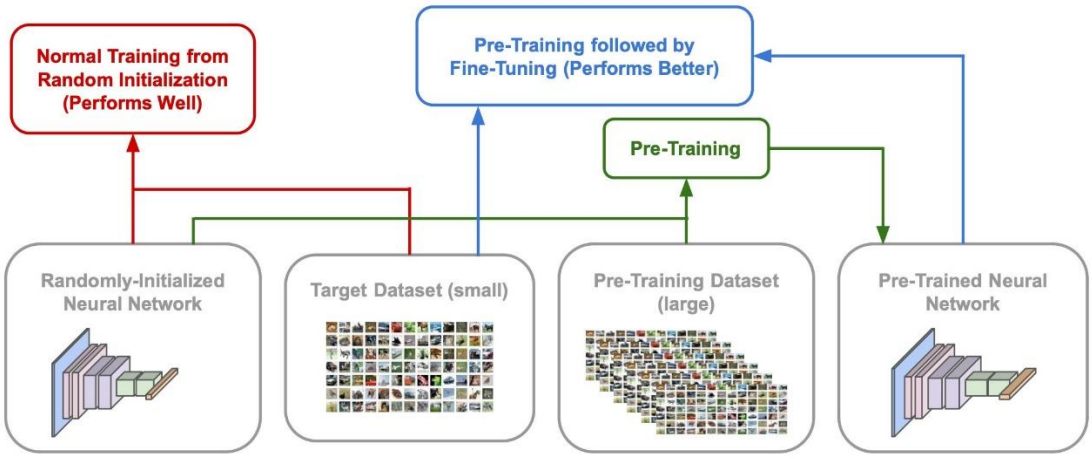


Рисунок 2.7 – Процесс сумаризації T5

3 ПРАКТИЧНЕ ЗАСТОСУВАННЯ МОДЕЛІ TRANSFORMER ДЛЯ РЕЗЮМУВАННЯ НОВИННИХ СТАТЕЙ

3.1 Проект World2News

Цей розділ присвячено застосуванню інформації з розділів 1 і 2 для створення власного автоматичного сумаризатора на основі архітектури Transformer для аналізу новинних статей заголовків. Перший розділ ставить наше дослідження в контекст наявних наукових праць, присвячених автоматичній сумаризації, та обґрунтовує вибір моделі для подальшої роботи, другий присвячений нейронним мережам та еволюції обраної нами архітектури.

Незважаючи на теоретичне обґрунтування й очевидні переваги деяких підходів, деякі нейромережі вимагають величезних часових і обчислювальних ресурсів залежно від кількості даних. Практичні ускладнення такого роду призводять до негативних результатів, і ми припускаємо, що вони могли негативно позначитися на результатах цього дослідження.

Робота, виконана раніше, ставила собі за мету створення базової глибокої рекурентної рекурентної нейронної мережі для генерації заголовків на багатомовних моделях із новинних статей. Як архітектуру нейронної мережі було обрано Sequence-to-sequence, вона ж Seq2seq, оскільки робота спирається на припущення про особливу застосовність глибокого навчання в задачі сумаризації, а ця архітектура є базовим рішенням для глибокого навчання. Крім того, автори висловлюють припущення, що модифікації та механізми поліпшення роботи, як-от механізм уваги, можуть забезпечити результат, який можна порівняти або який перевершує Transformer. Автори докладно розглядають особливості рекурентних блоків, що становлять важливу частину обраної ними

структури моделі і є стандартом для аналізу послідовностей, в якості якої в разі сумаризації новини розглядається текст.

У розділі 2 ми детально розглянули цю модель і виявили відмінності між двома архітектурами, однак ми вважаємо важливим повторити деякі ключові аспекти, заодно зачіпаючи інші істотні відмінності між двома роботами.

Основною відмінністю між Seq2seq і Transformer є відсутність рекурентних блоків, на яких заснована перша архітектура. Для опрацювання послідовностей і меншої втрати контексту Transformer використовує багатоголовий механізм уваги, який не імплементовано у створеній у більш ранній роботі моделі Seq2seq. Для вирішення проблеми втрати пам'яті в довгих реченнях використано блоки довгої короткострокової пам'яті (LSTM).

Істотною відмінністю між двома роботами стало використання в даному дослідженні концепції fine-tuning; попередня робота була створена і навчена з нуля.

Як і в попередній роботі, матеріал для навчання моделі був узятий із готових новинних корпусів видань ICTV.ua та «Україна Сьогодні», проте було також додано тексти з ресурсів RBK.ua та Вікіпедіа «Новини», зібрані в НДЛ «Когнітивні дослідження мови».

3.2 Матеріал дослідження

Навчання моделі проходило на основі матеріалів чотирьох новинних агентств:

ICTV.ua: корпус, що містить дані новинного сайту ICTV.ua був уперше представлений у 2018 році, однак відтоді неодноразово був оновлений. Він містить понад 700 тисяч пар «новина-заголовок»,

доповнених низкою атрибутів для інших завдань – серед них є посилання на матеріал на сайті (URL), тег, категорія.

Часовий період новин, представлений у корпусі – з вересня 1999 року по вересень 2019 року, що забезпечує різноманітність словника і репрезентативність корпусу [21].

«Україна Сьогодні»: корпус розміром у понад 1 мільйон новин було надано в рамках змагання з генерації новинних заголовків конференції «Діалог», що відбулася 2019 року. Часовий період цього корпусу значно менший, від січня 2010 року до грудня 2014 року, проте ми припускаємо, що обсяг корпусу достатній для забезпечення мовного розмаїття.

ТАСС: зібраний у межах проєкту World2News корпус із сайту новинного агентства ТАСС (tass.ua), корпус містить понад 700 тисяч новин. У ньому зібрано новини від 2014 до 2021 року, серед тем, представлених у корпусі, є «Армія», «Економіка», «Космос», «Культура», «Наука» і «Суспільство» (поділ за темами здійснено редакторами сайту).

Сумарно кількість зібраних пар «новина-заголовок» склала понад 2 мільйони, що є великою кількістю для донавчання нейронної мережі для завдання генерації новинних заголовків.

3.3 Опис інструменту розроблення

Весь код, створений під час роботи над практичною частиною дослідження, написаний мовою Python. Ця мова підходить для широкого спектра завдань завдяки розширюваності – для неї розроблено велику кількість бібліотек для специфічних завдань, таких як машинне навчання, робота з текстовими даними, обробка тексту природною мовою. Завдяки цим перевагам Python є мовою, яку активно використовують науковці та розробники у сфері NLP.

Під час роботи було використано велику кількість бібліотек і модулів, вони обслуговували весь процес побудови моделі – від даних до виведення. Серед них:

– pandas – бібліотека для роботи з даними. Ця бібліотека забезпечила аналіз, обробку та зберігання вибірки [25];

– re – бібліотека для підтримки регулярних виразів, незамінних у завданні попереднього опрацювання текстових даних і очищення від небажаних символів;

– nltk – natural language toolkit, одна з найбагатших бібліотек із роботи з природною мовою, застосовна в різних галузях підготовки тексту;

– transformers – бібліотека, що дає змогу використання переднавчених моделей. На практиці не додається в проєкт цілком, а дає змогу додати окремі модулі обраних токенізаторів і моделей;

– torch – фреймворк машинного навчання, що дозволяє потужні тензорні обчислення на основі роботи на GPU (graphics processing units) [5];

– scikit-learn – бібліотека інструментів для аналізу даних і машинного навчання. У цій роботі забезпечила поділ вибірки на тренувальну, тестову та валідаційну.

3.4 Підготовка текстових даних

Для успішного використання корпусів, як уже згадувалося в розділі 2, необхідно здійснити попередню підготовку текстових даних від «сирого» вигляду – вигляду, у якому новини надруковано на новинному ресурсі, звідки їх збирають у корпус, і до чисельного представлення, у якому їх надсилають у нейронну мережу для її навчання або дозвичаєння. Як уже було зазначено раніше, набір дій у процесі попереднього опрацювання залежить насамперед від завдання, поставленого користувачами вибірки. У нашому випадку процес дещо відрізнявся від ранньої роботи в проєкті

World2News, оскільки містив додаткові кроки зі створення корпусу із зібраного матеріалу.

Першим кроком з підготовки стало зібрання файлів формату .csv (comma-separated values) в єдиний датасет. Цей крок передбачає не тільки об'єднання файлів в один документ, а й перевірку на пропущені значення і видалення спотворених даних. Так, на цьому етапі з датасету було видалено понад 7 тисяч новин, оскільки під час їхнього запису до файлу в процесі збирання тексту відбулися спотворення.

Далі було проведено очищення текстів від зайвих символів. Крім стандартного позбавлення від розділових знаків і великих літер були проведені і менш очевидні процедури. До них включено, наприклад, очищення від символів, які гарантовано не повинні входити в текст, але можуть бути виправимим спотворенням унаслідок збору інформації в інтернеті – це теги мови веб-розмітки HTML і знаки, що є елементами синтаксису інших використовуваних у розмітці веб-сторінок мов програмування. Таким чином, було видалено символи «}», «]», «>» тощо. Видалення зайвих символів було здійснено функцією sub бібліотеки re.

Незважаючи на те, що корпус Lenta.ru вже пройшов часткову передпідготовку і не вимагав перевірки на пропущені значення, під час аналізу даних було виявлено помилку кодування даних, яка вимагала додаткового опрацювання – пробіл було замінено специфічною послідовністю символів.

Для адекватного опрацювання послідовностей нейронною мережею було здійснено додавання токенів <start> і <end>.

Після очищення текстів було здійснено поділ вибірки на три частини: тренувальну, тестову і валідаційну. Поділ найчастіше в подібних до нашого дослідженнях проводять у пропорції 70/30 або 80/20. У нашому випадку було використано поділ 80/20 для забезпечення моделі великою кількістю даних для навчання. Після цього поділу було здійснено додатковий поділ

тестової вибірки в тому самому співвідношенні для отримання валідаційної частини.

Існує чимало технік попередньої обробки тексту, які можна застосувати залежно від типу та призначення текстових даних. Одними з найпоширеніших є:

– токенізація: це процес розбиття тексту на менші одиниці, які називаються токенами. Токенами можуть бути слова, речення, абзаци тощо. Токенізація допомагає розбити текст на осмислені сегменти, які можна легко опрацювати за допомогою НЛП-моделей (рисунок 3.1).

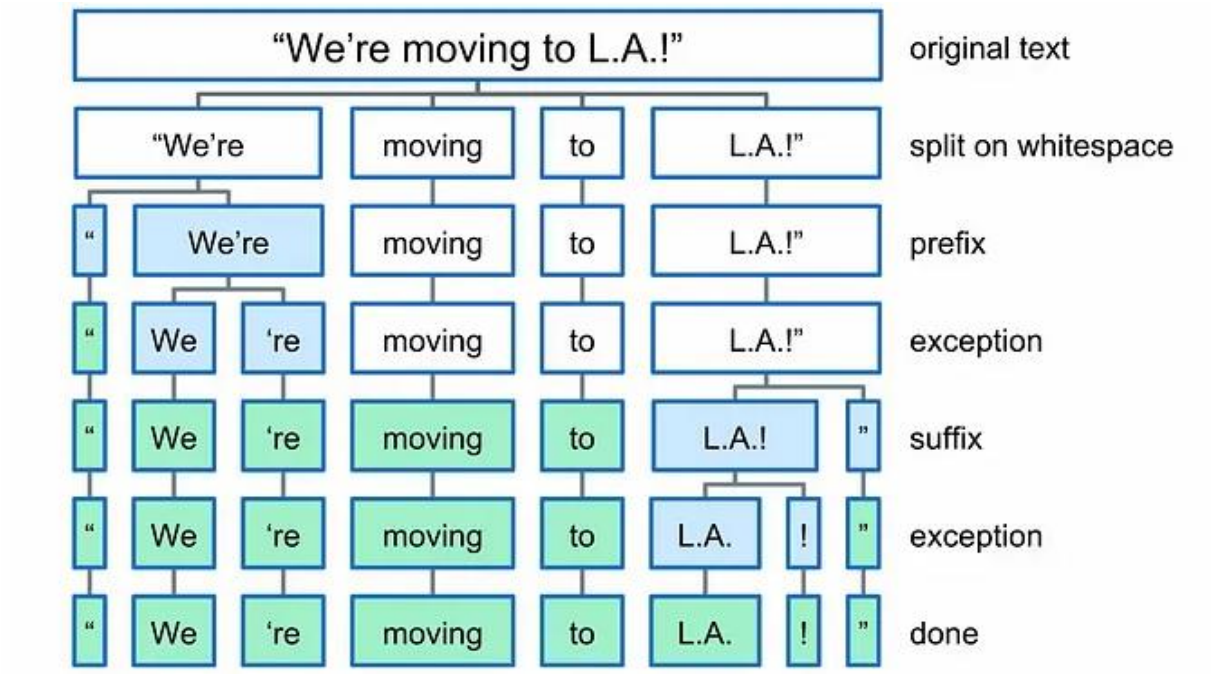


Рисунок 3.1 – Фізичний зміст процесу токенізації

– нормалізація: це процес перетворення тексту в стандартну або загальноприйняту форму. Нормалізація може включати:Перетворення регістру: це процес зміни регістру літер у тексті на нижній або верхній регістр. Перетворення регістру допомагає зменшити варіативність тексту та зробити його більш послідовним;

– стемінг: Це процес приведення слів до їхньої кореневої або базової форми шляхом видалення суфіксів. Наприклад, слова «running», «runs» і «ran» можна привести до слова «run». Стеммінг допомагає зменшити кількість слів у тексті та спростити словниковий запас;

– лематизація: це процес приведення слів до їхньої канонічної або словникової форми з урахуванням їхньої частини мови та контексту. Наприклад, слова «is», «are», і «were» можна лематизувати до «be». Лемматизація схожа на лематизацію, але є більш точною та витонченою;

– видалення стоп-слів: це процес видалення слів, які є дуже поширеними та не додають тексту великого сенсу чи інформації. Наприклад, «the», «a», «and» тощо. Видалення стоп-слів допомагає зменшити шум і розмір тексту та зосередитися на важливих словах;

– видалення пунктуації: це процес видалення з тексту знаків пунктуації, таких як коми, крапки, знаки питання тощо. Видалення розділових знаків допомагає прибрати непотрібні символи та зробити текст більш чистим і простим;

– орфографічне виправлення: це процес виправлення орфографічних помилок або друкарських помилок у тексті. Орфографічне виправлення допомагає поліпшити якість і читабельність тексту, а також уникнути плутанини або непорозумінь.

3.5 Гіперпараметри моделі

Для того, щоб перевести текстову послідовність у чисельний вигляд, необхідно провести аналіз даних, оскільки матричний вигляд вимагає фіксації даних у єдиній формі з готовими вимірами. Оскільки на вході ми керуємо переведенням слів у тензорний простір, ми можемо самостійно задати розмірність цих векторів. Для визначення відповідної розмірності необхідно позначити довжини послідовностей, які надходитимуть на вхід.

Для цього було проведено статистичний аналіз. Середня довжина тексту статті склала 242 слова, середня довжина заголовка статті склала 11 слів. Медіана для тексту дорівнює 226 слів, а для заголовка – 11 слів. Середньоквадратичне відхилення для заголовка виявилося таким, що дорівнює 2,42, для тексту воно дорівнює 112,73, і це дає нам змогу скористатися емпіричним правилом, згідно з яким майже всі дані можуть бути покриті одним діапазоном значень плюс-мінус три відхилення (для вибірок із нормальним розподілом). Таким чином ми можемо встановити максимальну довжину статті та заголовка рівними 578 і 18 слів. Однак не буде зайвим округлити ці числа в більший бік до 700 і 25 відповідно. Крім того, що це дасть змогу покрити більшу кількість даних, це зможе допомогти з подальшим донавчанням на вибірках, середні довжини в яких можуть бути трохи більшими. У цій моделі в разі перевищення текстовими даними цих значень, зайві текстові дані відсікаються.

Рисунки 3.2 і 3.3 представляють гістограми розподілу довжин тексту статті та заголовка статті: на осі абсцис – кількість слів, на осі ординат – кількість текстів.

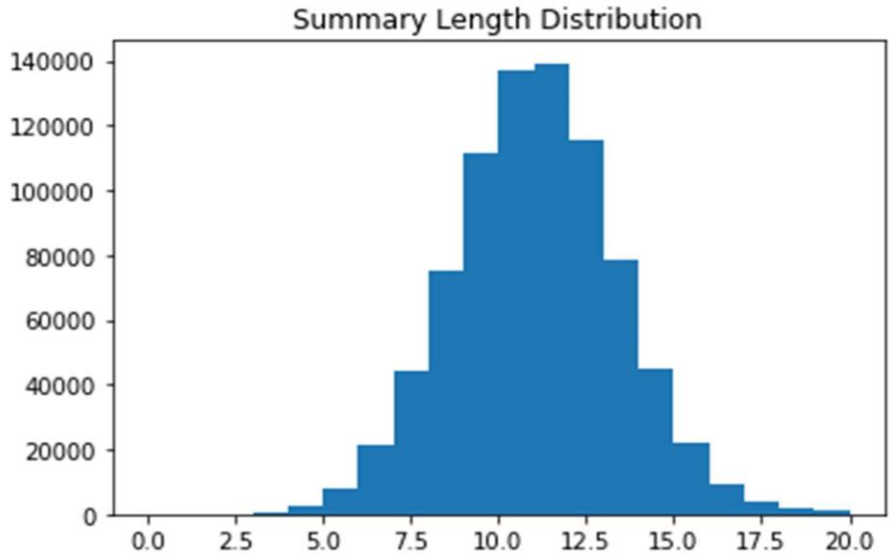


Рисунок 3.2 – Гістограма розподілу довжин заголовків заголовків новини

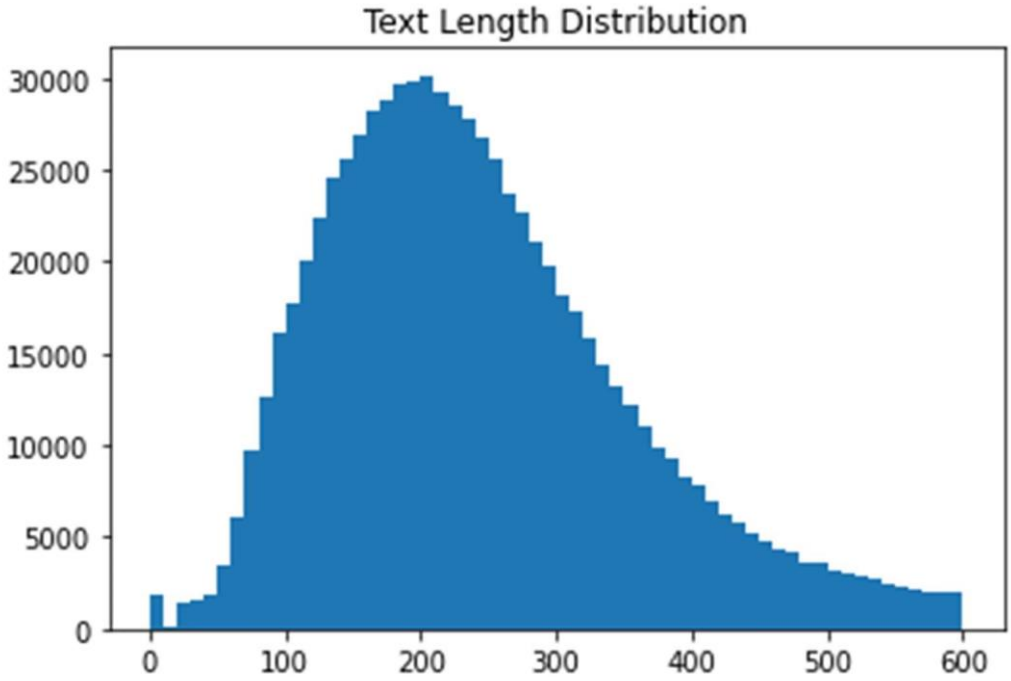


Рисунок 3.3 – Гістограма розподілу довжин текстів новини

Інші гіперпараметри були такими: розмірність вектора 512, як оптимізатор було обрано AdamW з кількістю warmup-кроків 500, валідація відбувалася на кожній епосі навчання. З метою прискорення навчання словник моделі був обмежений 40 000 словами.

Для навчання моделі було використано відеокарту, що надається сервісом Google Colab, Tesla T4. Навчання тривало 12 годин.

3.6 Оцінка отриманих результатів

Для емпіричного порівняння метрики для оцінювання якості моделі було обрано ті самі, що й у попередників, і способи оцінювання були схожими.

У процесі донавчання модель досягла помилки, що дорівнює 1,9134 на тренувальній вибірці та 2,3822 на валідаційній. На рисунку 3.4 наведено графік зміни помилки на тренувальній і валідаційній вибірці.

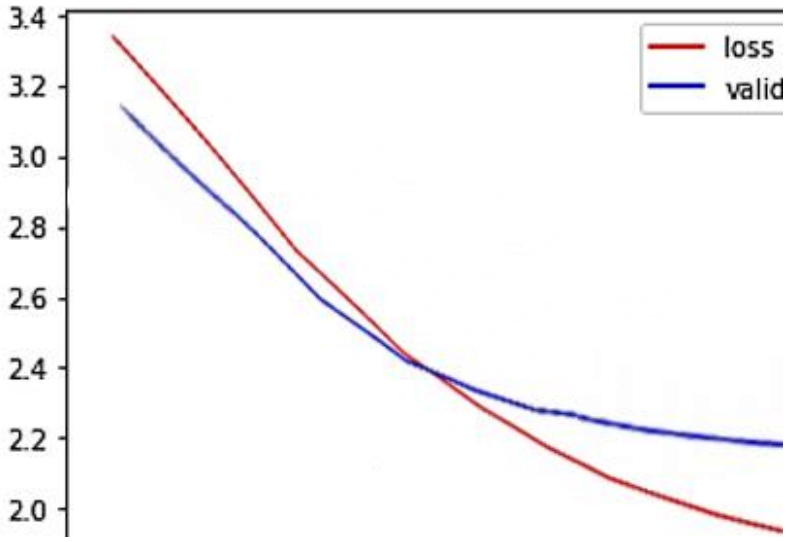


Рисунок 3.4 – Графік залежності помилки від часу

Оцінка результатів за допомогою варіації метрики ROUGE, середнього між F-мірами ROUGE-1, ROUGE-2 і ROUGE-L, модель досягла значення 0.15744886179299, що перевершує отриманий раніше результат 0.05800769685750703, проте не може бути порівнянний з результатами, отриманими більшими лабораторіями та користувачами, наприклад, у рамках змагання з генерації заголовків у рамках конференції Діалог-2019, результат якого наведено в роботі Шевчука – 0.23141523643530523.

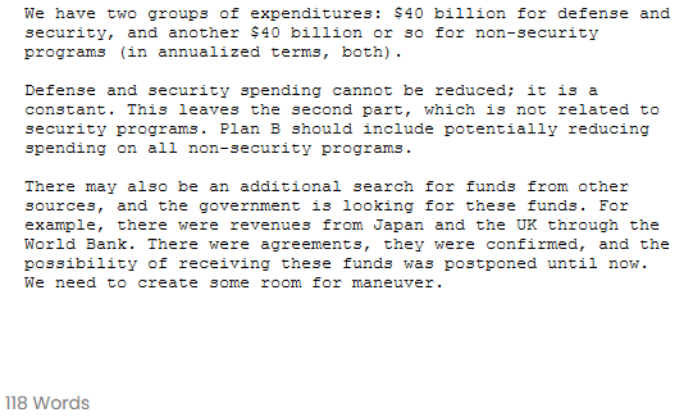
Приклади аналізу статей за заголовками наводяться нижче в (таблиці 3.1). Цей алгоритм має адаптування для роботи з текстовими графами -TextRank, який є працює за такою самою схемою як і оригінальний алгоритм.Вершинами графа є текстові структури (речення, слова, абзаци, документи, тощо). Ребрами текстового графа є залежності або зв’язки цих текстових сутностей між собою. Наприклад, схожість за сенсом, лексичні чи семантичні відносини, довжина найдовшої загальної підпослідовності, тощо.

Таблиця 3.1 – Приклади сумаризації заголовків новин

Текст, отриманий моделлю на введення	Оригінальний заголовок	Згенерований заголовок
Пасажир Харківського метрополітену заплатить штраф на суму 20 тисяч гривень за відмову пройти огляд багажу. ...	Пасажир метро заплатить 20 тисяч гривень за відмову від огляду	Штраф за відмову пройти огляд
У кожному муніципалітеті Київської області діє програма з відновлення святинь, загалом в області налічується близько двох тисяч православних храмів, заявив губернатор Київської області Кравченко Руслан.	Кравченко відзначив роботу з відновлення святинь у регіоні	Кравченко: у кожному муніципалітеті Києва діє відновлення
Нападник туринського "Ювентуса" Кріштіану Роналду визнав себе винним у справі про ухилення від сплати податків в Іспанії. Про це повідомляє AS. За інформацією видання, форвард пішов на угоду зі слідством і погодиться на дворічний умовний термін.	Роналду, який виплатив 13 мільйонів штрафу, знову опинився в суді	роналду зізнався в любові до…

Далі проведемо аналіз блоку новин. Наші методи узагальнення спираються на попередні роботи з кодувально-декодувальними нейронними мережами з LSTM та механізмами уваги. Одним із прикладів є робота Лопір'єва [3], який використовував кодувально-декодувальні рекурентні нейронні мережі з LSTM та увагою для генерації заголовків новинних статей. Це досягається шляхом подачі набору даних Gigaword як вхідного тексту новинної статті на кодер, по одному слову за раз. Кожне слово перетворюється на подання розподілу. Потім подання розподілу комбінується за допомогою багатошарової нейронної мережі зі прихованими шарами, згенерованими після подачі попереднього слова. На наступному кроці приховані шари потрапляють до декодера. Враховуючи

високу складність розуміння та узагальнення структури новинних статей, ми також використовували концепції, запозичені з робіт К. Лопір'єва, для створення ефективної моделі сумаризації. Принципове значення має здатність моделі розрізняти та адаптуватися до різних типів інформації, що містяться у текстах новин, щоб надати точні та компактні резюме. Наш підхід полягає у створенні моделі, приклад якої наведений на рисунку 3.5 та 3.6, яка вміє адекватно аналізувати контент новин та ефективно виділяти ключову інформацію.


We have two groups of expenditures: $40 billion for defense and security, and another $40 billion or so for non-security programs (in annualized terms, both).

Defense and security spending cannot be reduced; it is a constant. This leaves the second part, which is not related to security programs. Plan B should include potentially reducing spending on all non-security programs.

There may also be an additional search for funds from other sources, and the government is looking for these funds. For example, there were revenues from Japan and the UK through the World Bank. There were agreements, they were confirmed, and the possibility of receiving these funds was postponed until now. We need to create some room for maneuver.

118 Words

Рисунок 3.5 – Повний текст статті поданий на вхід

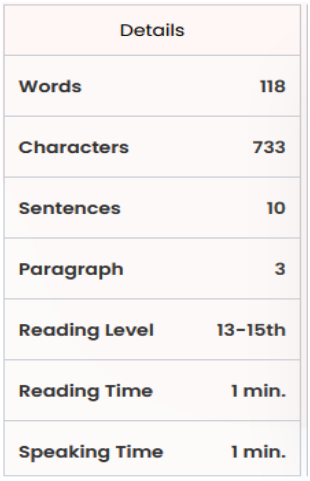

Details	
Words	118
Characters	733
Sentences	10
Paragraph	3
Reading Level	13-15th
Reading Time	1 min.
Speaking Time	1 min.

Рисунок 3.6 – Параметричний аналіз тексту та виведення в табличній формі

Спочатку ми познайомилися з моделлю довготривалої короткострокової пам'яті (LSTM) – рекурентною нейронною мережею, яку ми використали як для кодера, так і для декодера. Ми використовуємо модель LSTM кодера-декодера з механізмом уваги для аналізу заголовка та повного тексту новинної статті. Модель кодера-декодера – це модель від послідовності до послідовності, яка широко використовується в таких завданнях, як машинний переклад [10], чат-бот [11] та узагальнення тексту [3]. Наша робота тісно пов'язана з роботою [3], в якій автор також використовував кодер-декодерну LSTM-модель з механізмом уваги для генерації заголовка. Найбільші відмінності між нашою роботою та [3] полягають у двох аспектах. По-перше, ми використовуємо п'ять відер і оптимізували довжину відведення, щоб звести до мінімуму кількість прокладок, використаних у реченнях, тоді як у [3] використовується лише одне відро. По-друге, для генерації заголовкових слів ми використовуємо жадібний декодер, а не декодер з пошуком по променю для генерації заголовків. Ця модель складається з двох частин. Перша частина – це LSTM-кодер, який кодує вхідне речення «Ця стаття». Друга частина складається з декодуючого LSTM, який генерує вихідне речення «Цей заголовок». У цій роботі ми використовуємо перше речення новинної статті як вхідне, а відповідний заголовок новини – як вихідне. Ми подавали вхідні слова у зворотному напрямку кодеру так, щоб перші кілька слів у вхідному реченні були ближчими до перших кількох вихідних слів, щоб зафіксувати короткострокові залежності. Реверсування вхідного речення показало кращі результати в [10] у завданні машинного перекладу. Після генерування кожне передбачене слово буде подано як вхідне під час генерування наступного слова. «This» – це перше згенероване слово, яке буде подано на вхід, щоб спрогнозувати наступне слово «заголовок». Механізм уваги полягає в тому, щоб надавати ваги як вихідним прихованим станам у кодері, так і поточному прихованому стану, щоб вирішити, на яке слово звернути увагу,

передбачаючи поточне заголовкове слово. Механізм уваги реалізовано в [10] як декодер для машинного перекладу та в [8] як кодер для узагальнення речень.

Ми використали бібліотеку TensorFlow, щоб реалізувати нашу модель LSTM кодера-декодера з механізмом уваги. Наш код побудований на основі моделі «послідовність-послідовність» з TensorFlow та репозиторію GitHub, в якому автор також використовував модель «послідовність-послідовність» для побудови чат-бота. Зокрема, ми використовуємо три приховані шари LSTM як у кодерній, так і в дешифрувальній моделях, коефіцієнт вихідного вибуття між шарами – 0,2, 512 прихованих одиниць на комірку LSTM, а також навчили вбудовувати в них вбудовування слів до 512 вимірів. Ми вбудували найчастіші 80 000 слів як у реченнях, так і в заголовках, а всі інші слова позначили невідомими словами (UNK) або, як правило, словами, що не належать до словника (OOV). Нарешті, ми використовуємо п'ять відер ((30, 10), (30, 20), (40, 10), (40, 20), (50, 20)) (тобто п'ять різних LSTM-моделей кодерів-декодерів), що ґрунтуються на статистичних даних, отриманих у розділі 3.3, щоб зменшити кількість маркерів для відступів. У представленні відра (*a,b*), *a* означає довжину речення, а *b* – довжину заголовка. Якщо речення довше 50 лексем або заголовок довше 20 лексем, ми автоматично поміщаємо його в останнє ((50,20)) відро ((50,20)). Ми використовуємо спеціальний лексем (PAD), щоб підгортати як речення, так і заголовки до розміру відра. Інтеграція зазначених підходів дозволила досягти високої точності та ефективності в завданні сумаризації текстів. Наші дослідження демонструють потенціал для подальшого вдосконалення та розвитку моделей сумаризації на основі нейронних мереж (рисунок 3.7).

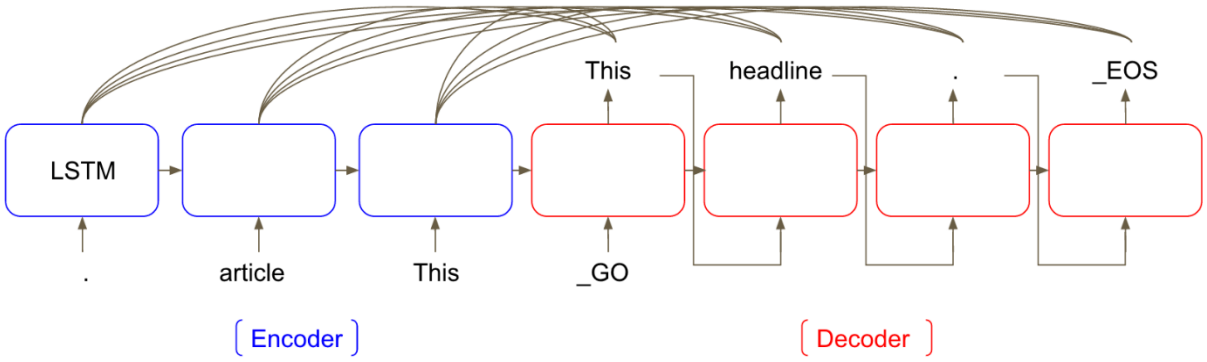


Рисунок 3.7 – Опис моделі LSTM

У нашому дослідженні ми реалізували LSTM-модель кодера-декодера. Для оцінки результатів запропонованої моделі ми використовували як системно-орієнтовану автоматичну оцінку, так і оцінку, орієнтовану на користувача, і порівнювали результати. Для оцінки користувачів ми створили однофакторне внутрішньо-суб'єктне опитування оцінки дизайну. Фактором були б методи генерації резюме: фактичне резюме та резюме, згенероване за допомогою моделі LSTM. У нашому експериментальному дизайні ми використовуємо Google-форму для розробки та розповсюдження опитування.

У цьому дослідженні було здійснено створення моделі нейронної мережі на основі архітектури Transformer для розв'язання задачі генерації новинних заголовків. Мета, поставлена нами у вступі, була досягнута.

Було виконано й завдання, поставлені для виконання цієї мети. Було вивчено наявні автоматичні сумаризатори та встановлено їхні переваги та недоліки. Було детально розглянуто архітектуру нейронної мережі Transformer і архітектуру EncoderDecoder, що їй передувала, на прикладі моделі Seq2seq. Для практичного застосування моделі було здійснено збір текстових даних у єдиний датасет, виконано його передобробку. На основі аналізу вибірки було отримано підстави для завдання гіперпараметрів.

ВИСНОВКИ

У цьому дослідженні було здійснено створення моделі нейронної мережі на основі архітектури Transformer для розв'язання задачі генерації новинних заголовків. Мета, поставлена нами у вступі, була досягнута.

Було виконано й завдання, поставлені для виконання цієї мети. Було вивчено наявні автоматичні сумаризатори та встановлено їхні переваги та недоліки. Було детально розглянуто архітектуру нейронної мережі Transformer і архітектуру EncoderDecoder, що їй передувала, на прикладі моделі Seq2seq. Для практичного застосування моделі було здійснено збір текстових даних у єдиний датасет, виконано його переобробку. На основі аналізу вибірки було отримано підстави для завдання гіперпараметрів.

Виконані теоретична і частина роботи дає змогу припустити, що обрана нами модель може бути використана для розв'язання задачі сумаризації на прикладі генерації новинних заголовків українською мовою. Однак ще належить зробити висновки про причини як низького формального показника, так і суб'єктивно низької якості згенерованих заголовків.

Припущення, висловлене в роботі про те, що міжтекстовий зв'язок поліпшується введенням механізму уваги, не підтверджується спостереженнями, але, як було зазначено у висновках до розділу 3, алгоритм векторизації, що працює з підрядковими частинами тексту, дійсно ефективно працює.

Поле варіантів для майбутньої роботи все ще видається величезним, проте найперспективнішим напрямком роботи вбачається збільшення кількості слів у словнику, оперативної пам'яті та масштабування обчислювальних ресурсів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1) Sequential latent Dirichlet allocation / L. Du et al. *Knowledge and Information Systems*. 2011. Vol. 31, no. 3. P. 475–503. URL: https://doi.org/10.1007/s10115-011-0425-1 (Дата звернення: 01.05.2024).

2) Labruna T., Magnini B. Fine-Tuning BERT for Generative Dialogue Domain Adaptation. *Text, Speech, and Dialogue*. Cham, 2022. P. 513–524.

3) Огурцова О., Шевченко О. НЕЙРОННИЙ МАШИННИЙ ПЕРЕКЛАД: ОСНОВНІ ПРИНЦИПИ, СИЛЬНІ ТА СЛАБКІ СТОРОНИ. *European Science*. 2023. Sge18-04. С. 86–92.

4) Baxendale P. B. Machine-Made Index for Technical Literature–An Experiment. *IBM Journal of Research and Development*. 1958. Vol. 2, no. 4. P. 354–361.

5) News Text Classification Model Based on Topic Model. International Journal of Recent Trends in Engineering and Research. 2017. Vol. 3, no. 7. P. 48–52. URL.

6) Agrawal A., Singh O. Large Scale Short Text Analysis to Recognize Categories. *International Journal of Computer Sciences and Engineering*. 2019. Vol. 7, no. 5. P. 1873–1877.

7) Чуриков М., Саннікова Е. Генерація заголовків: перше речення проти нейромашинного перекладу // Комп'ютерна лінгвістика та інтелектуальні технології. 2019.

8) Clark K. Напівсупервізоване моделювання послідовності за допомогою перехресного навчання // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018.

9) Conroy J.M., O'leary D.P. Summarization of Text via Hidden Markov Models and Pivoted QR Matrix Decomposition // Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. 2001.

10) Skurzhanskyi O. H., Marchenko O. O., Anisimov A. V. SPECIALIZED PRE-TRAINING OF NEURAL NETWORKS ON SYNTHETIC DATA FOR IMPROVING PARAPHRASE GENERATION. *KIBERNETYKA TA SYSTEMNYI ANALIZ*. 2024. P. 3–12.

11) Бук С. Н. Основи статистичної лінгвістики : навч.-метод. посіб. Львів : Львів. нац. ун-т ім. І.Франка, 2008. 124 с.

12) Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre- training for Natural Language Generation, Translation, (2019).

13) Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, WeiLi, PeterJ. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. (2019).

14) Graham Y. Переоцінка автоматичного узагальнення за допомогою BLEU і 192 відтінків ROUGE // Матеріали конференції – EMNLP 2015: Conference on Empirical Methods in Natural Language Processing. 2015. № вересень. С. 128-137.

15) A comprehensive survey on transfer learning / F.Zhuang, Z.Qi, K.Duan, D.Xi, Y.Zhu, H.Zhu, H.Xiong, Q.He//Pro cedings of the IEEE.—2020. т.109,No1, с.43–76.4.

16) Text Categorization using Support Vector machines / A. Cortez Vasquez et al. *Revista de investigación de Sistemas e Informática*. 2013. Vol. 10, no. 1. P. 33–44.

17) N. L. P. NLP Comprehensive. NLP : New Technology: The New Technology. HarperCollins Publishers, 2011.