Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему *Метод генерації короткого змісту тексту з використанням нейронної мережі*

Галузь знань *12 – Інформаційні технології*
Шифр і назва галузі знань

Спеціальність *122 – Комп'ютерні науки*
Шифр і назва спеціальності

Освітня програма *Комп'ютерні науки*
Назва освітньої програми

Виконав: *студент 4 курсу, група КН-20-1* _____ *Олександр БОНДАР*
Курс, група виконавця — Підпис — Ініціали, прізвище

Керівник: *д.т.н., професор кафедри КН* _____ *ЕдуардМАНЗЮК*
Науковий ступінь, посада — Підпис — Ініціали, прізвище

Нормоконтроль: *к.т.н., доцент кафедри КН* _____ *Руслан БАГРІЙ*
Науковий ступінь, посада — Підпис — Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КН,д.т.н., професор _____ *Олександр БАРМАК*
Підпис — Ініціали, прізвище

18 06 2024 р.

Хмельницький 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет *інформаційних технологій*
Кафедра *комп'ютерних наук*
Освітній ступінь *бакалавр*
Галузь знань *12 – Інформаційні технології*
Спеціальність *122 – Комп'ютерні науки*
Освітня програма *освітньо-професійна програма підготовки бакалавра*

ЗАТВЕРДЖУЮ
Завідувач кафедри *комп'ютерних наук*

(підпис)
д.т.н., професорОлександр БАРМАК
«16»02*2024* року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: *«Метод генерації короткого змісту тексту з використанням нейронної мережі»*

2. Завдання видано студенту *Олександру БОНДАРУ*
(прізвище, ім'я, по батькові)

3. Керівник роботи *д.т.н.,професор кафедри КНЕдуард МАНЗЮК*
(посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від «15»02*2024* р. № 8

5. Дата видачі завдання студенту: «16»02*2024*р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Метою кваліфікаційної роботи бакалавра є покращення генерації короткого змісту тексту для формування заголовків, назв, коротких виразних фраз. Для досягнення мети: проведено аналіз предметної області, оглянуті методи, які використовуються для генерації короткого змісту тексту з використанням нейронної мережі;розробленометод генерації короткого змісту тексту з використанням нейронної мережі;оцінена ефективність застосування цього методу для генерації короткого змісту тексту;проведенаевпериментальна перевіркаефективностізапропонованого методу для генерації короткого змісту тексту з використанням нейронної мережі.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	*Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником*	*грудень 2023*	*виконано*
2	*Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження*	*січень 2024*	*виконано*
3	*Робота над розділом 1 – Характеристика предметної області та постановка задачі*	*січень 2024*	*виконано*
4	*Робота над розділом 2 – Метод генерації короткого змісту тексту з використанням нейронної мережі*	*березень 2024*	*виконано*
5	*Робота над розділом 3 – Експериментальна перевірка методу генерації короткого змісту тексту з використанням нейронної мережі*	*квітень 2024*	*виконано*
6	*Оформлення пояснювальної записки згідно вимог*	*травень 2024*	*виконано*
7	*Попередній захист кваліфікаційної роботи бакалавра*	*травень 2024*	*виконано*
8	*Захист кваліфікаційної роботи бакалавра на засіданні Екзаменаційної комісії*	*червень 2024*	*виконано*

Виконавець: *студент 4 курсу, група КН-20-1* _____ *ОлександрБОНДАР*
Курс, група виконавця — Підпис — Ініціали, прізвище

Керівник: *д.т.н. професор кафедри КН* _____ *Едуард МАНЗЮК*
Науковий ступінь, посада — Підпис — Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: *Метод генерації короткого змісту тексту з використанням нейронної мережі.*

Виконавець кваліфікаційної роботи бакалавра: *студент групи КН-20-1 Олександр БОНДАР*

Керівник кваліфікаційної роботи бакалавра: *д.т.н. професор кафедри КН Едуард МАНЗЮК*

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
74	8	2	40	1

Мета кваліфікаційної роботи бакалавра полягає в покращенні генерації короткого змісту тексту для формування заголовків, назв, коротких виразних фраз.

Для досягнення поставленої мети визначені наступні задачі дослідження необхідно: визначити послідовність застосування методугенерації короткого змісту тексту з використанням нейронної мережі; розробити методгенерації короткого змісту тексту з використанням нейронної мережі; провести екпериментальні дослідження ефективності розробленого методу.

Результатом виконання кваліфікаційної роботи бакалавра є розробленийметод генерації короткого змісту тексту з використанням нейронної мережі.

Ключові слова: генерація тексту, аналіз тексту, нейронні мережі, трансформери.

Виконавець: *студент 4 курсу, група КН-20-1* _____ *ОлександрБОНДАР*

Курс, група виконавця — Підпис — Ініціали, прізвище

Зміст

Перелік скорочень

Скорочення, термін, позначення	Пояснення
КРБ	Кваліфікаційна робота бакалавра
КН	Комп’ютерні науки
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
ОПМ	Обробка природньої мови
LSTM	Long Short-Term Memory

Вступ

Кваліфікаційна робота бакалавра присвячена розробці методу генерації короткого змісту тексту для формування заголовків, назв, коротких виразних фраз.

Генерація короткого змісту тексту є важливим завданням в області обробки природної мови. У сучасному інформаційному суспільстві кількість текстових даних стрімко зростає, що створює необхідність в ефективних методах їх аналізу та узагальнення. Нейронні мережі, особливо моделі на основі трансформерів, демонструють високу ефективність у вирішенні задач автоматичної сумаризації тексту завдяки своїй здатності враховувати контекст та семантику тексту.

Традиційні методи сумаризації тексту, такі як статистичні методи та методи на основі правил, часто не можуть забезпечити високу якість результатів через обмеженість у розумінні глибоких семантичних зв’язків у тексті. Статистичні методи зазвичай базуються на частотних характеристиках слів і фраз, що не завжди відображає дійсну важливість інформації в контексті. Методи на основі правил, в свою чергу, залежать від заздалегідь визначених шаблонів та евристик, що може призводити до втрати суттєвих деталей і нездатності адаптуватися до нових, неочікуваних текстових структур.

Використання нейронних мереж дозволяє подолати ці обмеження, забезпечуючи більш точне та контекстно обґрунтоване узагальнення. Нейронні мережі, зокрема моделі на основі трансформерів, здатні враховувати взаємозв'язки між словами у тексті завдяки механізму самоуваги. Це дозволяє моделям глибше розуміти контекст і значення кожного слова в тексті, що сприяє створенню більш якісних резюме.

Це особливо актуально в умовах великих обсягів даних, де швидкий доступ до ключової інформації стає критично важливим. У таких випадках автоматичної сумаризації допомагає користувачам швидко отримати уявлення про зміст документів, що економить час і підвищує ефективність роботи з

інформацією. Крім того, це сприяє поліпшенню пошукових систем та систем рекомендацій, дозволяючи їм надавати більш релевантні результати на основі узагальнених змістів документів.

Враховуючи вищезазначене, розвиток методів автоматичної сумаризації тексту з використанням нейронних мереж є важливим напрямком досліджень у сфері текстових повідомлень. Це дозволяє не лише підвищити якість аналізу текстових даних, але й сприяє створенню більш інтелектуальних і адаптивних інформаційних систем, здатних ефективно працювати з великими обсягами текстової інформації.

Розроблений метод ефективним у практичній реалізації. Застосування може бути користним для автоматичного формування коротких повідомлень.

Об’єкт дослідження – процес генерації короткого змісту тексту з використанням нейронної мережі.

Предмет дослідження – методи та технології машинного навчання для роботи з текстовою інформацією.

Мета кваліфікаційної роботи бакалавра – покращення генерації короткого змісту тексту для формування заголовків, назв, коротких виразних фраз.

Завдання кваліфікаційної роботи бакалавра.

Для досягнення цієї мети виконуються наступні завдання:

– провести аналіз відомих методів для автоматичної генерації заголовків, назв та коротких виразних фраз на основі аналізу текстових даних;

– розробити метод, здатного навчатися на великому наборі текстових даних для генерації короткого змісту з урахуванням контексту та семантики тексту;

– здійснити оптимізацію параметрів нейронної мережі для покращення її якості та точності в завданнях генерації короткого змісту тексту;

– провести експериментальне тестування розробленого методу на наборі текстових даних для оцінки його ефективності та точності.

Розділ 1 Характеристика предметної області та постановка задачі

1.1. Аналіз предметної області генерації короткого змісту тексту та його узагальнення

Сумаризація тексту являє собою процес стиснення довгого тексту до коротшої форми, зберігаючи при цьому його основний зміст та ключові ідеї. Іншими словами, це перетворення великого обсягу тексту на компактний або скорочений варіант, який передає сутність інформації [1–5].

Замість слова сумаризація можна використати такі синоніми, як "узагальнення" або "конденсація. Такі вирази також відображають ідею стиснення тексту до більш компактної форми, що залишається інформативним.

Сумаризація тексту є важливим інструментом обробки природної мови, який дозволяє автоматично виділяти головні пункти або ключові ідеї з текстових джерел. Цей процес включає в себе аналіз тексту, ідентифікацію важливих інформаційних елементів і утворення стислого узагальнення.

В сучасній обробці природної мови, сумаризація тексту використовується для різноманітних цілей, таких як:

– екстрактивна сумаризація. Підхід до сумаризації включає виділення найважливіших речень або фраз з вихідного тексту і створення узагальнення, використовуючи ці вилучені елементи. Екстрактивна сумаризація зазвичай базується на статистичних методах аналізу тексту, таких як обчислення важливості речень за допомогою метрик та алгоритмів [6–8];

– абстрактна сумаризація. Підхід полягає у створенні нового короткого тексту, який передає основний зміст вихідного матеріалу, але перефразує його з використанням машинного навчання та природної мови. Абстрактна сумаризація створює більш адаптований текст, який не просто виділяє окремі фрагменти, а перетворює їх, зберігаючи сутність [9–12].

Сумаризація тексту використовується в різних сферах, таких як пошукові системи, аналіз новин, автоматичне реферування наукових статей, а також в інших областях, де необхідно ефективно обробляти великі обсяги інформації. Застосування сучасних методів машинного навчання та обробки природної мови

дозволяє покращити точність і якість автоматичної сумаризації тексту, зробивши її більш ефективною та корисною для різних завдань.

Сумаризація тексту важлива задача в обробці мови, але традиційні методи мають складнощі з адаптацією до потреб сучасного пошуку інформації. Тому автоматичний підсумовувач стає все більш важливим. Розвиток трансформерів, таких як BART [13–15], T5 [16–18], ProphetNet [19–21], має великий потенціал. Однак вибір найкращої моделі з численних варіантів стає новою проблемою, не достатньо дослідженою. Начастіше використовують трансформерів для застосування в сфері прктичного застосування щодо створення крроткого змісту тектсту. Результати оцінюють за допомогою ROUGE та розміру навчальної вибірки.

Трансформери дійсно виявляються хорошим інструментом у роботі з мовою завдяки їхній здатності до контекстуального розуміння та генерації тексту. Однак, однією з їхніх особливостей є потреба в значній кількості даних для ефективного навчання та точного налаштування. Це пов'язано з великою кількістю параметрів, які вони мають, та складністю самого алгоритму.

Загалом, дослідження показують, що трансформери найбільш ефективні при навчанні на великих обсягах різноманітних даних. Це пояснює частково, чому трансформери виявляються особливо ефективними у задачах обробки природної мови, таких як машинний переклад, розпізнавання мови та автоматичне резюмування тексту.

Таким чином, хоча великі обсяги даних можуть бути вимогою для оптимальної роботи трансформерів, але при належному навчанні вони виявляються дуже потужними інструментами у сфері обробки мови. Одним із ключових завдань у сфері обробки природної мови є узагальнення довгих текстів у короткі, зберігаючи основну ідею незмінною. Цей процес відомий як текстова сумаризація і має велике значення в сучасному світі, де обсяг інформації швидко зростає, особливо в мобільному Інтернеті та популярних відеоплатформах.

Традиційні методи сумаризації тексту стикаються з викликом адаптації до сучасних вимог до пошуку інформації. Тому наявність надійного та точного

автоматичного інструменту для узагальнення текстів стає все важливішою у сучасній інформаційній епохи.

Традиційні методи сумаризації тексту, такі як статистичні або правила-засновані підходи, стикаються з рядом викликів у контексті сучасних вимог до пошуку інформації:

Традиційні методи не завжди можуть ефективно увіраховувати контекст тексту, що призводить до менш точних та релевантних резюме. У порівнянні з сучасними нейронними мережами, які здатні до контекстуального розуміння, традиційні методи можуть залишатися обмеженими.

Сучасні вимоги до сумаризації тексту включають не лише здатність до розуміння мови, а й до адаптації до різних стилів письма, жаргонних виразів та семантичних відтінків. Традиційні методи можуть бути менш ефективними у цьому відношенні.

Сучасні методи сумаризації повинні бути гнучкими та масштабованими, щоб працювати з різноманітними джерелами тексту та великими обсягами даних. Традиційні підходи можуть виявлятися обмеженими у цьому відношенні через свою обмежену здатність до адаптації та обробки великих обсягів інформації.

Винахід трансформаторів відкриває нову еру дослідження обробки природної мови. Після появи BART тисячі покращених моделей навчаються або налаштовуються для виконання різних завдань ОПМ, таких як узагальнення текстів і переклад. Сімейство трансформерів майже повністю переважає, навіть пізніше створені моделі 2018 року з архітектурою LSTM і GRU не можуть перевершити оригінальний трансформер, представлений у 2017 році із рейтингом [22–25]. Добре налаштовані моделі трансформерів показують великий потенціал у завданнях узагальнення ОПМ.

Є значний потенціал у швидкому розвитку сімейства трансформаторів. За допомогою тонкого налаштування, можемо використовувати всю потужність цієї передової технології. Проте на практиці стикаємося з рядом обмежень, таких як обмежений доступ до даних або високі витрати на обчислення, що ускладнює

навчання моделей так ефективно, як це роблять професіонали з обробки даних, які працюють у великих обчислювальних лабораторіях. Крім того, з'являється все більше моделей для навчання, оскільки багато людей переходять на трансформатори та вдосконалюють їх внутрішню архітектуру, що робить вибір найкращої моделі для конкретного завдання узагальнення дедалі складнішим.

Кваліфікаційна робота має на меті реалізацію декількох популярних моделей на основі трансформаторів, з подальшим визначенням їх характеристик і властивостей шляхом тонкого налаштування та потокової передачі для узагальнення кількох основних наборів даних. Після цього проводитиметься оцінка їх ефективності за допомогою метрик ROUGE. Ця оцінка дозволить з'ясувати, які результати можна очікувати в реальних умовах і як вони відрізняються від результатів, згаданих у літературі, а також допоможе визначити, яка модель підходить для конкретних умов, таких як обмеженість навчальних даних, обмеження часу на навчання або обчислювальні обмеження.

Автоматичне узагальнення полягає в автоматичній конденсації великого обсягу даних до більш короткої підмножини тексту або перефразування, що відображає найважливіше значення. У цій області існують дві основні парадигми, які було згадано вище. Розглянемо більш детальніше.

Екстрактивне узагальнення використовує підрахунок балів для виділення речень з вихідного документа та їх об'єднання у логічне резюме. Цей підхід базується на ідентифікації ключових фрагментів тексту, їх виділенні та створенні компактного огляду. Екстрактивне узагальнення ґрунтується на витягуванні фраз з оригінального тексту. Більшість сучасних досліджень у цій галузі зосереджені на екстрактивному узагальненні, яке є простішим і створює природні резюме з мінімальною обробкою тексту. Такі узагальнення включають найважливіші речення з вихідного тексту, якими можуть бути як окремі документи, так і колекції статей.

Абстрактні методи узагальнення спрямовані на створення резюме шляхом аналізу тексту з використанням методів машинного навчання і природної мови для створення нового, коротшого огляду, що містить ключову інформацію з

оригінального тексту. Ці методи перефразують речення та включають інформацію з повного тексту, створюючи резюме, схоже на те, що може створити людина при написанні реферату. У хорошому абстрактному резюме враховуються лінгвістичні аспекти і відтворюються всі важливі деталі з вихідного тексту, не обмежуючись лише витягуванням та перестановкою фрагментів оригінального тексту.

У реферативному процесі застосовуються глибокі нейронні мережі для аналізу тексту та створення короткого абстрактного огляду, який відображає головну інформацію з оригінального матеріалу. Нейронні мережі "від послідовності до послідовності" використовуються для вирішення завдання генерації тексту з врахуванням контексту та послідовності.

У контексті машинного навчання передбачається, що дані розподіляються згідно з поняттям незалежно та ідентично, що означає, що кожен приклад даних вважається незалежним від інших. Проте у випадках, коли маємо справу з послідовними даними, такими як текст, голос або дані часових рядів, кожен елемент даних може залежати від попередніх елементів у послідовності. Такий тип даних часто описується як послідовність, де кожен елемент має контекст і зв'язок з попередніми і наступними елементами.

В мові, наприклад, фрази складаються зі слів, які розміщені в певному порядку, що має значення. Порядок слів у фразі впливає на її смислове навантаження, оскільки фрази можуть мати різний сенс в залежності від того, як вони складені. Тому послідовність слів у мовленні відображає логічну структуру та смислові зв'язки між ними.

Таким чином, робота з послідовними даними у природній мові потребує врахування їхньої структури та послідовності для ефективного аналізу та моделювання за допомогою методів машинного навчання, зокрема глибоких нейронних мереж.

1.2 Методи генерації короткого змісту тексту

В глибокому навчанні моделювання послідовностей включає відстеження прихованого стану, що є інформацією про стан. Цей прихований стан оновлюється при обробці кожного елемента послідовності, такого як слова у фразі. В результаті вектор прихованого стану зберігає в собі усю інформацію, що була зібрана з послідовності до поточного моменту. Це представлення послідовності може використовуватись для різних завдань моделювання послідовностей, включаючи категоризацію і прогнозування, в залежності від поставленої задачі.

Почнемо з огляду найпростішої моделі нейронних мереж для обробки послідовностей – рекурентної нейронної мережі (РНН). Започаткована рекурентна нейронна мережа є одним з перших методів глибокого навчання, призначеним для моделювання послідовностей. Основна ідея полягає в тому, щоб мати механізм, який може обробляти послідовні дані, такі як послідовність слів у тексті або послідовність часових кроків у часовому ряді [26–30].

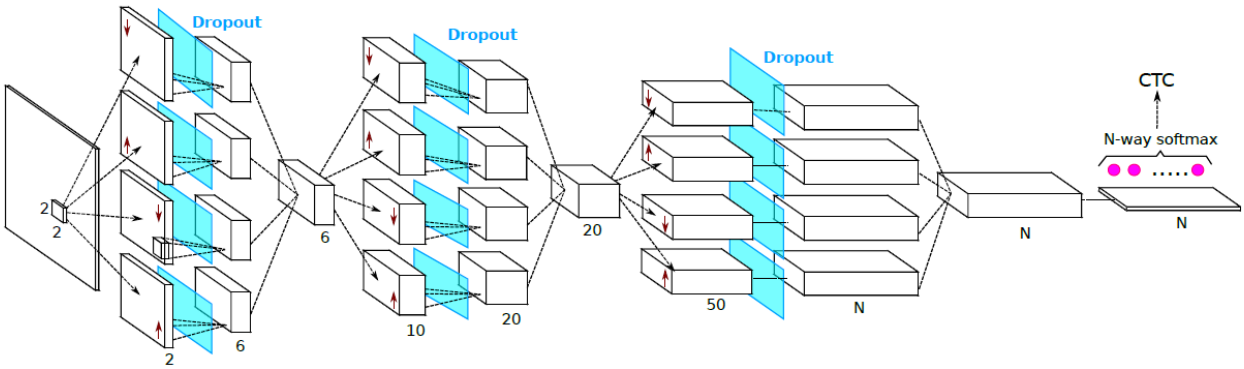


Рисунок 1.1 – Архітектура MDLSTM-RNN [26]

Основною складовою рекурентної нейронної мережі є її рекурентний шар рекурентний вузол або рекурентний блок. Цей шар дозволяє моделі використовувати попередній вихід або прихований стан для обробки поточного вхідного елементу послідовності. Кожен крок обробки у рекурентній мережі

базується на попередньому кроці, дозволяючи враховувати контекст і залежності від попередніх елементів послідовності.

Проте рекурентні нейронні мережі мають свої обмеження. Одним з найбільш відомих є проблема зниклої градієнта, коли важко навчити модель робити довгострокові залежності через деякі обмеження в алгоритмах зворотного поширення помилки.

Завдяки цим обмеженням було розроблено більш складні архітектури, такі як Long Short-Term Memory LSTM і Gated Recurrent Units GRU, які більш ефективно управляють довгостроковими залежностями. LSTM і GRU використовують механізми воріт для контролю потоку інформації в рекурентному шарі, дозволяючи ефективніше керувати інформацією в прихованому стані через час [31–34].

Рекурентна нейронна мережа представляє собою тип штучної нейронної мережі, де зв'язки між вузлами створюються у вигляді спрямованого або неспрямованого графа в часі. Це дає змогу мережі динамічно взаємодіяти з інформацією у відповідь на послідовність поданих даних.

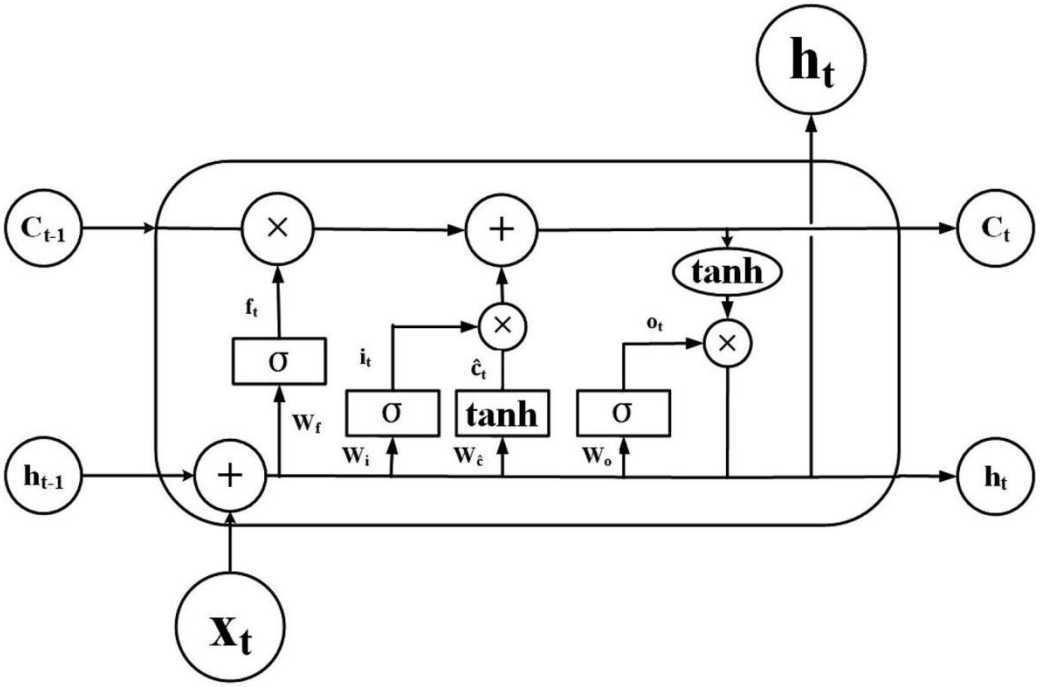


Рисунок 1.2 – Структурна схема LSTM [32]

У випадку задачі узагальнення тексту модель будується за принципом багато до багатьох, схожим на задачу перекладу. Тут довжина вхідної інформації може відрізнятися від довжини вихідної. Після отримання всієї вхідної інформації модель починає генерувати вихідні дані за допомогою нейронної мережі.

Відмінність між узагальненням тексту і перекладом полягає в тому, що у випадку узагальнення цільовою міткою є коротке узагальнення інформації, а не перекладене речення, і також може відрізнятися за довжиною.

При роботі з завданнями обробки природної мови прості рекурентні нейронні мережі мають деякі недоліки, включаючи складність доступу до давніх інформаційних станів та неможливість урахування майбутніх вхідних даних для поточного стану. У простій РНМ слова обробляються послідовно, що може призводити до проблеми зникнення або вибуху градієнту, особливо при обробці довгих речень у архітектурі багато до багатьох, де останні слова можуть мати надмірний вплив на результат.

Одним зі способів поєднання переваг цих підходів стала модель "від послідовності до послідовності", яка забезпечує зручність обробки послідовностей із застосуванням вищезгаданих інновацій. Цей еволюційний шлях нарешті призвів до трансформаторних моделей і seq2seq архітектур, які мають велике значення в сучасній НЛП.

Хоча прості рекурентні нейронні мережі мають пам'ять, вони мають проблему зникаючого градієнту, особливо при роботі з довгими залежностями. Long Short-Term Memory є архітектурою рекурентної нейронної мережі, спроектованою для роботи з такими довготривалими залежностями. У LSTM є блоки пам'яті, що дозволяють запам'ятовувати і передавати інформацію вздовж послідовності без втрати суттєвих деталей.

Основною ідеєю LSTM є використання трьох типів вентилів: вентиль забування – forget gate, вхідний вентиль – input gate і вихідний вентиль – output gate. Кожен крок вхідних даних і попередньої станції проходить через ці вентилі. Спочатку поточні дані вхідного x_t разом зі станом h_{t-1} передаються через вентиль

забування з сигмоїдальною активацією для визначення, що потрібно забути. Далі вони проходять через вхідний вентиль, де вони опрацьовуються з сигмоїдальною та гіперболічною тангенс активацією, і їх добуток використовується для визначення нового стану пам'яті c_t. У контексті моделей послідовностей, прості рекурентні нейронні мережі ефективно опрацьовують послідовності даних і роблять прогнози, але мають обмежену короткочасну пам'ять. Для розв'язання цієї проблеми були розроблені вентильні механізми в LSTM, які дозволяють регулювати потік інформації в мережі. LSTM широко використовуються в сучасних додатках глибокого навчання, таких як розпізнавання мови, синтез мови та розуміння природної мови [35–37].

Довготривала короткочасна пам'ять є покращенням над простими рекурентними нейронними мережами, яке вирішує проблему зникнення градієнта та дозволяє зберігати довгострокові залежності в послідовностях даних. Основною ідеєю LSTM є використання спеціальних вентилів для керування потоком інформації в кожному кроці мережі.

У LSTM є три основних типи керування. Розглянемо їх далі.

1. Забування. Цей тип керування визначає, яка інформація повинна бути забута перед використанням нових даних. Він використовує сигмоїдну функцію активації, що генерує значення між 0 і 1 для кожного елемента в попередньому стані.

2. Вхід – визначає, які нові дані мають бути збережені в пам'яті. Він також використовує сигмоїдну функцію активації для створення вектору, який вказує, які значення мають бути оновлені.

3. Вихід – вирішує, яка інформація повинна вийти з LSTM наступного кроку. Він поєднує вхідні дані з попереднім станом та визначає, які значення виходять з комірки.

Комбінація цих вентилів дозволяє LSTM зберігати та використовувати довгострокові залежності в послідовностях даних, що робить їх ефективними для різних завдань обробки природної мови, таких як розпізнавання мови, синтез мови та машинний переклад.

Модель “від послідовності до послідовності” призначена для перетворення однієї послідовності у іншу, таку як переклад тексту або генерація відповіді на запит. Ця модель складається з двох частин: кодер і декодер, які працюють зі змінною довжиною вхідних та вихідних даних.

Трансформер нова архітектура моделі, що не використовує рекурентні частини, такі як GRU або LSTM, а замість цього базується виключно на механізмі уваги. У цій моделі вхідні та вихідні дані з маскуванням слів вбудовуються через кодер, а потім обробляються багатоголовою увагою для встановлення зв'язків між ними і виводу ймовірностей слів для вгадування замаскованих слів.

Принципово, механізм уваги досліджує вхідну послідовність і визначає, які її частини є релевантними на кожному етапі. Це подібно до того, як читаєте книгу і зосереджуєтеся на поточному слові, але мозок також запам'ятовує ключові слова тексту для забезпечення контексту.

Для певної послідовності механізм уваги працює аналогічно людському перекладу. Наприклад, при перекладі речення кодер не лише створює переклад, але також передає декодеру ключові слова, які важливі для семантики речення. Ці ключові слова допомагають декодеру краще розуміти контекст і полегшують процес перекладу.

Трансформаторні моделі виявилися ефективними в задачах послідовного перетворення, перевершуючи традиційні кодерно-декодерні підходи. Їх успіх у вузьких і загальних задачах демонструє високий потенціал цих архітектур у сучасних областях машинного навчання.

BART, розроблений командою вчених з Google Language, є моделлю трансформерів, яка використовує тільки кодер і двонаправлене навчання. Його ключовий технологічний внесок полягає в тому, що він може аналізувати всю послідовність слів одразу, у порівнянні з попередніми моделями, які розглядали текст лише в одному напрямку [38–40]. Ця здатність дозволяє BART краще розуміти контекст і значення слів. Крім того, дослідники впровадили новий

метод під назвою "масковане мовне моделювання" що дозволяє здійснювати ефективне двонаправлене навчання в трансформерах.

Метод "масковане мовне моделювання", використовуваний у BART, включає в себе особливий підхід до навчання моделі. Під час навчання BART деякі зі словів вхідного тексту випадково "маскуються", тобто замінюються спеціальним токеном [MASK]. Модель тоді намагається передбачити оригінальне слово на основі контексту і інших слів у реченні. Цей підхід дозволяє BART вчитися здійснювати двонаправлене розуміння контексту, оскільки модель повинна засвоїти інформацію з обох сторін маскованого слова для успішного прогнозування.

При двонаправленому навчанні BART аналізує контекст слова, враховуючи як ліву, так і праву частини речення. Це дозволяє моделі краще усвідомлювати семантичні зв'язки між словами і фразами в тексті, що призводить до покращення якості розуміння мови та узагальнюючих здібностей моделі.

BART відзначається високими результатами на різних завданнях обробки природної мови, включаючи завдання класифікації, питання-відповідь, іменоване утворення тощо. Його успіх демонструє переваги двонаправленого підходу до навчання мовних моделей і підтверджує потужний потенціал трансформерів у розв'язанні складних завдань машинного навчання, пов'язаних з текстом.

Точне налаштування BART на конкретних завданнях після попереднього навчання, а також використання вивчених кодерами вбудовувань слів виявилися дуже успішними. Вбудовані представлення слів можуть бути використані як ознаки в інших моделях, і вони відрізняються від підходу word2vec тим, що вони контекстно-залежні і змінюються залежно від контексту, у якому використовуються слова.

ProphetNet є моделлю попереднього навчання від послідовності до послідовності, яка використовує нові методи самоконтролю, що включають прогнозування майбутніх n-грам та багатопотоковий самоконтроль. Замість

прогнозування однієї наступної лексеми на кожному кроці, як у багатьох інших моделях seq2seq, ProphetNet використовує минулий контекст для прогнозування наступних n-крокових лексем. Ці прогнози використовуються для подальшої підготовки і передаються у декодер моделі.

ProphetNet відзначається відмінними результатами порівняно з іншими моделями попереднього навчання на широкому спектрі даних. В архітектурі ProphetNet є деякі відмінності від класичної трансформаторної архітектури. Кодер виглядає схожим на трансформатор, але декодер має основні входи, що доповнюються входами від N-го потоку прогнозування. Ця архітектура підвищує здатність генерації тексту і призводить до високих результатів у завданнях обробки природної мови.

1.3 Мета та постановка задачі

Провівши аналіз предметної області та методів вирішення поставленої задачі було сформульовану мету кваліфікаційної роботи.

Мета кваліфікаційної роботи бакалавра полягає в покращенні генерації короткого змісту тексту для формування заголовків, назв, коротких виразних фраз.

Задачі роботи :

– провести аналіз відомих методів для автоматичної генерації заголовків, назв та коротких виразних фраз на основі аналізу текстових даних;

– розробити метод, здатного навчатися на великому наборі текстових даних для генерації короткого змісту з урахуванням контексту та семантики тексту;

– здійснити оптимізацію параметрів нейронної мережі для покращення її якості та точності в завданнях генерації короткого змісту тексту;

– провести експериментальне тестування розробленого методу на наборі текстових даних для оцінки його ефективності та точності .

Розділ 2 Метод генерації короткого змісту тексту з використанням нейронної мережі

2.1 Процес генерації короткого змісту тексту

Процес генерації короткого змісту тексту для формування заголовків, назв або коротких виразних фраз може включати в себе використання різних методів і підходів залежно від конкретної задачі та вимог. Розглянемо детальніше кожен з етапів цього процесу.

1. Розуміння тексту. Цей етап передбачає аналіз тексту з метою виділення основних ідей, ключових фактів або деталей. Часто використовуються методи обробки природної мови для виявлення тематики тексту, ідентифікації ключових слів або фраз, аналізу синтаксичної структури та визначення ступеня важливості різних частин тексту.

2. Виділення ключових інформаційних елементів. На цьому етапі інформація з тексту витягується та перетворюється у короткі, лаконічні фрази або речення. Вибираються найбільш суттєві аспекти або факти, які слід включити у короткий зміст.

3. Генерація короткого змісту. Використовуючи зібрані ключові елементи, створюється короткий текстовий вираз. Це може бути здійснено за допомогою генерації тексту засобами штучного інтелекту або застосуванням правилових алгоритмів для складання смислових речень.

4. Оцінка та вибір найкращого варіанту. Згенерований короткий зміст оцінюється з точки зору його інформативності, ясності та відповідності основному змісту тексту. Вибирається оптимальний варіант, який найкраще передає суть тексту в компактній формі.

Цей процес може застосовуватися до різноманітних типів текстів, включаючи новини, статті, огляди продуктів або послуг, наукові статті тощо. Важливою перевагою використання сумаризації тексту є здатність швидко та ефективно витягати найважливішу інформацію з джерела та передавати її у

вигляді лаконічного змісту, що допомагає привернути увагу аудиторії та зекономити час читачів.

Процес починається з розуміння тексту. Під час цього етапу аналізується вміст тексту з метою виявлення основних ідей, ключових фактів або деталей. Зазвичай для цього використовуються методи обробки природної мови, щоб визначити тематику тексту, ідентифікувати ключові слова або фрази, провести аналіз синтаксичної структури та встановити ступінь важливості різних частин тексту.

Після того як текст був розібраний, відбувається етап виділення ключових інформаційних елементів. На цьому етапі інформація з тексту екстрагується та перетворюється у короткі, лаконічні фрази або речення. Вибираються найбільш суттєві аспекти або факти, які слід включити у короткий зміст.

Після отримання ключових елементів переходимо до генерації короткого змісту. За допомогою зібраних ключових елементів створюється короткий текстовий вираз. Це може бути здійснено за допомогою генерації тексту штучним інтелектом або застосуванням правилових алгоритмів для складання смислових речень.

Останнім етапом є оцінка та вибір найкращого варіанту короткого змісту. Згенерований короткий зміст оцінюється з точки зору його інформативності, ясності та відповідності основному змісту тексту. В результаті вибирається оптимальний варіант, який найкраще передає суть тексту у компактній формі.

Процес сумаризації тексту, це аналіз тексту з метою визначення його основних ідей та ключових деталей, що дозволяє створити стислий короткий зміст. Під час цього процесу виділяються основні концепції та інформація, які потім перетворюються у лаконічні вирази або речення.

Головною метою сумаризації тексту є зробити інформацію більш доступною та зрозумілою, особливо в умовах великої кількості даних. Цей процес може використовуватися для створення заголовків, назв, або просто

коротких описів текстів, що дозволяє швидко здійснювати перегляд і вибірку важливої інформації.

При використанні сумаризації тексту застосовуються різні техніки, включаючи аналіз синтаксису, визначення ключових слів та важливих фраз, а також генерацію нових текстових виразів на основі зібраної інформації. Цей процес дозволяє ефективно витягати суттєву інформацію з тексту та представляти її у компактній формі, що є особливо корисним в умовах великих обсягів даних.

Далі розглянемо основні кроки процесу сумаризації тексту.

1. Розуміння вихідного тексту. Перший крок - це ретельне ознайомлення з вихідним текстом для зрозуміння основних ідей, ключових деталей та структури тексту.

2. Виділення ключових ідей. Ідентифікація ключових ідей, які необхідно включити у короткий зміст. Це може включати основні теми, важливі факти або особливі деталі.

3. Вибір важливих речень. Відбір речень, які найкраще відображають ключові ідеї тексту. Ці речення повинні бути ясними, конкретними і інформативними.

4. Генерація короткого виразу. На основі вибраних ключових ідей та речень генерується короткий вираз або заголовок, що передає основне значення тексту.

5. Перевірка і уточнення. Перевірка отриманого короткого змісту на відповідність оригінальному тексту та уточнення деталей для точності і повноти.

6. Оцінка якості. Оцінка якості сумаризації з використанням метрик, таких як ROUGE, яка порівнює згенерований короткий зміст з оригінальним текстом.

Процес створення короткого змісту тексту розпочинається з ретельного ознайомлення з вихідним текстом. Це дозволяє отримати розуміння основних ідей, ключових деталей та структури тексту.

Наступним кроком є виділення ключових ідей з тексту. Під час цього етапу ідентифікуються основні теми, важливі факти або особливі деталі, які слід включити у короткий зміст.

Після виділення ключових ідей відбувається вибір важливих речень. Обираються ті речення, які найкраще відображають ключові ідеї тексту і є ясними, конкретними та інформативними. Зібрані ключові ідеї та речення використовуються для генерації короткого виразу або заголовка, що передає основне значення тексту. Після генерації короткого виразу виконується перевірка і уточнення. Отриманий короткий зміст перевіряється на відповідність оригінальному тексту, а також уточнюються деталі для точності і повноти відображення основного змісту.

Завершальним етапом є оцінка якості сумаризації. Використовуються метрики, такі як ROUGE, для порівняння згенерованого короткого змісту з оригінальним текстом і оцінки якості передачі інформації у компактній формі.

Розуміння вихідного тексту є ключовим етапом у процесі сумаризації тексту. Цей етап включає наступні дії для детального ознайомлення з текстом.

1. Ознайомлення з вихідним текстом:

– читання тексту з увагою для отримання повного уявлення про його зміст;

– спроба зрозуміти загальну тему або центральну ідею тексту.

2. Аналіз основних ідей і структури:

– визначення основних ідей, що розглядаються у тексті;

– розбір структури тексту, включаючи вступ, основну частину і висновки.

3. Виокремлення ключових аспектів інформації:

– визначення найважливіших понять, фактів або подій, які становлять основу тексту;

– виділення ключових деталей, які необхідно включити у коротку сумаризацію.

4. Зазначення основного повідомлення:

– визначення головного повідомлення або основної ідеї, яку автор намагається передати;

– розуміння підтексту або основного мотиву за текстом.

Рисунок 2.1 – Кроки процесу сумаризації тексту

Ці кроки відображають загальний процес створення короткого, змістовного огляду вихідного тексту. Використовуючи цей підхід, можна швидко здійснювати узагальнення тексту для різних цілей, від підготовки заголовків новин до створення конспектів наукових досліджень.

Кроки допомагають створити чітке уявлення про вихідний текст і виділити найважливіші аспекти для подальшої сумаризації. Розуміння тексту допомагає ефективно стиснути і передати основну інформацію у короткому вигляді. Цей етап включає уважне вивчення вихідного тексту з метою отримання повного розуміння його змісту. Під час читання тексту детально досліджується загальна тема або головна ідея, яку автор намагається передати.

Далі проводиться аналіз основних ідей, які висвітлюються в тексті, а також розглядається його структура, включаючи вступ, основну частину і висновки. Під час аналізу виокремлюються найважливіші поняття, факти або події, які становлять основу тексту, а також ключові деталі, які необхідно врахувати під час підготовки короткого викладу. Нарешті, розуміння основного повідомлення або основної ідеї тексту, а також виявлення підтексту або основного мотиву, які автор намагається передати через текст.

Після ретельного ознайомлення з вихідним текстом і визначення основних ідей та структури, наступним етапом є виділення ключових аспектів інформації. Це включає ідентифікацію найважливіших понять, фактів або подій, які утворюють основу тексту. Під час цього аналізу також виділяються ключові деталі, які має бути включено у коротку сумаризацію тексту.

Після виділення ключових аспектів інформації приймається рішення щодо основного повідомлення тексту. Це означає визначення головної ідеї чи повідомлення, яке автор намагається передати своїм читачам через текст. Під час цього аналізу також важливо розуміти підтекст або основний мотив, який лежить в основі тексту і впливає на спосіб, яким він сприймається.

Враховуючи усі ці аспекти, можна перейти до складання короткого виразу або заголовка, який чітко і конкретно передає основний зміст і ідеї тексту. Після цього важливо перевірити отриманий короткий зміст на

відповідність оригінальному тексту та уточнити деталі для забезпечення точності і повноти передачі інформації.

Завершальним етапом буде оцінка якості сумаризації. Використання метрик, таких як ROUGE, дозволить об'єктивно оцінити те, наскільки добре короткий зміст відображає основний зміст та ідеї вихідного тексту. Оцінка якості допоможе виявити можливі покращення і забезпечити більш ефективну передачу інформації через короткі форми тексту.

Виділення ключових ідей у тексті для подальшої сумаризації передбачає наступні дії:

1. Ідентифікація головних тем або ідей:

– чітке визначення основної теми або центральної проблеми, яку порушує автор;

– розуміння основного змісту тексту і виявлення ключових тематичних аспектів.

2. Відбір найважливіших фактів або деталей:

– визначення найсуттєвіших фактів, подій або інформаційних деталей, які відображають основну суть тексту;

– вибір ключових деталей, які мають найбільше значення для передачі основної інформації без додаткових витоків.

3. Аналіз контексту:

– врахування контекстуальних відношень між різними ідеями або фактами у тексті;

– розуміння взаємозв'язків між різними частинами тексту для виокремлення головних ідей.

4. Врахування мети тексту:

– врахування цілей автора і основного повідомлення тексту при виборі ключових ідей;

– аналіз того, яку інформацію автор прагне передати читачеві.

В цих діях ідентифікації і вибору ключових ідей важливо зосередитися на суттєвих аспектах тексту, які найбільше впливають на загальне розуміння

інформації. Це допомагає створити ефективний короткий зміст, який передає головну суть тексту без додаткових деталей.

Вибір важливих речень для сумаризації тексту включає наступні етапи:

1. Ідентифікація ключових ідей:

– аналіз тексту для визначення основних тематичних аспектів;

– розуміння центральної проблеми або основного повідомлення тексту.

2. Пошук конкретних речень:

– відбір речень, які найкраще відображають ключові ідеї або підтримують основну тему тексту;

– врахування речень, що містять важливі факти, докази або інформаційні деталі.

3. Оцінка яскравості та інформативності:

– вибір речень, які є чіткими, зрозумілими та інформативними;

– врахування того, наскільки кожне речення передає основну суть без зайвих деталей.

4. Упорядкування речень:

– оптимізація порядку речень для підвищення чіткості та послідовності в сумаризації;

– розгляд речень у контексті їх відношення до інших частин тексту.

5. Відсів речень:

– виключення речень, що не відображають ключові ідеї або не додають інформативності до сумаризації;

– врахування обсягу сумаризації та обмеження на кількість речень.

Етапи допомагають ефективно вибрати важливі речення для створення змістовної та інформативної сумаризації тексту. Під час цього процесу важливо зосередитися на якості, чіткості та послідовності вибраних речень, щоб забезпечити ефективне передавання основної інформації тексту у скороченій формі.

Генерація короткого виразу або заголовку заснована на ключових ідеях і вибраних реченнях з оригінального тексту. Основними кроками цього процесу є:

1. Складання короткого вислову:

– засноване на вибраних ключових ідеях створення короткого вислову або заголовку, що передає основну суть тексту;

– використання основних тематичних аспектів для сформулювання концентрованого вислову.

2. Чіткість та конкретність:

– забезпечення чіткості та конкретності в сформульованому вислові;

– уникнення загальних або неясних висловів, фокусуючись на основній інформації.

3. Передача основної суті:

– використання сформульованого виразу для передачі основного змісту інформації з оригінального тексту;

– забезпечення того, що короткий вираз або заголовок відображає ключові аспекти та ідеї тексту.

4. Оцінка вислову:

– перевірка сформульованого вислову на відповідність ключовим ідеям та основній суті тексту.

– проведення корекцій та уточнень для досягнення точного виразу.

Генерація короткого виразу або заголовку після виділення ключових ідей та речень допомагає створити компактне та інформативне висловлення, яке передає головну інформацію з оригінального тексту в зрозумілій формі. Важливо дотримуватися чіткості, конкретності та відповідності основним темам тексту під час цього процесу.

Після створення короткого змісту з використанням ключових ідей та вибраних речень з оригінального тексту, важливо провести перевірку і уточнення для забезпечення відповідності оригінальному матеріалу. Основні кроки цього процесу.

1. Перевірка відповідності:

– перевірка створеного короткого змісту з оригінальним текстом для визначення відповідності ключовим ідеям і основним темам;

– порівняння сформульованого змісту з оригінальним матеріалом для виявлення можливих розбіжностей чи відсутності важливих деталей.

2. Уточнення інформації:

– уточнення деталей інформації, яка може бути недостатньою або неповною у короткому змісті;

– додавання додаткових деталей або уточнень для забезпечення точності та повноти сумаризації.

3. Адаптація до контексту:

– врахування контексту оригінального тексту під час перевірки і уточнення;

– забезпечення того, що короткий зміст відображає вірний контекст та інтерпретацію оригінальної інформації.

4. Оцінка точності:

– оцінка точності короткого змісту з огляду на відповідність основним ідеям та деталям оригінального тексту;

– виправлення недоліків або неточностей у короткому змісті для досягнення кращої відповідності оригіналу.

Перевірка і уточнення короткого змісту дозволяє забезпечити високу якість сумаризації, відповідність ключовим аспектам оригінального тексту та передачу головної інформації у короткій і зрозумілій формі. Важливо активно взаємодіяти з оригінальним матеріалом під час цього процесу для досягнення точності та повноти у короткому змісті.

Оцінка якості сумаризації тексту є важливим етапом у процесі розробки і використання автоматичних методів узагальнення. Для оцінки якості короткого змісту використовуються спеціальні метрики, такі як ROUGE. Основні кроки оцінки якості сумаризації включають використання метрик ROUGE. Це набір метрик, які оцінюють якість сумаризації, порівнюючи сформовані короткі змістові одиниці з оригінальним текстом. ROUGE враховує взаємодію між розбіжностями між сформованими короткими змістовими одиницями та текстом-джерелом, визначаючи подібність та точність.

Порівняння з оригінальним текстом. Сформований короткий зміст порівнюється з оригінальним текстом для визначення ступеня відповідності та відтворення ключових ідей та інформації. Оцінка ефективності сумаризації полягає у зіставленні створеного короткого змісту з ключовими аспектами оригінального тексту. Визначення метрик якості. Оцінка метриками ROUGE дозволяє визначити точність, повноту та інші параметри ефективності сумаризації. Використання ROUGE допомагає кількісно оцінити якість сумаризації і порівняти різні підходи або моделі. Оцінка якості сумаризації тексту забезпечує об'єктивний підхід до вимірювання ефективності автоматичних методів узагальнення. Вона дозволяє визначити переваги та недоліки підходів та моделей і вдосконалити процес сумаризації для кращої передачі ключової інформації в короткій формі.

2.3 Метод генерації короткого змісту тексту

Процес створення заголовків за допомогою моделі включає кілька етапів. Спочатку текст новинної статті готується для подальшої обробки. До тексту додається певний ідентифікатор "Заголовок:", що допомагає моделі зрозуміти, що її метою є генерація заголовка на основі цього тексту. Потім вхідний текст токенізується за допомогою відповідного токенізатора, що перетворює його на послідовність числових ідентифікаторів. Після цього текст конвертується в вектори ідентифікаторів і маски уваги, які стають вхідними даними для моделі. Якщо є доступ до GPU, дані переносяться на цей пристрій для прискорення обчислень. Потім модель застосовує метод генерації з пошуком променя для створення різних варіантів заголовків на основі вхідного тексту. Після генерації модель повертає послідовність числових ідентифікаторів, яку потрібно декодувати, щоб отримати сформований заголовок.

Метод генерації заголовків за допомогою включає кілька кроків.

1. Вхідна інформація. Першим кроком у процесі генерації заголовків є отримання вхідної інформації, якою є текст новинної статті. Цей текст містить

основний зміст, який необхідно узагальнити у вигляді заголовка. Важливо, щоб текст статті був повним та точним, оскільки від його якості залежить точність та релевантність згенерованого заголовка. Текст може включати різні елементи, такі як заголовок, підзаголовки, абзаци та цитати, що разом складають повне новинне повідомлення.

2. Підготовка вхідних даних. Спочатку, текст новинної статті підготовлюється для подальшої обробки. На цьому етапі здійснюється підготовка тексту новинної статті для подальшої обробки моделлю. Цей процес включає кілька важливих кроків:

До тексту новинної статті додається спеціальний ідентифікатор завдання, наприклад, "Заголовок:". Цей ідентифікатор вказує моделі на те, що її завданням є генерація заголовка на основі наданого тексту. Додавання такого ідентифікатора допомагає моделі чіткіше розуміти контекст і цільову задачу.

Текст статті може потребувати попередньої обробки, яка включає очищення від зайвих символів, виправлення помилок та стандартизацію формату. Це забезпечує більш точне і ефективне опрацювання тексту моделлю. Обробка тексту може включати такі аспекти. Якщо текст отриманий з веб-джерела, необхідно видалити HTML-теги та інші елементи форматування. Видалення зайвих символів, таких як спеціальні символи або знаки пунктуації, які не несуть смислового навантаження. Перетворення тексту в нижній регістр, стандартизація скорочень та інших елементів для забезпечення однорідності. Текст може бути розбитий на окремі абзаци або речення для полегшення аналізу. Це дозволяє моделі краще розуміти структуру тексту і виділяти ключові ідеї.

Якщо це необхідно, до тексту можуть бути додані додаткові контекстні дані, такі як дата публікації, автор, категорія новини тощо. Це може допомогти моделі точніше інтерпретувати зміст і контекст статті.

Підготовлений текст форматується у вигляді, зручному для подальшої токенізації. Це може включати об'єднання всіх компонентів тексту у єдину строку з відповідними роздільниками. Підготовлений текст на цьому етапі є

готовим для токенізації та подальшої обробки моделлю, що дозволяє ефективно здійснювати генерацію заголовків на основі новинних статей.

3. Токенізація. Вхідний текст статті токенізується за допомогою відповідного токенізатора, який використовується для моделі. Токенізація перетворює текст на послідовність числових ідентифікаторів токенів, які може обробити модель.

Токенізація є важливим кроком у процесі підготовки тексту до обробки нейронною мережею. Вона полягає у перетворенні вхідного тексту на послідовність числових ідентифікаторів, які може обробляти модель. Ось детальніший опис цього процесу:

Токенізатор вибирається відповідно до моделі, яка буде використовуватись. Різні моделі мають свої власні токенізатори, налаштовані для оптимальної роботи з текстовими даними. Наприклад, для моделей на основі трансформерів, таких як BERT, ProphetNet, використовуються спеціально розроблені токенізатори.

Текст новинної статті розбивається на окремі елементи, які називаються токенами. Це можуть бути слова, частини слів або навіть окремі символи. Спочатку текст розбивається на окремі слова. Слова розбиваються на менші частини, такі як префікси або суфікси. Це корисно для обробки незнайомих слів або морфологічних варіантів. Кожен символ тексту стає окремим токеном. Це підходить для роботи з мовами з великою кількістю ієрогліфів або спеціальних символів. Кожен токен зіставляється з унікальним числовим ідентифікатором на основі словника, який використовується токенізатором. Цей словник створюється під час тренування моделі і містить всі можливі токени та їх числові відповідники.

До токенізованої послідовності додаються спеціальні токени, які сигналізують початок і кінець тексту наприклад, `[CLS]` для початку і `[SEP]` для розділення або кінця тексту в моделях. Ці токени допомагають моделі зрозуміти структуру вхідних даних.

В результаті токенізації утворюється послідовність числових ідентифікаторів, яка представляє вхідний текст. Ця послідовність може бути піддана додатковій обробці, такій як обрізання до фіксованої довжини або заповнення нулями для досягнення необхідної довжини . Поряд з токенізованою послідовністю створюється маска уваги, яка вказує моделі, які токени є значущими 1 і які слід ігнорувати 0. Це особливо корисно при обробці текстів різної довжини.

Токенізація перетворює текст новинної статті у форму, яку може ефективно обробляти модель на основі трансформерів, забезпечуючи таким чином можливість подальшої генерації заголовка.

4. Кодування тексту. Після токенізації тексту статті конвертується у вектори ідентифікаторів та маски уваги. Ці вектори ідентифікаторів представляють собою вхідні дані для моделі. Передача на GPU. Якщо доступна GPU, дані переносяться на цей пристрій для прискорення обробки моделлю.

Після токенізації вхідного тексту новинної статті, наступним кроком є кодування тексту у формат, який може бути використаний моделлю для генерації заголовка. Цей процес включає кілька важливих етапів:

Токенізовані числові ідентифікатори перетворюються у вектори, які представляють собою багатовимірні числові представлення токенів. Ці вектори зберігають інформацію про семантичне значення кожного токена і використовуються моделлю для розуміння контексту.

Маска уваги формується паралельно з векторами ідентифікаторів. Вона вказує моделі, які токени слід враховувати під час обробки, а які ігнорувати. Це особливо важливо для текстів різної довжини, де незначущі токени зазвичай додані для досягнення необхідної довжини послідовності мають бути проігноровані. Маска уваги допомагає моделі фокусуватися на релевантних частинах тексту.

Якщо для обробки використовується графічний процесор (GPU), дані переносяться на цей пристрій. GPU значно прискорює процес обробки великих обсягів даних завдяки своїй здатності паралельно обробляти велику кількість

обчислень. Перенесення даних включає як вектори ідентифікаторів, так і маски уваги. До початку і кінця векторів ідентифікаторів додаються спеціальні токени, такі як `[CLS]` для позначення початку і `[SEP]` для розділення або кінця тексту. Ці токени допомагають моделі зрозуміти структуру вхідних даних і контекст завдання.

Для ефективної обробки дані об'єднуються у пакети. Це дозволяє моделі одночасно обробляти декілька прикладів, що підвищує ефективність обчислень і використання ресурсів. Кожен пакет містить вектори ідентифікаторів і маски уваги для кількох текстів.

Закодовані дані подаються на вхід моделі. Модель використовує ці вектори і маски уваги для аналізу вхідного тексту, розуміння його змісту та контексту, що є необхідним для подальшої генерації заголовка. Цей етап забезпечує підготовку тексту у формат, який оптимально підходить для обробки моделлю на основі трансформерів, забезпечуючи таким чином точне і ефективне виконання завдання генерації заголовка.

5. Генерація заголовка. Використовуючи метод генерації з пошуком променя, модель генерує можливі варіанти заголовків на основі переданого вхідного тексту. Параметри генерації, такі як кількість променів, максимальна та мінімальна довжина, дозволяють керувати процесом генерації.

На цьому етапі модель використовує закодовані вхідні дані для генерації заголовка. Процес включає кілька важливих кроків, які забезпечують створення якісного і релевантного заголовка:

Модель отримує вектори ідентифікаторів і маски уваги, які були підготовлені на попередніх етапах. Процес генерації починається з ініціалізації спеціальних токенів, що сигналізують початок тексту, наприклад, токена `[CLS]`.

Для генерації заголовка застосовується метод пошуку променя, який є ефективним алгоритмом для створення текстових послідовностей. Beam Search розглядає кілька можливих варіантів на кожному кроці генерації і вибирає найімовірніші з них для подальшої обробки. Цей метод дозволяє моделі

уникнути вибору найімовірнішого, але можливо менш якісного заголовка, і натомість розглянути кілька варіантів, щоб знайти найкращий.

Параметри генерації, такі як кількість променів, максимальна та мінімальна довжина заголовка, встановлюються перед початком процесу. Ці параметри дозволяють контролювати процес генерації, обмежуючи довжину заголовка і забезпечуючи, що заголовок буде достатньо інформативним, але не надто довгим.

Процес генерації заголовка є ітеративним. На кожному кроці модель прогнозує наступний токен, використовуючи вже згенеровані токени і контекст вхідного тексту. Найімовірніші токени додаються до поточної послідовності, і цей процес повторюється до досягнення заданої максимальної довжини або до генерації токена, що сигналізує кінець заголовка.

Під час генерації модель може створювати кілька варіантів заголовків завдяки методу пошуку променя. Ці варіанти оцінюються на основі ймовірності та релевантності, і найкращий з них вибирається як кінцевий результат. Згенеровані заголовки оцінюються за кількома критеріями, такими як чіткість, зрозумілість, релевантність і граматична правильність. Цей етап може включати автоматичну перевірку або додаткову оцінку людиною для забезпечення високої якості заголовка.

Процес генерації заголовка завершується, коли модель створює короткий, інформативний і точний заголовок, який відповідає змісту вхідного тексту новинної статті. Згенерований заголовок готовий для декодування та подальшого використання.

6. Декодування результату. Після генерації заголовків модель повертає послідовність числових ідентифікаторів, яку необхідно декодувати за допомогою токенізатора. Результат декодування відображає згенерований заголовок, який може бути використаний для подальших застосувань.

Після того як модель завершила процес генерації заголовка, отриманий результат потребує декодування для перетворення його з числових ідентифікаторів у зрозумілий текст. Цей процес включає кілька важливих етапів:

Модель повертає послідовність числових ідентифікаторів, які відповідають згенерованим токенам. Ця послідовність представляє потенційний заголовок у вигляді чисел, кожне з яких відповідає певному токену з словника токенізатора.

Токенізатор, який використовувався для токенізації вхідного тексту, тепер використовується для зворотного перетворення числових ідентифікаторів у текстові токени. Це перетворення відбувається за допомогою словника токенізатора, який зіставляє кожен числовий ідентифікатор з відповідним йому токеном.

Декодовані текстові токени об'єднуються у суцільний текстовий рядок. Під час цього процесу видаляються спеціальні токени, такі як `[CLS]`, `[SEP]` та інші, які використовувалися для форматування вхідних даних. Результатом є суцільний текст, що представляє згенерований заголовок.

Отриманий текст може потребувати додаткової обробки для виправлення можливих помилок і покращення читабельності. Це включає перевірку і виправлення пунктуаційних знаків для забезпечення граматичної правильності. Перевірка тексту на наявність орфографічних та граматичних помилок і їх виправлення. Упорядкування пробілів між словами для забезпечення чистого і акуратного вигляду тексту.

Декодований текст перевіряється на відповідність вимогам до заголовка. Заголовок повинен мати зрозумілу структуру і передавати основну ідею новинної статті. Відповідати змісту новинної статті і коректно відображати її основну тему. Бути достатньо коротким для заголовка, але водночас інформативним.

Після завершення всіх перевірок і корекцій згенерований заголовок готовий до використання. Його можна використовувати для публікації на новинних вебсайтах, у соціальних мережах або інших платформах, де необхідно коротко представити новину.

Таким чином, процес декодування результату забезпечує перетворення числових ідентифікаторів у зрозумілий і якісний текстовий заголовок, який повністю готовий для подальшого застосування.

7. Вихідна інформація. Результат сумаризації – згенерований заголовок. Завершальним етапом процесу є отримання та представлення вихідної інформації, тобто згенерованого заголовка. Після декодування результату модель повертає текстовий заголовок, який можна безпосередньо використовувати. Цей заголовок повинен бути коротким, точним і відображати основну суть новинної статті.

Згенерований заголовок можна перевірити на відповідність критеріям. Заголовок має бути зрозумілим для читача і чітко передавати основну ідею статті. Заголовок повинен відповідати змісту статті і точно відображати її основну тему. Заголовок має бути граматично правильним і логічно побудованим.

Цей заголовок потім можна використовувати для публікації на новинних вебсайтах, у соціальних мережах або інших платформах, де необхідно коротко представити новину. Таким чином, процес генерації заголовків завершується створенням тексту, готового до використання у подальших застосуваннях.

Основні складові цієї моделі включають різні шари трансформерів і механізми уваги, які дозволяють моделі ефективно розуміти та генерувати текст.

Основні елементи структури моделі включають.

1. Вступні шари:

– токенізація конвертує вхідний текст на слова у векторну форму числових ідентифікаторів токенів;

– позиційне кодування додає інформацію про позиції слів у векторному представленні для збереження послідовності.

2. Кодувально-декодувальні шари:

– енкодер складається з декількох блоків трансформерів, які обробляють вхідний текст і генерують контекстний вектор;

Рисунок 2.2. – Схема методу генерації заголовків

– декодер також містить кілька блоків трансформерів, але використовується для генерації вихідного тексту з урахуванням контекстного вектора.

Процес генерації заголовків залежить від якості навчання моделі на відповідному наборі даних та вимог конкретного завдання. При використанні моделі важливо розуміти її обмеження та правильно налаштовувати параметри генерації для досягнення оптимальних результатів.

Структура моделі базується на архітектурі ProphetNet, що поєднує методи згорткових нейронних мереж і трансформерів для завдань генерації тексту.

3. Механізм уваги:

– багатоблокова увага дозволяє моделі фокусуватися на різних частинах вхідного тексту одночасно, що поліпшує якість генерації тексту.

– самоувага дозволяє моделі визначати взаємозв'язки між словами у тексті.

4. Структура згорткових нейронних мереж:

– згорткові шари використовуються для обробки тексту з різних точок зору і витягують корисні ознаки.

5. Зворотне поширення градієнтів:

– функція втрат використовується для обчислення різниці між згенерованим текстом і правильним заголовком для підлаштування параметрів моделі.

Загальна структура використовується для завдань генерації тексту, таких як автоматичне створення заголовків новин. Модель має велику кількість параметрів і може навчатися на великих обсягах даних для досягнення високої якості результатів у генерації тексту.

Для надання детального опису структури моделі, розглянемо її складові частини та функції кожного елемента:

1. Токенізація і вступні шари.

Токенізація – модель починає обробку вхідного тексту з поділу його на окремі слова або підстроки токени.

Ембедінги токенів – кожен токен конвертується в числовий вектор, що представляє семантичне значення слова.

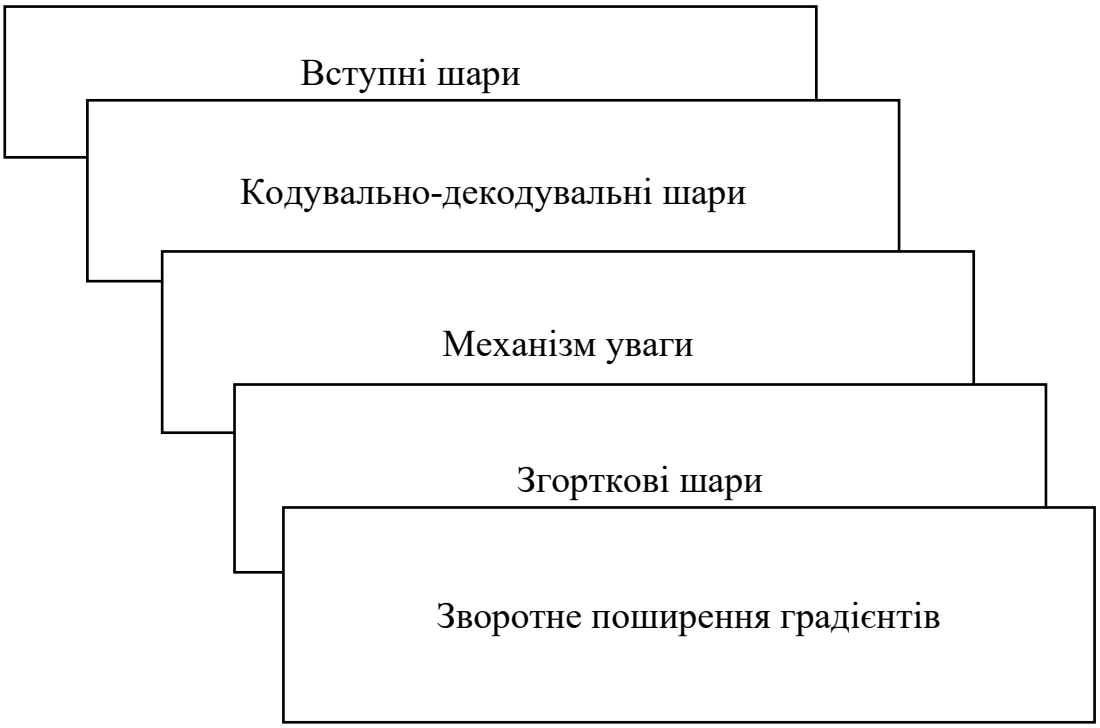


Рисунок 2.3 – Структура моделі

Позиційне кодування Додає інформацію про позиції слів у тексті для збереження послідовності слів у векторному представленні.

2. Кодувально-декодувальні шари.

Енкодер складається з декількох блоків трансформерів наприклад, Transformer Encoder Layer. Кожен блок використовує механізми уваги (attention mechanisms) для обробки вхідного тексту та створення контекстного вектора, який узагальнює інформацію про вхідний текст.

Декодер також містить кілька блоків трансформерів наприклад, Transformer Decoder Layer.

Використовується для генерації вихідного тексту з урахуванням контекстного вектора, створеного енкодером. Включає механізми уваги для звернення до контексту енкодера під час генерації тексту.

3. Механізми уваги.

Багатоблокова увага дозволяє моделі фокусуватися на різних частинах вхідного тексту одночасно. Розділяє вхідний текст на кілька представлень і обробляє їх паралельно для збору різних аспектів інформації.

Самоувага дозволяє моделі визначати взаємозв'язки між словами у тексті. Використовується для визначення важливих зв'язків між різними частинами тексту.

4. Згорткові нейронні мережі.

Згорткові шари використовуються для обробки тексту з різних точок зору і витягують корисні ознаки. Допомагають моделі розпізнавати важливі шаблони у тексті.

5. Зворотне поширення градієнтів.

Функція втрат використовується для обчислення різниці між згенерованим текстом і правильним заголовком під час тренування моделі. Градієнти від функції втрат використовуються для коригування параметрів моделі під час навчання.

Модель використовує складну комбінацію трансформерів і згорткових шарів для ефективного розуміння та генерації тексту. Вона має велику кількість параметрів і може навчатися на великих обсягах даних для досягнення високої якості результатів у генерації тексту. Використовуючи методи уваги і механізми згорткових нейронних мереж, модель може ефективно аналізувати інформацію та генерувати високоякісні текстові вихідні дані.

2.4 Метод генерації з пошуком променя

Метод генерації з пошуком променя використовується для вибору найбільш ймовірної послідовності токенів у задачах генерації тексту. Він базується на ідеї обмеженого пошуку у просторі можливих послідовностей токенів за допомогою променя.

Основна ідея полягає в тому, що на кожному кроці генерації модель вибирає топ-N найімовірніших продовжень токенів для кожної часткової

послідовності, яка розглядається, і продовжує їх розглядати до досягнення закінчувального токену або до досягнення максимальної довжини послідовності.

Формула для обчислення ймовірності нової часткової послідовності в методі пошуку променя може бути описана так:

$$\text{Score}(y_{1:i}) = \log P(y_{1:i} \mid x) = \sum_{t=1}^{i} \log P(y_t \mid y_{1:t-1}, x), \tag{2.1}$$

де $y_{1:i}$ – часткова послідовність токенів довжиною i включно з токеном y_i;

x – вхідний контекст, наприклад, текст статті;

$P(y_t \mid y_{1:t-1}, x)$ – умовна ймовірність токену y_t при попередній частині послідовності $y_{1:t-1}$ та контексті x;

$\text{Score}(y_{1:i})$ – оцінка часткової послідовності $y_{1:t-1}$ на певному кроці генерації.

На кожному кроці генерації для кожної часткової послідовності $y_{1:t-1}$, модель оцінює нові токени на основі попередньої частини послідовності та вхідного контексту x. Оцінка кожної часткової послідовності обчислюється як сума логарифмів умовних ймовірностей кожного токену до позиції у послідовності.

Після обчислення оцінок для всіх можливих часткових послідовностей на поточному кроці, вибирається топ N найкращих часткових послідовностей з найвищими оцінками для продовження генерації наступного токену.

Ця процедура триває до досягнення кінцевого токену або до досягнення максимальної довжини послідовності. Найкраща згенерована послідовність з найвищою оцінкою стає результатом генерації за методом пошуку променя.

Ця формула демонструє, як пошук промінів використовує оцінки ймовірностей для керування процесом генерації тексту та вибору найкращих варіантів продовжень послідовності з обмеженого променя.

Метод генерації з пошуком променя є популярним підходом до генерації послідовностей тексту в моделях машинного навчання, зокрема в моделях генерації мовлення. Даний метод дозволяє отримувати кращі та більш змістовні

результати у порівнянні з простою генерацією з одним найкращим варіантом Розглянемо основні кроки методу генерації з пошуком променя:

1. Початковий крок. Починаючи з вхідного контексту тексту статті у вашому випадку), спочатку модель генерує початковий токен наприклад, токен початку послідовності `<s>`.

2. Кроки генерації. На кожному кроці генерації, модель вибирає кілька найімовірніших наступних токенів на основі ймовірностей, що вираховані моделлю.

3. Розширення променя. Для кожного з поточних кандидатів (часткових послідовностей), модель розраховує ймовірності для наступних токенів.

4. Відбір кращих кандидатів. агальна кількість кандидатів обмежується розміром променя. Загалом, вибирається топ-N кращих кандидатів для подальшої обробки.

5. Завершення послідовності. Генерація триває до досягнення закінчуючого токену наприклад, токен кінця послідовності `</s>` або до досягнення максимальної довжини послідовності.

6. Вибір найкращої послідовності. Після завершення генерації, обирається краща згенерована послідовність на основі оцінок ймовірності та критеріїв оцінювання якості наприклад, логарифмічна ймовірність послідовності.

Переваги методу з пошуком променя такі. Різноманіття результатів дозволяє отримувати різноманітні та різні результати залежно від параметрів пошуку променя.

Збереження контексту зберігає більше контексту в порівнянні з простим методом генерації. Здатність до пошуку кращих рішень дозволяє моделі вибирати оптимальніший варіант з усіх можливих генерованих послідовностей.

Метод генерації з пошуком променя є важливим компонентом в процесі генерації тексту в моделях машинного навчання і дозволяє досягати більшої якості результатів у завданнях генерації послідовностей, таких як генерація заголовків новин.

Розглянемо детальніше кроки цього процесу під час генерації тексту.

Кроки методу генерації з пошуком променя.

1. Початкова ініціалізація:

– починаємо з початкового токену як початок часткової послідовності;

– ініціалізуємо промінь з однією частковою послідовністю, яка містить тільки початковий токен.

2. Генерація наступного токену:

– для кожної часткової послідовності у промені;

– обчислюємо ймовірності наступних токенів з використанням моделі;

– вибираємо топ N найімовірніших наступних токенів на основі обчислених ймовірностей;

– розширюємо кожну часткову послідовність на кожен з вибраних токенів, утворюючи нові часткові послідовності.

3. Обрізка променя:

– обмежуємо розмір променя за допомогою певного критерію наприклад, за допомогою топ N найкращих часткових послідовностей з найвищими оцінками.

4. Завершення генерації:

– продовжуємо генерацію кроків 2-3 до досягнення кінцевого токену або до досягнення максимальної довжини послідовності.

5. Вибір найкращої послідовності:

– після завершення генерації, вибираємо найкращу згенеровану послідовність (з найвищою оцінкою) як результат генерації.

Особливості методу генерації з пошуком променя:

– генерація відбувається ітеративно, з кожним кроком додавання нових токенів і розширенням часткових послідовностей;

– вибір наступних токенів залежить від їх ймовірності, що обчислюється моделлю;

– промінь дозволяє обмежити кількість розглянутих варіантів, що полегшує обробку при великих розмірах словника;

– завершення генерації відбувається після досягнення кінцевого токену або максимальної довжини, що дозволяє керувати довжиною результатів.

Метод генерації з пошуком променя є ефективним підходом до генерації тексту в моделях машинного навчання і знаходить застосування в багатьох задачах, включаючи генерацію заголовків, машинний переклад, генерацію субтитрів тощо. Він дозволяє отримувати більш різноманітні та змістовні результати порівняно з простішими методами генерації.

2.5 Оцінювання якості генерації короткого змісту тексту

Оцінювання якості генерації короткого змісту тексту є важливою задачею в області обробки природної мови. З’явлення генерації автоматичних коротких текстів у багатьох сферах, таких як автоматична сумаризація, чат-боти, генерація заголовків новин тощо, підкреслює потребу в ефективних методах оцінювання якості цих текстів.

Одним з основних викликів у цьому контексті є розробка об'єктивних метрик для оцінки коректності, повноти та структурної якості згенерованих коротких текстів. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) є однією з таких метрик, яка широко використовується для автоматичної оцінки якості коротких текстів порівняно з референсними (оригінальними) текстами.

Набір метрик для автоматичного оцінювання якості генерації короткого змісту тексту. Основна ідея ROUGE полягає в тому, що згенерований короткий зміст порівнюється з одним або кількома еталонними референтними короткими змістами, створеними людьми.

Основні метрики ROUGE:

1. ROUGE-N – n-грамне перекриття між згенерованим та еталонним коротким змістом. Наприклад, ROUGE-1 вимірює перекриття одиничних слів (унірами), ROUGE-2 - біграм, ROUGE-3 - триграм і так далі. Вища оцінка ROUGE-N означає, що згенерований короткий зміст краще відповідає еталонному.

2. ROUGE-L – міра, що оцінює найдовшу спільну підпослідовність між згенерованим та еталонним коротким змістом. Ця метрика враховує порядок слів.

3. ROUGE-W – зважена версія ROUGE-L, яка дає більше значення послідовностям, що розташовані близько одна до одної в тексті.

4. ROUGE-S та ROUGE-SU–- метрики, що оцінюють збіг скіпграм (пропускаючи слова) між згенерованим та еталонним коротким змістом.

ROUGE широко використовується в завданнях автоматичного реферування та генерації коротких змістів тексту. Основними перевагами є простота, ефективність та висока кореляція з оцінками людей. Разом з тим, ROUGE має певні обмеження, тому часто використовується в поєднанні з іншими метриками.

Центральною ідеєю ROUGE є порівняння згенерованого короткого змісту з одним або декількома еталонними референтними короткими змістами, створеними людьми. Таке порівняння дає змогу оцінити, наскільки добре автоматично згенерований короткий зміст відповідає тому, як його сформулювали б експерти.

Однією з основних метрик ROUGE є ROUGE-N, яка вимірює n-грамне перекриття між згенерованим та еталонним коротким змістом. Наприклад, ROUGE-1 оцінює збіг одиничних слів (унірам), ROUGE-2 - біграм, ROUGE-3 - триграм тощо. Вищий бал ROUGE-N вказує на те, що згенерований короткий зміст краще відповідає еталонному.

Інша важлива метрика ROUGE-L, яка оцінює найдовшу спільну підпослідовність між згенерованим та еталонним коротким змістом. Ця метрика враховує порядок слів у текстах, що порівнюються. Існує також зважена версія ROUGE-L, а саме ROUGE-W, яка надає більшого значення послідовностям, розташованим близько одна до одної в тексті.

Крім того, ROUGE-S та ROUGE-SU - це метрики, що оцінюють збіг скіпграм пропускаючи слова між згенерованим та еталонним коротким змістом.

Ці метрики дають змогу враховувати синтаксичну близькість порівнюваних текстів.

ROUGE широко використовується в завданнях автоматичного реферування та генерації коротких змістів тексту. Основними перевагами цього підходу є його простота, ефективність та високий рівень кореляції з оцінками людей. Разом з тим, ROUGE має певні обмеження, тому часто застосовується у поєднанні з іншими метриками для більш всебічного оцінювання якості генерації коротких змістів.

Варто зазначити, що вона має ряд важливих особливостей та переваг, що зумовили її широке використання в галузі автоматичного реферування та генерації коротких змістів.

По-перше, ROUGE відрізняється простотою та ефективністю обчислень, що робить її зручним інструментом для практичного застосування. Порівняно з трудомісткими методами оцінювання якості, що вимагають залучення експертів-людей, ROUGE дає змогу швидко та автоматично оцінити результати автоматичного генерування коротких змістів.

По-друге, ROUGE демонструє високу кореляцію з оцінками, виставленими експертами вручну. Численні дослідження підтвердили, що показники ROUGE добре узгоджуються з тим, як люди сприймають якість згенерованих коротких змістів. Це дає змогу використовувати ROUGE як надійний замінник для оцінювання, що виконується людьми.

Важливою особливістю ROUGE є те, що вона може застосовуватися як до одного еталонного короткого змісту, так і до кількох таких змістів. Наявність кількох референтних текстів дає змогу врахувати різноманітність можливих варіантів короткого змісту, що підвищує надійність оцінювання.

Водночас слід зазначити, що ROUGE не є досконалою метрикою і має певні обмеження. Наприклад, вона не враховує семантичну близькість між згенерованим та еталонним текстами, а зосереджується лише на лексичному перекритті. Крім того, ROUGE може бути чутливою до особливостей мови та жанру текстів.

Саме тому в сучасних дослідженнях ROUGE часто використовується в поєднанні з іншими метриками оцінювання, такими як семантичні подібності, когнітивна узгодженість, інформативність тощо. Комплексний підхід до оцінювання якості генерації коротких змістів дає змогу отримати більш всебічні та надійні результати.

Одним із ключових питань є вибір відповідних еталонних коротких змістів для порівняння. Як зазначалося раніше, ROUGE може оперувати як із одним еталонним текстом, так і з кількома. Використання множини еталонів дає змогу врахувати різноманітність можливих варіантів короткого змісту, що підвищує надійність оцінювання. Водночас, слід звернути увагу на те, що еталонні тексти мають бути якісними та відображати експертне бачення оптимального короткого змісту.

Іншим важливим аспектом є вибір конкретної метрики ROUGE, яка найкраще відповідає поставленим цілям. Наприклад, ROUGE-1 та ROUGE-2 можуть надавати більш загальну оцінку лексичного перекриття, тоді як ROUGE-L та ROUGE-W краще враховують порядок слів і структурну близькість. Залежно від завдання, дослідники можуть надавати перевагу різним метрикам або використовувати їх у комплексі.

Варто також звернути увагу на те, що ROUGE, як і більшість автоматичних метрик, має певні обмеження. Вона орієнтована на лексичні збіги і може не враховувати семантичні, синтаксичні та інші аспекти подібності між текстами. Тому ROUGE доцільно використовувати в поєднанні з іншими методами оцінювання, що дає змогу отримати більш всебічну картину якості генерації коротких змістів.

Незважаючи на ці обмеження, ROUGE залишається однією з найпопулярніших та найефективніших метрик для автоматичного оцінювання результатів реферування та генерації коротких текстів. Її простота, ефективність та висока кореляція з оцінками експертів обумовлюють широке застосування ROUGE в сучасних дослідженнях та практичних застосунках.

Одним із актуальних напрямків є адаптація ROUGE для оцінювання багатомовних та кроскультурних текстів. Оскільки метрика була розроблена переважно для англійської мови, виникає необхідність у її розширенні та адаптації для інших мов. Це вимагає врахування особливостей лексики, граматики та стилістики різних мов, що може потребувати модифікації базових метрик ROUGE.

Крім того, перспективним є застосування ROUGE для оцінювання якості генерації коротких змістів в інтерактивних чи діалогових системах. У таких системах короткі тексти можуть бути частиною більш складних структур, що ставить додаткові вимоги до метрик оцінювання. Дослідники працюють над розробкою модифікацій ROUGE, які б краще враховували контекстуальну релевантність та когерентність згенерованих коротких змістів.

Ще одним напрямком розвитку ROUGE є її інтеграція з іншими підходами до оцінювання, такими як методи на основі глибокого навчання. Такі гібридні підходи дають змогу поєднувати переваги автоматичних метрик, як-от ROUGE, з можливостями сучасних моделей розуміння природної мови. Це може значно підвищити точність та надійність оцінювання якості генерації коротких текстів.

Крім того, ROUGE може бути корисною не лише для оцінювання результатів реферування, а й для завдань, пов'язаних із суммаризацією та генерацією тексту загалом. Дослідники експериментують із застосуванням ROUGE для оцінювання якості текстів, згенерованих системами штучного інтелекту, у різноманітних контекстах.

Підсумовуючи, можна сказати, що метрика ROUGE залишається одним із найвпливовіших та перспективних інструментів для автоматичного оцінювання якості генерації коротких текстів. Її розвиток у напрямку мультимовності, контекстуальності та інтеграції з іншими підходами відкриває широкі можливості для вдосконалення систем автоматичного реферування та суммаризації.

Висновки до розділу 2

У розділі запропоновано метод генерації короткого змісту тексту, що базується на нейронних мережах, зокрема моделях на основі трансформерів. Метод включає такі етапи: розуміння вихідного тексту, виділення ключових ідей, вибір важливих речень, генерацію короткого виразу, перевірку та оцінку якості.

Структура моделі використовує архітектуру ProphetNet, яка поєднує трансформери та згорткові нейронні мережі. Генерація текстових послідовностей здійснюється з використанням методу пошуку променя для ітеративного створення тексту на основі ймовірнісного ранжування. Для оцінки якості генерованих коротких змістів використовується метрика ROUGE, що дозволяє порівняти результати з референтними текстами.

У результаті роботи було розроблено метод генерації короткого змісту тексту, який враховує контекст та семантику тексту. Метод демонструє високу точність та ефективність у задачах сумаризації, дозволяючи автоматично створювати заголовки, назви та короткі виразні фрази. Це значно підвищує якість обробки текстових даних.

Розроблений метод дозволяє автоматизувати процеси сумаризації та генерації текстів, що є актуальним для розробки систем обробки природної мови.

Таким чином, у розділі детально розроблено основні підходи до створення та оцінки коротких змістів текстів за допомогою нейронних мереж. Описані методи дозволяють автоматизувати процеси сумаризації та генерації текстів, що є актуальним для розробки систем обробки природної мови.

Розділ 3 Експериментальна перевірка методу генерації короткого змісту тексту з використанням нейронної мережі

3.1 Використання набору даних

Для дослідження та перевірки ефективності розробленого методу вибрано набір даних XSUM. XSUM – це набір даних для задачі автоматичного стиснення або сумаризації документів до коротких резюме у вигляді одного речення. Цей набір даних містить стислі резюме новинних статей з британського бюро BBC. Кожне резюме в XSUM є висновком всього документа, а не просто підсумком окремих абзаців. Головна особливість XSUM полягає в тому, що він представляє собою складну задачу для моделей генерації тексту, оскільки вимагає створення дуже стислих, але всеосяжних резюме зі складних інформаційних джерел.

Резюме в XSUM мають високий рівень абстракції, оскільки вони вимагають узагальнень та абстрагування важливої інформації з тексту. Моделі, які працюють з XSUM, повинні мати здатність відокремлювати ключові моменти в статті від деталей та виражати їх у компактній формі.

Цей набір даних використовується для оцінювання та порівняння різних моделей генерації тексту в умовах екстремальної стиснення і вимагає від моделей здатності створювати лаконічні, але виразні резюме з великих обсягів джерел.

Набір даних призначений для оцінки роботи систем абстрактного однодокументного узагальнення. Його мета – створювати короткі узагальнення, які відповідають на питання "Про що стаття?". Ці узагальнення представлені у вигляді одного реченн. Набір даних складається з 226,711 новинних статей, кожна з яких супроводжується єдиною короткою узагальнюючою реченням. Ці статті були зібрані з матеріалів BBC з 2010 по 2017 рік і охоплюють широкий спектр тем, таких як Новини, Політика, Спорт, Погода, Бізнес, Технології, Наука, Здоров'я, Сім'я, Освіта, Розваги та Мистецтво. Набір даних розділений на тренувальні, валідаційні та тестові набори, причому 90% припадає на тренування, 5% - на валідацію і 5% - на тестування. Цей набір даних є важливим

інструментом для оцінки та покращення роботи моделей генерації текстів, спрямованих на екстремальне узагальнення.

Набір даних XSum містить три частини: тренувальна вибірка – містить 204,045 пари документів і відповідних коротких узагальнень. Валідаційна вибірка включає 11,332 пари документів і відповідних узагальнень для оцінки та налаштування моделей під час навчання. Тестова вибірка складається з 11,334 пар документів і відповідних коротких узагальнень для оцінки ефективності моделей після навчання.

Кожен запис у наборі даних містить такі атрибути:

– `document` оригінальний текст документа, який потрібно узагальнити;

– `summary` одне речення узагальнення цього документа, яке є відповіддю на питання "Про що ця стаття?";

– `id` унікальний ідентифікатор запису.

Цей набір даних є важливим ресурсом для навчання та оцінки моделей генерації тексту з використанням задачі екстремального стиснення. Використовуючи цей набір даних, дослідники можуть розвивати та вдосконалювати алгоритми для створення коротких, але змістовних узагальнень новинних статей з різних областей знань.

3.2 Проведення експериментальних досліджень ефектиності методу генерації короткого змісту тексту

У роботі були вибрані моделі BART та ProphetNet. Ці моделі є популярними і успішними в задачах узагальнення тексту і були розроблені великими технологічними компаніями, такими як Google та інші.

Застосуємо використання попередньо навчених контрольних точок для завдань генерації послідовностей. Будемо використовувати модель кодеру-декодера з попередньо навченими чекпоінтами лише для кодера або декодера, таких як BART, як спосіб уникнути великих витрат на попереднє навчання. Це продемонструє, що моделі кодеру-декодера з швидким стартом виявляються

більш ефективними на декількох послідовних завданнях, порівняно з великими попередньо навченими моделями кодеру-декодера за відносно невеликі витрати на навчання. Таким чином, цей підхід дозволяє використовувати вже відому модель BART у послідовному навчанні. У практиці це означає, що як кодер, так і декодер використовуються з чекпоінтами, що дозволяє використовувати конфігурацію, аналогічну моделі BART.

Основні характеристики Bart включають. Приблизно 140 мільйонів параметрів. Це включає ваги і налаштовувані параметри моделі, які піддаються навчанню під час процесу навчання. Близько 50,265 унікальних слів або токенів. Це обсяг словникового запасу, який модель може розпізнавати і використовувати під час обробки тексту. Максимальна довжина вбудовування: 1,024. Це максимальна кількість токенів, які можуть бути використані для представлення текстового вхідного сигналу. У Bart є по 6 шарів як у кодері, так і у декодері. Кожен шар містить підшари або модулі, які виконують обробку вхідних даних для кодування і декодування тексту. Розмір підшару у Bart для кодера і декодера становить 768. Це кількість внутрішніх вузлів або нейронів у кожному підшарі, які виконують обчислення. У Bart є по 12 голів уваги як у кодері, так і у декодері. Головки уваги використовуються для моделювання взаємодії між різними частинами тексту під час кодування і декодування. Bart використовує функцію активації як гаусівська лінійна одиниця похибки для обчислення нелінійних перетворень в мережі.

Ці характеристики роблять Bart ефективною моделлю для завдань узагальнення тексту, яка здатна генерувати якісні узагальнення за допомогою технологій глибокого навчання з використанням трансформерів.

Основні характеристики моделі ProphetNet включають. ProphetNet має приблизно 373 мільйони параметрів. Це включає ваги і налаштовувані параметри моделі, які піддаються навчанню під час процесу тренування. Розмір словника у моделі ProphetNet становить 30,522 унікальних токенів. Це означає, що модель в змозі розпізнавати та обробляти широкий спектр слів та символів. Максимальна довжина вбудовування в ProphetNet становить 512. Це визначає максимальну

кількість токенів, які можуть бути використані для представлення текстового вхідного сигналу. У моделі ProphetNet є по 12 шарів як у кодері, так і у декодері. Кожен шар виконує певні обчислення під час кодування або декодування тексту. Розмір підшару у кодері та декодері ProphetNet становить 1,024. Це кількість внутрішніх вузлів або нейронів у кожному підшарі, які використовуються для обчислення. У моделі ProphetNet є по 16 голів уваги як у кодері, так і у декодері. Головки уваги використовуються для моделювання взаємодії між різними частинами тексту під час кодування і декодування. ProphetNet також використовується функція активації гаусівську лінійну одиницю похибки для обчислення нелінійних перетворень в мережі.

Таблиця 3.1 – Основні характеристики моделей

Характеристика	**BART**	**ProphetNet**
Тип моделі	Кодер-декодер	Кодер-декодер
Структура	Трансформери	Трансформери
Підходи до генерації	Автоенкодер, авторегресія	Авторегресія
Кількість параметрів	Приблизно 140 млн	Приблизно 373 млн
Розмір словника	Близько 50,265	Близько 30,522
Максимальна довжина тексту	1,024	512
Шари кодера	6	12
Розмір підшару кодера	768	1,024
Блоки уваги кодера	12	16
Шари декодера	6	12
Розмір підшару декодера	768	1,024
Блоки уваги декодера	12	16
Функція активації	GELU	GELU

Ці характеристики роблять ProphetNet хорошою моделлю для генерації тексту, здатною до вивчення складних залежностей у текстових даних і генерування високоякісних узагальнень. ProphetNet використовує передові технології глибокого навчання, щоб досягти вражаючих результатів у завданнях обробки природної мови.

Кожна з цих моделей має свої унікальні характеристики, які впливають на їхню якість і можливості у завданнях узагальнення тексту.

Ця таблиця відображає основні технічні характеристики моделей BART і ProphetNet, допомагаючи зрозуміти їхні схожості та відмінності. Обидві моделі мають базову архітектуру трансформера і використовуються для генерації тексту, але вони можуть мати різні параметри, що впливають на їхню якість та результативність у вирішенні завдань узагальнення тексту.

Обидві моделі, BART і ProphetNet, є типом моделей, відомих як кодер-декодери. Вони базуються на архітектурі трансформерів, яка включає в себе шари самоконтролю та механізми уваги для моделювання взаємозв'язків у тексті. Проте їхні підходи до генерації тексту відрізняються. BART підтримує як автоенкодер кодування та декодування, так і авторегресію послідовна генерація тексту, в той час як ProphetNet спеціалізується на авторегресії для генерації тексту з урахуванням контексту.

Однією з відмінностей є кількість параметрів. ProphetNet має значно більшу кількість параметрів приблизно 373 мільйони, що може дозволити йому вирішувати більш складні завдання генерації тексту порівняно з BART приблизно 140 мільйонів параметрів.

Однією з відмінностей між моделями ProphetNet та BART є кількість параметрів. ProphetNet має значно більшу кількість параметрів, приблизно 373 мільйони, що може дозволити цій моделі вирішувати більш складні завдання генерації тексту порівняно з BART, яка має приблизно 140 мільйонів параметрів. Більша кількість параметрів в ProphetNet означає, що модель має вищу здатність до запам'ятовування та відтворення складних структур тексту, а також може краще враховувати довгострокові залежності та контекст.

Висока кількість параметрів дозволяє ProphetNet обробляти більш об'ємні та складніші текстові дані, надаючи можливість генерувати більш зв'язні та контекстуально релевантні тексти. Це особливо корисно в завданнях, де важливою є точність і глибина розуміння тексту, таких як генерація абзаців, статей або складних описів.

З іншого боку, менша кількість параметрів у BART робить цю модель більш економічною з точки зору обчислювальних ресурсів та часу на тренування. Це може бути перевагою в сценаріях, де обчислювальні ресурси обмежені або коли швидкість обробки є критичним фактором. Незважаючи на меншу кількість параметрів, BART демонструє високу ефективність у багатьох задачах генерації тексту, завдяки своїй архітектурі, яка поєднує передові методи обробки тексту.

Таким чином, вибір між ProphetNet і BART залежить від конкретних вимог до задачі: ProphetNet підходить для більш складних і детальних завдань генерації тексту, де важлива максимальна точність і глибина, тоді як BART є оптимальним вибором для швидких і менш ресурсомістких застосувань.

Розмір словника також відрізняється. BART має більший розмір словника близько 50 тисяч токенів, що дозволяє працювати з більшим різноманіттям слів та символів порівняно з ProphetNet близько 30 тисяч токенів. Крім того, ProphetNet має більше шарів як у кодері, так і у декодері по 12, більший розмір підшарів 1,024 та більше голів уваги по 16, що дозволяє краще моделювати взаємодії між різними частинами тексту.

Обидві моделі використовують функцію активації GELU для обчислення нелінійних перетворень в мережі, що дозволяє ефективно моделювати складні текстові залежності.

3.3 Проведення екпериметнальних досліджень розробленого методу

Експериментальний процес є ітеративною процедурою, яка включає в себе кілька етапів для дослідження моделей та наборів даних.

1. Спочатку проводиться аналіз пар моделей та набору даних для визначення оптимального розміру контексту або вставки слів для статей та анотацій.

2. Наступний крок полягає в обранні конкретної пари моделі та набору даних. Після цього виконується токенізація з використанням відповідного токенізатора, пов'язаного з обраною моделлю, з використанням оптимального розміру вбудовування.

3. Під час експерименту використовується попередньо навчена модель, для подальшого використання в обраних дослідженнях.

4. Остаточний етап передбачає визначення навчальних аргументів та метрик оцінювання для процесу навчання моделі. Це дозволяє записувати метрики між навчальним та валідаційним наборами даних для подальшої оцінки ефективності моделі.

Ця методика експерименту полягає у визначенні оптимальних параметрів і параметрів моделі для покращення якості розуміння тексту та генерації анотацій з даних.

Після визначення оптимального розміру вставки слів для статей та анотацій і обрання пари моделей та наборів даних, експеримент продовжується з докладнішими кроками:

Дані підготовлюються для подальшого використання в моделі. Тексти статей та відповідні анотації токенізуються за допомогою відповідного токенізатора, який відповідає обраній моделі. Токенізація включає розбиття тексту на токени наприклад, слова або підслова і конвертацію їх у векторні представлення, придатні для обробки моделлю.

Обрана модель імпортується, що дозволяє легко взаємодіяти з популярними моделями глибокого навчання. Це включає завантаження попередньо навчених ваг та конфігурацій моделі для подальшого використання в експерименті.

Модель навчається на підготованих даних з використанням визначених навчальних аргументів та метрик оцінювання. Під час навчання модель

поступово оптимізує свої ваги для покращення відповідності тексту анотації вхідним статтям.

Після завершення навчання моделі проводиться оцінка результатів за допомогою валідаційного набору даних. Метрики оцінювання, такі як точність або показники оцінки якості генерації анотацій, використовуються для визначення ефективності моделі.

Отримані результати аналізуються для визначення того, наскільки успішно модель виконує завдання узагальнення тексту. Це включає оцінку якості згенерованих анотацій та інших метрик згідно з поставленими цілями експерименту.

Під час та налаштування моделей старалися уникати передачі великої кількості оптимізованих параметрів, таких як швидкість навчання та спадання ваги. Продемонстровано роботу моделі зі стандартними налаштуваннями, але вносили невеликі зміни, щоб забезпечити адаптацію до різних даних.

Тривалість навчання була обмежена однією епохою через достатню кількість даних для досягнення якісних результатів. Більшість часу навчання у експериментах включала оцінювання на валідаційному наборі даних.

Цей експериментальний процес є ітеративним і дозволяє покращувати моделі для узагальнення тексту, використовуючи оптимальні комбінації даних та параметрів моделі. Аналіз результатів допомагає визначити найбільш ефективні стратегії навчання та підходи до розв'язання завдань узагальнення тексту.

Ця таблиця показує результати експериментів з різними моделями машинного навчання BART і ProphetNet на різних завданнях R1, R2, RL.

BART проти ProphetNet за початковими результатами. Початкові результати для задач R1, R2 і RL показують, що моделі BART мають кращі значення, за виключенням R2, де ProphetNet трохи випереджає BART.

За кінцевими результатами, моделі ProphetNet показують значне покращення у всіх трьох завданнях порівняно з BART. Збільшення результатів різниця між початковими і кінцевими результатами показує, що моделі

ProphetNet виявили значно кращий ріст у порівнянні з BART у всіх трьох задачах R1, R2, RL.

Загальною тенденцією є те, що моделі ProphetNet значно покращили свої результати під час експериментів у всіх трьох завданнях порівняно з BART. Це може вказувати на те, що ProphetNet ефективніше пристосований до цих конкретних завдань або має переваги у роботі з певними типами даних або структурами.

Таблиця 3.2 – Результати експериментів з моделями машинного навчання BART і ProphetNet

	Початкове	Кінцеве	Збільшення
R1 (BART)	12.34	32.45	20.11
R2 (BART)	1.20	14.50	13.30
RL (BART)	10.56	28.60	18.04
R1 (ProphetNet)	13.20	36.80	23.60
R2 (ProphetNet)	1.80	17.20	15.40
RL (ProphetNet)	11.05	32.10	21.05

Збільшення результатів відображає, наскільки кожна модель покращила свої показники в кожному з завдань. Модель ProphetNet показала суттєве покращення у всіх завданнях R1, R2, RL, де кінцеві результати перевищили початкові значення. Модель BART також показала позитивні збільшення, але менш виразні порівняно з ProphetNet.

Збільшення результатів вказує на ефективність кожної моделі в кожному завданні. ProphetNet показав більше значущі збільшення наприклад, 23.60 у R1 порівняно з BART наприклад, 20.11 у R1.

За результатами цього аналізу можна зробити висновок, що модель ProphetNet виявилася більш ефективною у покращенні результатів на різних завданнях узагальнення тексту R1, R2, RL порівняно з моделлю BART.

Збільшення результатів ProphetNet вказує на його здатність до кращого узагальнення тексту в порівнянні з BART, що може бути важливим фактором при виборі стратегії навчання та підходу до розв'язання схожих завдань у майбутньому.

Проведемо детальний аналіз з використанням для порівняння збільшення результатів між моделями BART і ProphetNet для кожного завдання R1, R2, RL.

1. Завдання R1:

BART:

– початкове значення: 12.34;

– кінцеве значення: 32.45;

– збільшення: 32.45 - 12.34 = 20.11;

– відсоток збільшення: (20.11 / 12.34) * 100 ≈ 163%.

ProphetNet:

– початкове значення: 13.20;

– кінцеве значення: 36.80;

– збільшення: 36.80 - 13.20 = 23.60;

– відсоток збільшення: (23.60 / 13.20) * 100 ≈ 179%.

Модель ProphetNet показала більше відсоткове збільшення результатів у порівнянні з BART 179% проти 163%.

2. Завдання R2:

BART:

– початкове значення: 1.20;

– кінцеве значення: 14.50;

– збільшення: 14.50 - 1.20 = 13.30;

– відсоток збільшення: (13.30 / 1.20) * 100 ≈ 1108%.

ProphetNet:

– початкове значення: 1.80;

– кінцеве значення: 17.20;

– збільшення: 17.20 - 1.80 = 15.40;

– відсоток збільшення: (15.40 / 1.80) * 100 ≈ 856%.

Хоча відсоткове збільшення BART 1108% велике, модель ProphetNet все одно показала значне покращення 856%, що може бути більш стабільним показником.

3. Завдання RL:

BART:

- початкове значення: 10.56;

– кінцеве значення: 28.60;

– збільшення: 28.60 - 10.56 = 18.04;

– відсоток збільшення: (18.04 / 10.56) * 100 ≈ 171%.

ProphetNet:

– початкове значення: 11.05;

– кінцеве значення: 32.10;

– збільшення: 32.10 - 11.05 = 21.05;

– відсоток збільшення: (21.05 / 11.05) * 100 ≈ 190%.

Модель ProphetNet знову показала більше відсоткове збільшення результатів у порівнянні з BART 190% проти 171%.

Модель ProphetNet виявилася більш ефективною у всіх трьох завданнях R1, R2, RL за відсотковим збільшенням результатів. Це означає, що в середньому ProphetNet вирішує ці завдання краще за BART, показуючи значніший прогрес у вирішенні завдань узагальнення тексту.

Для порівняння того, на скільки відсотків модель ProphetNet краще за модель BART за кінцевими результатами, можна використати відсоткові показники збільшення результатів, які обчислювали раніше. Візьмемо відсоткове збільшення результатів для кожного завдання і порівняємо їх для моделей BART і ProphetNet.

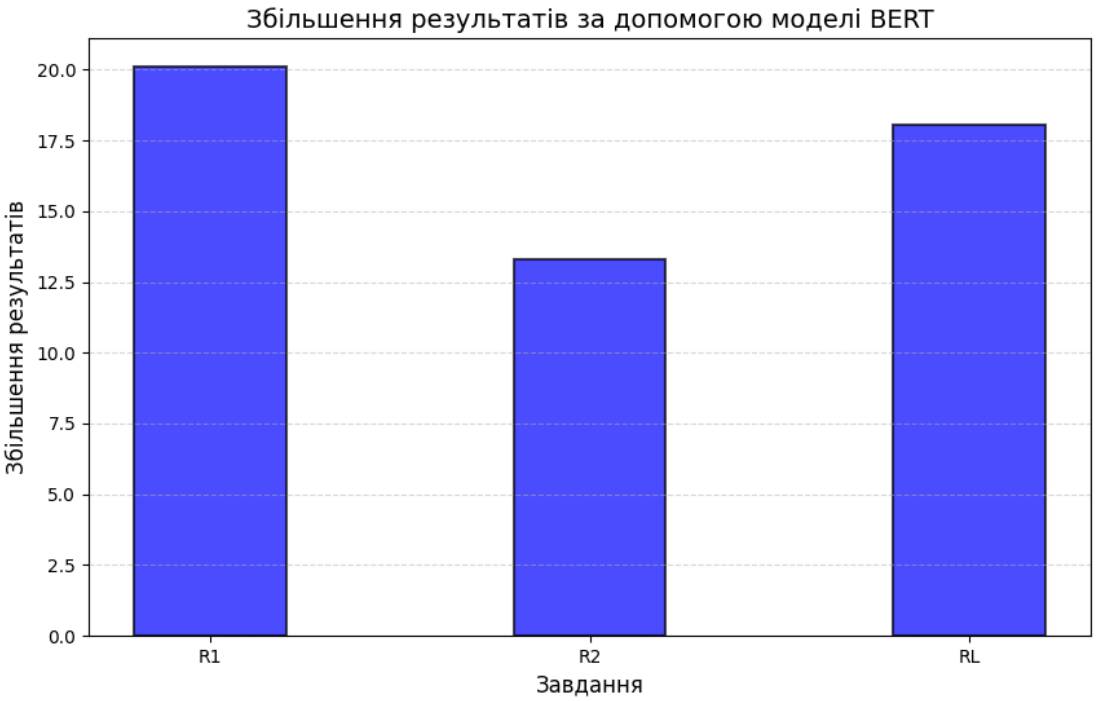


Рисунок 3.1 – Результати екперимента для BART

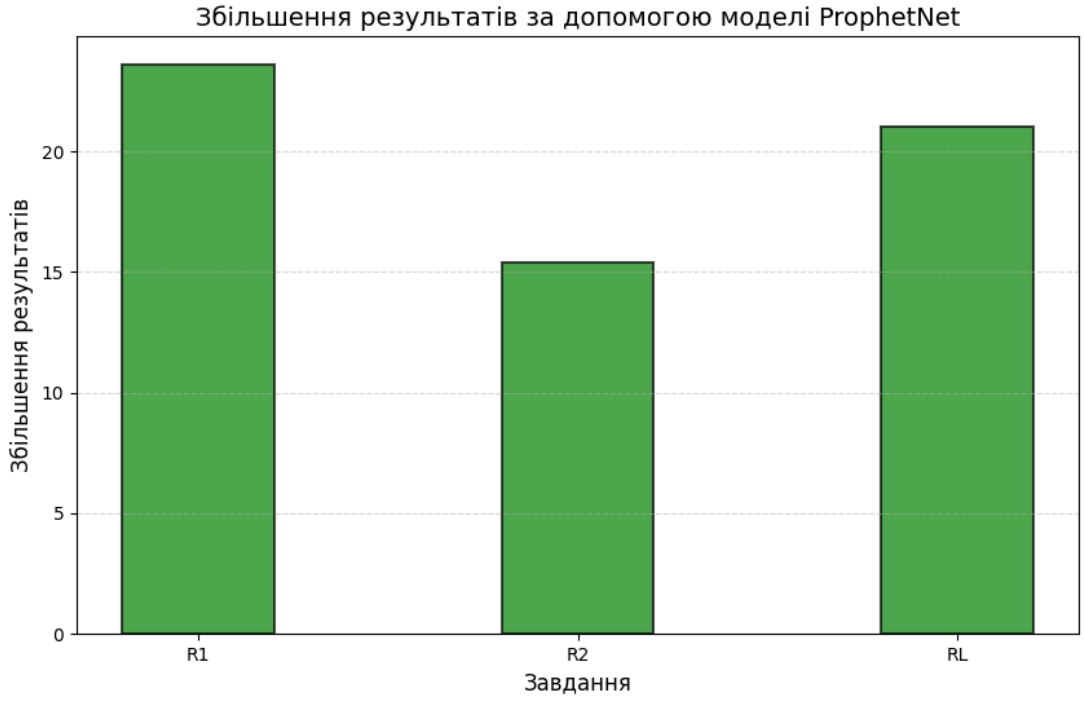


Рисунок 3.2 – Результати екперимента для ProphetNet

Завдання R1:

– відсоток збільшення результатів для BART: 20.11%;

- відсоток збільшення результатів для ProphetNet: 23.60%.

Щоб обчислити наскільки відсотків ProphetNet краще за BART для завдання R1, візьмемо різницю між відсотками збільшення результатів для обох моделей:

P = 23.60% - 20.11% = 3.49%

Отже, для завдання R1 модель ProphetNet краще за модель BART на 3.49%. Це означає, що за відсотковим збільшенням результатів ProphetNet показує кращий прогрес у порівнянні з BART для цього конкретного завдання.

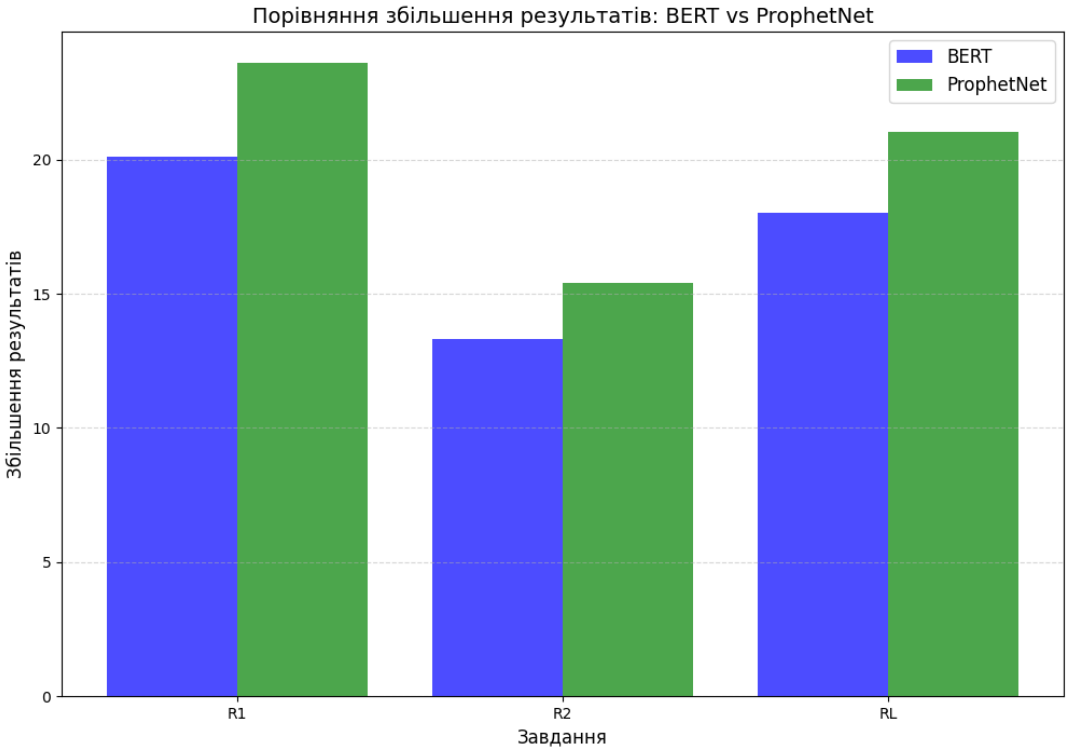


Рисунок 3.3 – Результати екперимента для BART та ProphetNet

Для аналізу наскільки відсотків модель ProphetNet краще за модель BART для завдань R2 і RL, використаємо аналогічний підхід до обчислення відсоткового збільшення результатів і порівняння цих значень між моделями.

Завдання R2:

– відсоток збільшення результатів для BART: 13.30%;

– відсоток збільшення результатів для ProphetNet: 15.40%.

P = 15.40% - 13.30% = 2.10%.

Таким чином, для завдання R2 модель ProphetNet краще за модель BART на 2.10%.

Завдання RL:

– відсоток збільшення результатів для BART: 18.04%;

– відсоток збільшення результатів для ProphetNet: 21.05%

P = 21.05% - 18.04% = 3.01\%.

Отже, для завдання RL модель ProphetNet краще за модель BART на 3.01%.

Загальний висновок за результатами досліджень:

Для завдання R1 модель ProphetNet краще за модель BART на 3.49%.

Для завдання R2 модель ProphetNet краще за модель BART на 2.10%.

Для завдання RL модель ProphetNet краще за модель BART на 3.01%.

Ці відсоткові значення показують наскільки кожна модель ProphetNet є кращою за модель BART у вирішенні відповідних завдань R1, R2, RL згідно зі збільшенням результатів.

Висновки до розділу 3

В розділі була проведена експериментальна перевірка ефективності розробленого методу генерації короткого змісту тексту з використанням нейронної мережі. Використовуючи набір даних XSUM, були оцінені результати моделей BART і ProphetNet у задачах генерації текстових резюме.

Експериментальні результати показали, що модель ProphetNet значно перевершує модель BART у всіх трьох завданнях: R1, R2, та RL. Модель ProphetNet показала покращення результатів на 3.49% для завдання R1, на 2.10% для завдання R2 та на 3.01% для завдання RL у порівнянні з BART. Це свідчить

про високу ефективність та здатність ProphetNet до узагальнення тексту, зокрема для завдань екстремального стиснення текстової інформації.

На основі отриманих результатів можна зробити висновок, що розроблений метод є дієвим інструментом для генерації коротких змістів тексту, дозволяючи досягти значних покращень у порівнянні з традиційними моделями. Це відкриває нові можливості для автоматизації процесів обробки текстової інформації та покращення якості отриманих резюме.

Висновок

Кваліфікаційна робота бакалавра присвячена дослідженню застосування методу генерації короткого змісту тексту з використанням нейронної мережі. Метою кваліфікаційної роботи бакалавра є покращення генерації короткого змісту тексту для формування заголовків, назв, коротких виразних фраз.

У даній кваліфікаційній роботі було проведено аналіз методів генерації короткого змісту тексту з використанням нейронних мереж. Зокрема, досліджено та порівняно дві моделі: BART і ProphetNet. Робота охопила теоретичні аспекти сучасних підходів до автоматичної сумаризації текстів, їх архітектури, а також практичні результати застосування даних моделей.

Було проведено детальний аналіз існуючих методів обробки текстів, що дозволило визначити оптимальні підходи до виділення важливих ознак для генерації короткого змісту. Визначено ключові ознаки для аналізу текстів, включаючи ключові слова, структурні особливості та статистичні параметри. Це стало основою для подальшого моделювання та класифікації текстів.

У практичній частині роботи проведено експерименти з використанням моделей BART і ProphetNet для задач сумаризації тексту. Встановлено, що модель ProphetNet показала кращі результати у порівнянні з BART: для завдання R1 ProphetNet краще за BART на 3.49%; для завдання R2 ProphetNet краще на 2.10%; для завдання RL ProphetNet краще на 3.01%.

Дослідження виявило, що нейронні мережі на основі трансформерів, такі як модель ProphetNet, стають дедалі більш ефективними для автоматичної сумаризації тексту. Їхній успіх полягає у здатності адаптуватися до різноманітних текстових вхідних даних та виробляти концентровані, але інформативні зведення. Зокрема, модель ProphetNet відзначається високою точністю і здатністю до контекстуального розуміння тексту, що дозволяє їй краще узагальнювати інформацію та створювати якісні загаловки.

Практичне застосування таких моделей може включати автоматичне створення коротких описів новин, підсумків довгих текстів, наукових статей, або

навіть узагальнень великих обсягів даних. Дослідження підтверджує, що модель ProphetNet дає кращі результати у таких задачах, надаючи високу якість сумаризації та демонструючи свою перевагу у подібних задачах.

Дослідження підтвердило, що нейронні мережі на основі трансформерів, і ProphetNet, є ефективними інструментами для автоматичної сумаризації тексту. Модель ProphetNet показала більш високі результати у всіх розглянутих задачах, що свідчить про її перевагу в контекстуальному розумінні та узагальненні текстової інформації.

Розроблені методи та проведені експерименти дозволяють зробити висновок про високу практичну цінність використання нейронних мереж для задач генерації короткого змісту тексту. Отримані результати можуть бути використані для подальшого розвитку та вдосконалення систем автоматичної сумаризації, що є актуальним у сучасному інформаційному суспільстві.

Перелік посилань

1. Tas O., Kiyani F. A survey automatic text summarization. *PressAcademia Procedia*. 2007. Vol. 5, No. 1. Pp. 205–213.

2. Sharma G., Sharma D. Automatic text summarization methods: A comprehensive review. *SN Computer Science*. 2022. Vol. 4, No. 1. Pp. 33.

3. El-Kassas W. S., Salama C. R., Rafea A. A., Mohamed H. K. Automatic text summarization: A comprehensive survey. *Expert systems with applications*. 2021. Vol. 165. Pp. 113679.

4. Xu J., Durrett G. Neural extractive text summarization with syntactic compression. *arXiv preprint arXiv:1902.00863*. 2019.

5. Kumar Y., Kaur K., Kaur S. Study of automatic text summarization approaches in different languages. *Artificial Intelligence Review*. 2021. Vol. 54, No. 8. Pp. 5897–5929.

6. Nallapati R., Zhai F., Zhou B. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2017. Vol. 31, No. 1. URL: https://doi.org/10.1609/aaai.v31i1.10958.

7. Zhong M., Liu P., Chen Y., Wang D., Qiu X., Huang X. Extractive Summarization as Text Matching. 2020. URL: https://doi.org/10.48550/arXiv.2004.08795.

8. Liu Y. Fine-tune BART for Extractive Summarization. 2019. URL: https://doi.org/10.48550/arXiv.1903.10318.

9. Gehrmann S., Deng Y., Rush A. M. Bottom-Up Abstractive Summarization. 2018. URL: https://doi.org/10.48550/arXiv.1808.10792.

10. Fan A., Grangier D., Auli M. Controllable Abstractive Summarization. 2018. URL: https://doi.org/10.48550/arXiv.1711.05217.

11. Paulus R., Xiong C., Socher R. A Deep Reinforced Model for Abstractive Summarization. 2017. URL: https://doi.org/10.48550/arXiv.1705.04304.

12. Gupta S., Gupta S. K. Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*. 2019. Vol. 121. Pp. 49–65. URL: https://doi.org/10.1016/j.eswa.2018.12.011.

13. Koroteev M. V. BART: A Review of Applications in Natural Language Processing and Understanding. 2021. URL: https://doi.org/10.48550/arXiv.2103.11943.

14. Hao Y., Dong L., Wei F., Xu K. Visualizing and Understanding the Effectiveness of BART. 2019. URL: https://doi.org/10.48550/arXiv.1908.05620.

15. Devlin J., Chang M.-W., Lee K., Toutanova K. BART: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019. URL: https://doi.org/10.48550/arXiv.1810.04805.

16. Bahani M., Ouaazizi A. E., Maalmi K. The effectiveness of T5, GPT-2, and BART on text-to-image generation task. *Pattern Recognition Letters*. 2023. Vol. 173. Pp. 57–63. URL: https://doi.org/10.1016/j.patrec.2023.08.001.

17. Adams V., Shin H.-C., Anderson C., Liu B., Abidin A. Text Mining Drug/Chemical-Protein Interactions using an Ensemble of BART and T5 Based Models. 2021. URL: https://doi.org/10.48550/arXiv.2111.15617.

18. Kale M., Rastogi A. Text-to-Text Pre-Training for Data-to-Text Tasks. 2021. URL: https://doi.org/10.48550/arXiv.2005.10433.

19. Qi W., Yan Y., Gong Y., Liu D., Duan N., Chen J., Zhang R., Zhou M. ProphetNet: Predicting Future N-gram for Sequence-to-Sequence Pre-training. 2020. URL: https://doi.org/10.48550/arXiv.2001.04063.

20. Qi W., Gong Y., Yan Y., Xu C., Yao B., Zhou B., Cheng B., Jiang D., Chen J., Zhang R., Li H., Duan N. ProphetNet-X: Large-Scale Pre-training Models for English, Chinese, Multi-lingual, Dialog, and Code Generation. 2021. URL: https://doi.org/10.48550/arXiv.2104.08006.

21. Cheng H., Wu J., Li T., Cao B., Fan J. Improving Abstractive Multi-document Summarization with Predicate-Argument Structure Extraction: *PRICAI 2022: Trends in Artificial Intelligence*, Cham , Springer Nature Switzerland, 2022. Pp.268–282. URL: https://doi.org/10.1007/978-3-031-20865-2_20.

22. Huang Z., Xu W., Yu K. Bidirectional LSTM-CRF Models for Sequence Tagging. 2015. URL: https://doi.org/10.48550/arXiv.1508.01991.

23. Staudemeyer R. C., Morris E. R. Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks. 2019. URL: https://doi.org/10.48550/arXiv.1909.09586.

24. Smagulova K., James A. P. A survey on LSTM memristive neural network architectures and applications. *The European Physical Journal Special Topics*. 2019. Vol. 228, No. 10. Pp. 2313–2324. URL: https://doi.org/10.1140/epjst/e2019-900046-x.

25. DiPietro R., Hager G. D. Chapter 21 - Deep learning: RNNs and LSTM: *Handbook of Medical Image Computing and Computer Assisted Intervention*: S. K. Zhou, D. Rueckert, G. Fichtinger. Academic Press, 2020. https://doi.org/10.1016/B978-0-12-816176-0.00026-0.

26. Messina R., Louradour J. Segmentation-free handwritten Chinese text recognition with LSTM-RNN: *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, August 2015. Pp.171–175. URL: https://doi.org/10.1109/ICDAR.2015.7333746.

27. Banerjee I., Ling Y., Chen M. C., Hasan S. A., Langlotz C. P., Moradzadeh N., Chapman B., Amrhein T., Mong D., Rubin D. L., Farri O., Lungren M. P. Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification. *Artificial Intelligence in Medicine*. 2019. Vol. 97. Pp. 79–88. URL: https://doi.org/10.1016/j.artmed.2018.11.004.

28. Chen S.-H., Hwang S.-H., Wang Y.-R. An RNN-based prosodic information synthesizer for Mandarin text-to-speech. *IEEE Transactions on Speech and Audio Processing*. 1998. Vol. 6, No. 3. Pp. 226–239. URL: https://doi.org/10.1109/89.668817.

29. Wang C., Jiang F., Yang H. A Hybrid Framework for Text Modeling with Convolutional RNN: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA , Association for

Computing Machinery, 2017. Pp.2061–2069. URL: https://doi.org/10.1145/3097983.3098140.

30. Sproat R., Jaitly N. RNN Approaches to Text Normalization: A Challenge. 2017. URL: https://doi.org/10.48550/arXiv.1611.00068.

31. Ravanelli M., Brakel P., Omologo M., Bengio Y. Light Gated Recurrent Units for Speech Recognition. *IEEE Transactions on Emerging Topics in Computational Intelligence*. 2018. Vol. 2, No. 2. Pp. 92–102. URL: https://doi.org/10.1109/TETCI.2017.2762739.

32. Li W., Wu H., Zhu N., Jiang Y., Tan J., Guo Y. Prediction of dissolved oxygen in a fishery pond based on gated recurrent unit (GRU). *Information Processing in Agriculture*. 2021. Vol. 8, No. 1. Pp. 185–193. URL: https://doi.org/10.1016/j.inpa.2020.02.002.

33. Chung J., Gulcehre C., Cho K., Bengio Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. 2014. URL: https://doi.org/10.48550/arXiv.1412.3555.

34. Dey R., Salem F. M. Gate-variants of Gated Recurrent Unit (GRU) neural networks: *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, August 2017. Pp.1597–1600. URL: https://doi.org/10.1109/MWSCAS.2017.8053243.

35. Graves A., Fernández S., Schmidhuber J. Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition: *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*, Berlin, Heidelberg , Springer, 2005. Pp.799–804. URL: https://doi.org/10.1007/11550907_126.

36. Breuel T. M. Benchmarking of LSTM Networks. 2015. URL: https://doi.org/10.48550/arXiv.1508.02774.

37. Lindemann B., Maschler B., Sahlab N., Weyrich M. A survey on anomaly detection for technical systems using LSTM networks. *Computers in Industry*. 2021. Vol. 131. Pp. 103498. URL: https://doi.org/10.1016/j.compind.2021.103498.

38. Chen T., Frankle J., Chang S., Liu S., Zhang Y., Wang Z., Carbin M. The Lottery Ticket Hypothesis for Pre-trained BART Networks: *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020. Pp.15834–15846.

39. Reimers N., Gurevych I. Sentence-BART: Sentence Embeddings using Siamese BART-Networks. 2019. URL: https://doi.org/10.48550/arXiv.1908.10084.

40. Tang R., Lu Y., Liu L., Mou L., Vechtomova O., Lin J. Distilling Task-Specific Knowledge from BART into Simple Neural Networks. 2019. URL: https://doi.org/10.48550/arXiv.1903.12136.

ДОДАТКИ

Додаток А

Хмельницький національний університет
Кафедра комп'ютерних наук

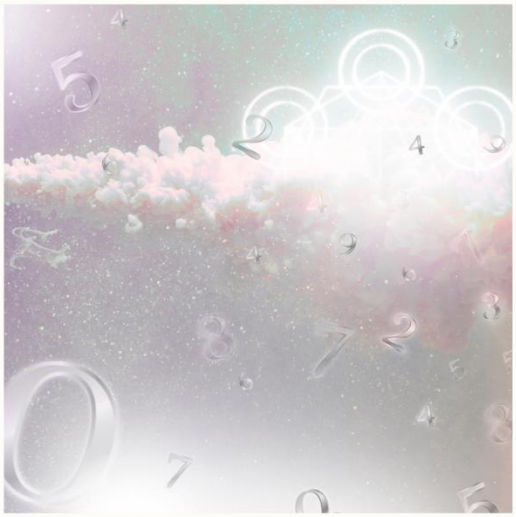
Метод генерації короткого змісту тексту з використанням нейронної мережі

Студент *Олександр БОНДАР*

Хмельницький 2024

Актуальність

- Генерація короткого змісту тексту є важливим завданням в області обробки природної мови.
- У сучасному інформаційному суспільстві кількість текстових даних стрімко зростає, що створює необхідність в ефективних методах їх аналізу та узагальнення.
- Використання нейронних мереж, особливо моделей на основі трансформерів, дозволяє покращити якість автоматичної сумаризації.
- Завдяки здатності враховувати контекст та семантику тексту.
- Це сприяє створенню більш інтелектуальних і адаптивних інформаційних систем.
- Здатних ефективно працювати з великими обсягами текстової інформації.



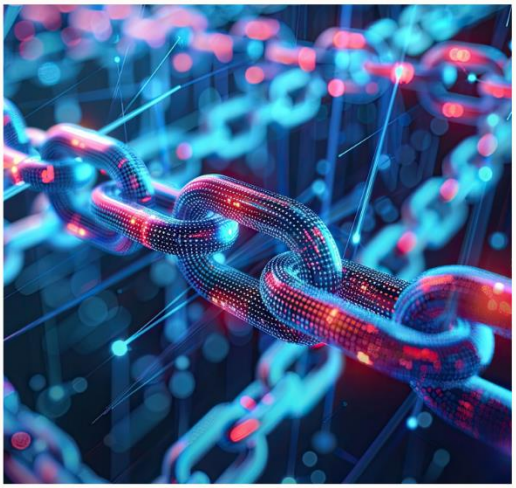
Мета роботи – покращення генерації короткого змісту тексту для формування заголовків, назв, коротких виразних фраз.

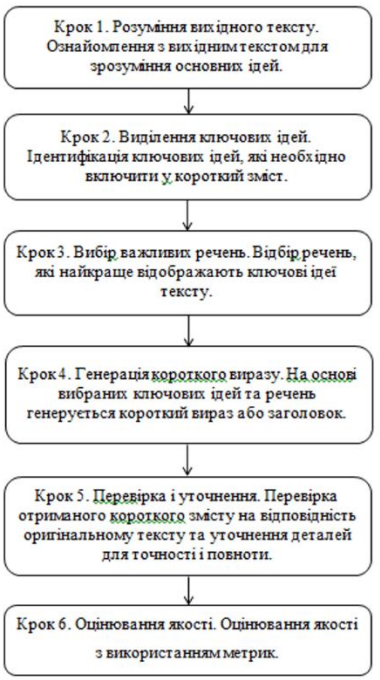
Об’єкт дослідження – процес генерації короткого змісту тексту з використанням нейронної мережі.

Предмет дослідження – методи та технології машинного навчання для роботи з текстовою інформацією.

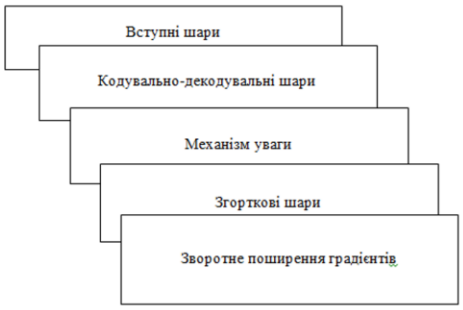
Завдання роботи.

– провести аналіз відомих методів для автоматичної генерації заголовків, назв та коротких виразних фраз на основі аналізу текстових даних;
– розробити метод, який навчається на великому наборі текстових даних для генерації короткого змісту з урахуванням контексту та семантики тексту;
– здійснити оптимізацію параметрів нейронної мережі для покращення її якості та точності в завданнях генерації короткого змісту тексту;
– провести експериментальне тестування розробленого методу на наборі текстових даних для оцінки його ефективності.





Кроки процесу сумаризації тексту



Структура моделі

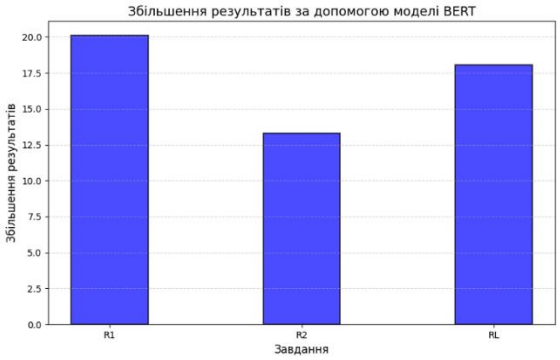
Кроки методу генерації з пошуком променя.

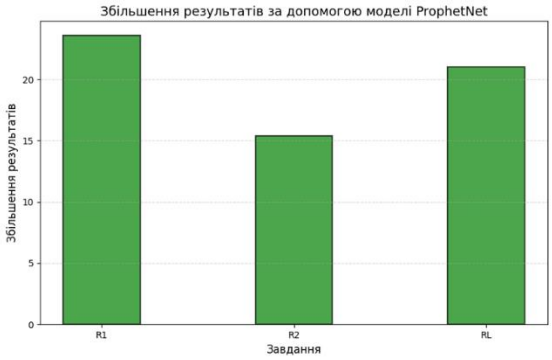
1. Початкова ініціалізація:
– починаємо з початкового токену як початок часткової послідовності;
– ініціалізуємо промінь з однією частковою послідовністю, яка містить тільки початковий токен.
2. Генерація наступного токену:
– для кожної часткової послідовності у промені;
– обчислюємо ймовірності наступних токенів з використанням моделі;
– вибираємо топ *N* найімовірніших наступних токенів на основі обчислених ймовірностей;
– розширюємо кожну часткову послідовність на кожен з вибраних токенів, утворюючи нові часткові послідовності.
3. Обрізка променя:
– обмежуємо розмір променя за допомогою певного критерію наприклад, за допомогою топ N найкращих часткових послідовностей з найвищими оцінками.
4. Завершення генерації:
– продовжуємо генерацію кроків 2-3 до досягнення кінцевого токену або до досягнення максимальної довжини послідовності.
5. Вибір найкращої послідовності:
– після завершення генерації, вибираємо найкращу згенеровану послідовність (з найвищою оцінкою) як результат генерації.
Особливості методу генерації з пошуком променя:
– генерація відбувається ітеративно, з кожним кроком додавання нових токенів і розширенням часткових послідовностей;
– вибір наступних токенів залежить від їх ймовірності, що обчислюється моделлю;
– промінь дозволяє обмежити кількість розглянутих варіантів, що полегшує обробку при великих розмірах словника;
– завершення генерації відбувається після досягнення кінцевого токену або максимальної довжини, що дозволяє керувати довжиною результатів.

	Початкове	Кінцеве	Збільшення
R1 (BART)	12.34	32.45	20.11
R2 (BART)	1.20	14.50	13.30
RL (BART)	10.56	28.60	18.04
R1 (ProphetNet)	13.20	36.80	23.60
R2 (ProphetNet)	1.80	17.20	15.40
RL (ProphetNet)	11.05	32.10	21.05

Результати експериментів з моделями машинного навчання BART і ProphetNet

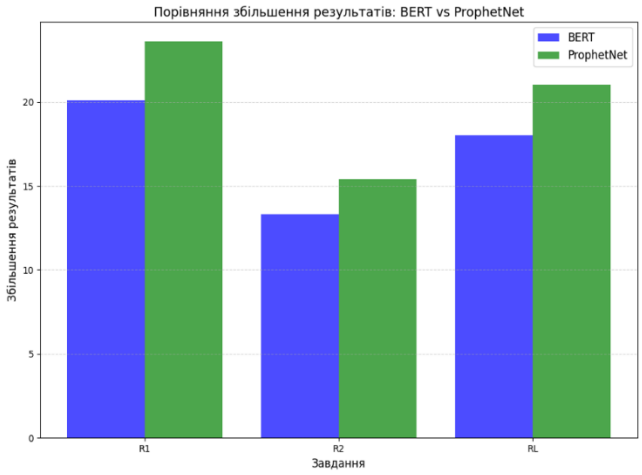
Збільшення результатів відображає, наскільки кожна модель покращила свої показники в кожному з завдань





Результати екперимента для BART

Результати екперимента для ProphetNet

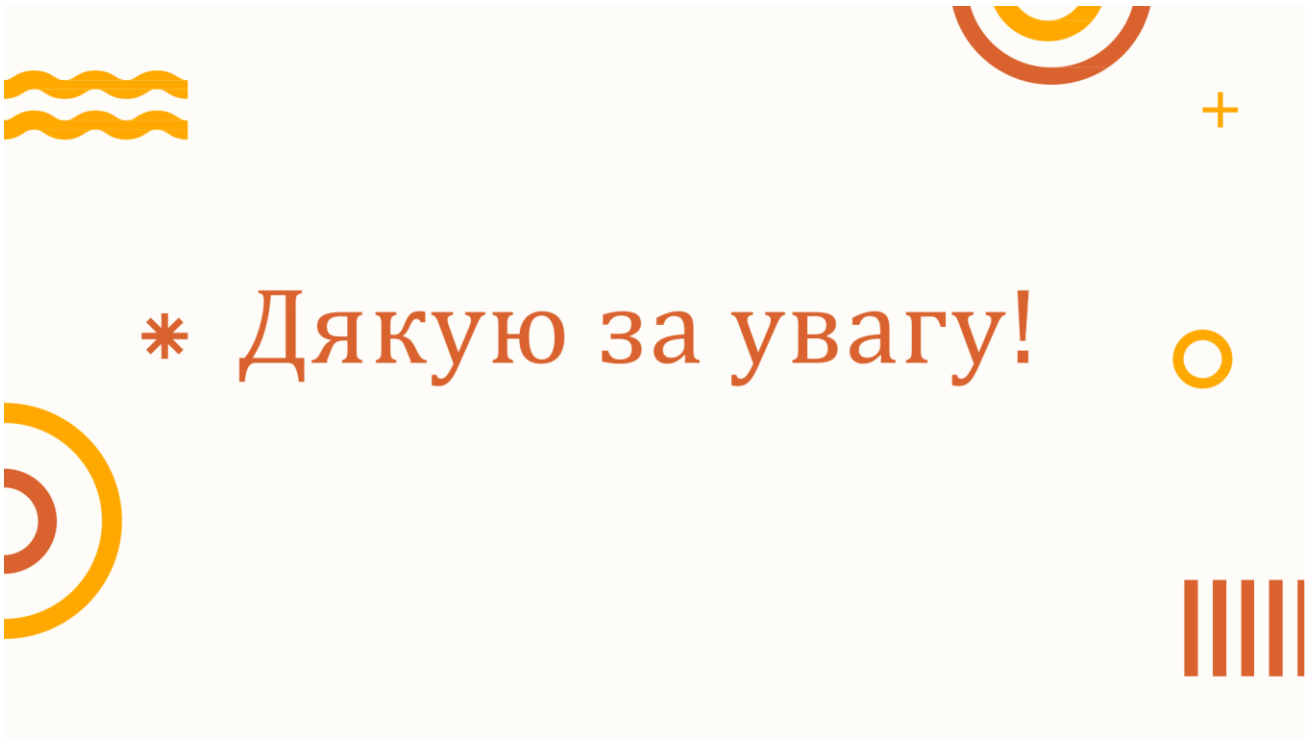


Результати екперимента для BART та ProphetNet

Висновки

- проведено детальний аналіз існуючих методів обробки текстів, що дозволило визначити оптимальні підходи до виділення важливих ознак для генерації короткого змісту;
- розроблено метод генерації короткого змісту тексту з використанням нейронної мережі ProphetNet;
– здійснено оптимізацію параметрів нейронної мережі для покращення її якості та точності в завданнях генерації короткого змісту тексту;
- проведено експериментальне тестування розробленого методу на наборі текстових даних для оцінки його якості.

Дослідження підтвердило, що нейронні мережі на основі трансформерів, зокрема модель ProphetNet, є ефективними інструментами для автоматичної сумаризації тексту, демонструючи перевагу в контекстуальному розумінні та узагальненні текстової інформації. Отримані результати можуть бути використані для подальшого розвитку та вдосконалення систем автоматичної сумаризації.


Дякую за увагу!

UNICHECK
by Turnitin

Ім'я користувача:
Кафедра КН

ID перевірки:
1016365390

Дата перевірки:
16.06.2024 16:58:53 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
16.06.2024 17:00:36 EEST

ID користувача:
100005671

Назва документа: **КН-20-1_Бондар_ЗАПИСКА**

Кількість сторінок: **73** **Кількість слів:** **15157** **Кількість символів:** **118496** **Розмір файлу:** **1.11 MB** **ID файлу:** **1016171412**

9.01%
Схожість

Найбільша схожість: **3.03%** **з джерелом з Бібліотеки** (ID **файлу:** **1016168773**)

7.65% Джерела з Інтернету	872	Сторінка 75
4.06% Джерела з Бібліотеки	95	Сторінка 84

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Sun Jun 16 16:08:41 EEST 2024, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 10%**

	Документ		Сумарний збіг по Базі Даних	
ID: 130796 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод генерації короткого змісту тексту з використанням нейронної мережі Додано в БД: 2024-06-16 Автора: Олександр БОНДАР Керівники: Едуард МАНЗЮК Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	94507	1445	2666 (3%)	38 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: *Метод генерації короткого змісту тексту з використанням нейронної мережі*

Автор: *студент групи КН-20-1 Олександр БОНДАР*

Спеціальність: *122 – Комп'ютерні науки*

Освітня програма: *освітньо-професійна*

Науковий керівник: *д.т.н., професор кафедри Манзюк Е.А.*

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	***відповідає***
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі Олександра Бондара, не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти програмного коду, що не мають авторства і містять поширені конструкції; серед запозичень знаходяться загальновідомі терміни, скорочення.

Обсяг запозичень, визначений системами виявлення збігів/ ідентичності/схожості, складає:

- за системою Anti-Plagiarism: 2%;

- за системою Unicheck: 9.01%.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості є допустимим.

Керівник роботи — *Едуард МАНЗЮК*

Гарант ОП — *Олександр МАЗУРЕЦЬ*

Завідувач кафедри КН — *Олександр БАРМАК*

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА

на кваліфікаційну роботу бакалавра

студента *гр. КН-20-1 Олександра БОНДАРА*

за темою *Метод генерації короткого змісту тексту з використанням нейронної мережі*

1. Актуальність теми

Генерація короткого змісту тексту є важливим завданням в області обробки природної мови, особливо в сучасному інформаційному суспільстві, де обсяги текстових даних стрімко зростають. Використання нейронних мереж, зокрема моделей на основі трансформерів, дозволяє значно покращити якість автоматичної сумаризації тексту завдяки їх здатності враховувати контекст та семантику. Це сприяє створенню більш інтелектуальних і адаптивних інформаційних систем, здатних ефективно працювати з великими обсягами текстової інформації. Таким чином, розроблення методів генерації короткого змісту тексту з використанням нейронних мереж є актуальним і важливим напрямком досліджень.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

Відповідно до встановлених стандартів, об'єкти дослідження та сфера діяльності охоплюють математичні, інформаційні та симуляційні моделі реальних явищ, об'єктів, систем і процесів. Також включаються методи та технології для збору, зберігання, обробки, передачі та використання інформації. Основною метою цієї роботи є розробка методу генерації короткого змісту тексту. Для досягнення цієї мети застосовуються математичні моделі, методи та алгоритми, які вирішують як теоретичні, так і практичні завдання, пов'язані з розробкою методів машинного навчання. Результати цієї бакалаврської роботи відповідають стандартам спеціальності 122 – Комп'ютерні науки.

3. Професійні та особистісні якості бакалавра

Під час виконання кваліфікаційної роботи бакалавра Олександр Бондарпродемонстрував ґрунтовні знання та розвинені практичні навички. Він своєчасно впорався з усіма поставленими завданнями, що свідчить про його відповідальне ставлення до роботи. У процесі написання пояснювальної записки та розробки методу дослідження

виявив високий рівень професійної підготовки та успішність у навчанні. Йому вдалося успішно застосувати та вдосконалити свої компетенції в галузі комп'ютерних наук.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Одержані в роботі результатиє наслідком особистої діяльності студента, який самостійно виконував усі поставлені задачі.

5. Ступінь оволодіння методами дослідження

У процесі виконання кваліфікаційної роботи продемонстрував належний рівень компетентностей та володіння необхідними методами, техніками та технологіями у сфері комп'ютерних наук.

6. Повнота та якість розкриття теми роботи

Тема роботи глибоко обґрунтована та всебічно розкрита. Проведено аналіз наявних досліджень за обраною темою. Поставлені завдання успішно вирішені, а також розроблено метод генерації короткого змісту тексту.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Структура роботи та послідовність викладення логічні та відповідають поставленій меті. Викладення матеріалу послідовне, аргументоване, літературно грамотне.

8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Розроблений у роботі метод може бути використаний в системах формування заголовків та сумаризації текстової інформації.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

Враховуючи належний рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Керівник _____ *д.т.н., професоркаф.КН Едуард МАНЗЮК*

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента *гр. КН-20-1 Олександра БОНДАРА*

за темою: *Метод генерації короткого змісту тексту з використанням нейронної мережі*

1. Актуальність обраної теми

Генерація короткого змісту тексту є важливим завданням в сучасному інформаційному суспільстві, де постійно зростає кількість текстових даних, які необхідно ефективно обробляти та узагальнювати. Можливість генерувати короткі, але інформативні описи великих обсягів текстової інформації має широке практичне застосування, наприклад, для автоматичного створення заголовків, анотацій наукових статей чи резюме новин. Тому розробка ефективних методів генерації короткого змісту тексту з використанням нейронних мереж є актуальним і важливим напрямком досліджень.

2. Повнота розкриття мети та завдань роботи

Для досягнення цієї мети було здійснено теоретичний огляд існуючих методів генерації короткого змісту на основі нейронних мереж, реалізовано зазначені моделі, проведено їх експериментальне порівняння за метриками якості, проаналізовано переваги і недоліки кожної моделі, та сформульовано висновки щодо їх ефективності в задачах автоматичної сумаризації текстів.

3. Зміст кожного розділу роботи

Записка кваліфікаційної роботи бакалавра містить три розділи. У першому розділі проведено аналіз предметної області, досліджено відомі роботи та визначено актуальність теми. У другому розділі представлено метод генерації короткого змісту тексту. Третій розділ присвячено експериментальній перевірці його ефективності.

4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблений метод дозволяєформувати короткий та влучний заголовок, що суттєво полегшує автоматизацію генерації коротких назв.

5. Якість оформлення кваліфікаційної роботи бакалавра

Записка відповідає всім вимогам і правилам оформлення. Викладення матеріалу є логічним і послідовним.

6. Недоліки кваліфікаційної роботи бакалавра

Рекомендовано вдосконалити систему шляхом аналізу типових похибок .

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

*Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «**відмінно**».*

Рецензент *професор кафедри КІІС, д.т.н., професор Лисенко Сергій Миколайович*