Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

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

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів сумаризації українських текстів

(тема)

Виконав:
студент 2 курсу, групи СШІм-22-1
Кардаш Д.М.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)
Керівник доц. Турута О.П.
(посада, прізвище, ініціали)

Допускається до захисту

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

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

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

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

Освітня програма Системи штучного інтелекту
(повна назва)

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

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Кардашу Дмитру Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів сумаризації українських текстів

затверджена наказом університету від 1 квітня 20 24 р. № 260Ст
2. Термін подання студентом роботи до екзаменаційної комісії 11 червня 20 24 р.
3. Вихідні дані до роботи Теоретичне дослідження рішення і опис технологій, робота з потоком даних, математична основа рішення, задача в предметній галузі, практична частина роботи з вибраним алгоритмом, вибір алгоритму машинного навчання, обробка текстових даних, попередня обробка та візуалізація даних для роботи, застосовані технології, програмна реалізація застосунку, процес роботи з текстами та вибір ЄВМ, проектування та тестування

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	01.04.2024	виконано
2	Аналіз предметної галузі та постановка задачі	04.04.2024 – 15.04.2024	виконано
3	Аналіз методів вирішення поставленої задачі	15.04.2024 – 20.04.2024	виконано
4	Вибір методу для вирішення задачі	20.04.2024 – 25.04.2024	виконано
5	Розробка базового прототипу	01.05.2024 – 15.05.2024	виконано
6	Підготовка пояснювальної записки	16.05.2024 – 05.06.2024	виконано
7	Захист перед ЕК	08.06.2024	виконано

Дата видачі завдання 1 квітня 20 24 р.

Студент _____
(підпис)

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

РЕФЕРАТ

Пояснювальна записка: 89 с., 26 рис., 1 табл., 1 дод., 21 джерело.

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

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

Предмет дослідження – практична реалізація одного з алгоритмів штучного інтелекту для розв'язання задачі суммаризації тексту для обраної предметної галузі української мови.

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

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

ABSTRACT

Master`s thesis contains: 89 pp., 26 fig., 1 tabl., 1 ann., 21 references.

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

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

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

The purpose of the study is to apply artificial intelligence methods to solve the problem of summarizing a news text.

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

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ММ – мовне моделювання;

НКУМ – національний корпус української мови;

ПМ – природна мова;

ШІ – штучний інтелект;

RNN – Recurrent Neural Network – рекурентна нейронна мережа;

DL – Deep Learning – глибинне навчання;

DNN – Deep Neural Network – глибока нейронна мережа;

FL – Feature Learning – навчання ознак;

NER – Named entity recognition – розпізнавання іменованих об'єктів;

IE – Information Extraction – добування інформації.

ВСТУП

Поява мережі Інтернет і бурхливе зростання доступної текстової інформації значно прискорило розвиток наукової галузі, яка існує вже багато десятків років і відома як автоматичне опрацювання текстів (Natural Language Processing) і комп'ютерна лінгвістика (Computational Linguistics). У рамках цієї галузі запропоновано багато перспективних ідей щодо автоматичного опрацювання текстів природною мовою (ПМ), які було втілено в багатьох прикладних системах, у тому числі комерційних. Сфера застосувань комп'ютерної лінгвістики постійно розширюється, з'являються дедалі нові задачі, які успішно розв'язуються, зокрема із залученням результатів суміжних наукових галузей. Про наукові досягнення галузі можна отримати уявлення з інтернет-сайту ACL (Association of Computational Linguistics) [1] – міжнародної Асоціації з Комп'ютерної Лінгвістики, на якому агрегуються роботи численних наукових конференцій у цій галузі.

Комп'ютерна лінгвістика (КЛ) – міждисциплінарна галузь, що виникла на стику таких наук, як лінгвістика, математика, інформатика (Computer Science), штучний інтелект (Artificial Intelligence), у своєму розвитку вона дотепер убирає й застосовує (за потреби адаптуючи) розроблені в цих науках методи та інструменти.

Витоки КЛ сягають досліджень відомого американського лінгвіста Н. Хомського з формалізації структури природної мови, перших експериментів з машинного перекладу, виконаних програмістами та математиками, а також розроблених у галузі штучного інтелекту перших програм розуміння природної мови.

У цій роботі ми хочемо дослідити, розширити, розробити, оцінити та порівняти набір мовних моделей для української мови. Основна мета полягає в тому, щоб запропонувати оціночний корпус і встановити низку базових орієнтирів. Будуть вирішені наступні задачі:

– надати огляд існуючих підходів до мовного моделювання інших мов;

– скласти корпус даних, достатньо великий для навчання нейромовідтворювальних мовних моделей для української мови, та здійснити відповідну попередню обробку зібраних даних;

– провести експерименти над різними мовними одиницями, які потенційно можуть краще моделювати послідовність послідовних послідовностей мовних одиниць, що можуть потенційно краще моделювати послідовність послідовних послідовностей мовних одиниць;

– провести експерименти з різними лінгвістичними одиницями, які потенційно можуть краще моделювати послідовну інформацію в українських даних;

– дослідити, відтворити та розширити набір основних підходів, які вважаються ефективними для інших мов;

– оцінити та порівняти мовні моделі, навчені для української мови. Запропонувати оціночний корпус та встановити низку базових показників.

Оскільки в КЛ об'єктом опрацювання виступають тексти природної мови, її розвиток неможливий без базових знань у галузі загальної лінгвістики (мовознавства). Лінгвістика вивчає загальні закони природної мови – її структуру та функціонування, і включає такі галузі:

– фонологія – вивчає звуки мови та правила їхнього сполучення під час формування мови;

– морфологія – займається внутрішньою структурою та зовнішньою формою слів мови, включно з частинами мови та їхніми категоріями;

– синтаксис – вивчає структуру речень, правила сполучуваності та порядку слідування слів у реченні, а також загальні його властивості як одиниці мови;

– семантика і прагматика – а тісно пов'язані царини: семантика опікується смислом слів, речень та інших мовленнєвих одиниць, а прагматика особливостями вираження цього смислу у зв'язку з конкретними цілями спілкування;

– лексикографія описує лексикон конкретної ЕМ – її окремі слова, їхні граматичні та семантичні властивості, а також методи створення словників.

Найтісніше комп'ютерна лінгвістика пов'язана з галуззю штучного інтелекту (ШІ), у рамках якої розробляють програмні моделі окремих інтелектуальних функцій. Незважаючи на очевидний перетин досліджень у царині комп'ютерної лінгвістики та ШІ (оскільки володіння мовою належить до інтелектуальних функцій), ШІ не поглинає всю КЛ, оскільки вона має свій теоретичний базис і методологію. Спільним для зазначених наук є комп'ютерне моделювання як основний спосіб і підсумкова мета досліджень, евристичний характер багатьох застосовуваних методів.

Дещо спрощено, завдання комп'ютерної лінгвістики може бути сформульоване як розробка методів і засобів побудови лінгвістичних процесорів для різних прикладних завдань з автоматичного опрацювання текстів на ЕЯ. Розроблення лінгвістичного процесора для деякої прикладної задачі передбачає формальний опис лінгвістичних властивостей оброблюваного тексту (хоча б найпростіший), який можна розглядати як модель тексту (або модель мови)

У роботі наведено аналітичний підсумок того, що було зроблено науковою спільнотою для англійської мови з точки зору мовного моделювання. Розділи містять огляд відповідних публікацій та базову теорію, яка лежить в основі роботи, представленої далі. У розділах описано методологію, за допомогою якої ми вирішували питання мовного моделювання для української мови.

1 ОГЛЯД ЗАДАЧ У МЕЖАХ МЕТОДІВ СУМАРИЗАЦІЇ ТЕКСТІВ

1.1 Додатки комп'ютерної лінгвістики

Сфера застосувань КЛ постійно розширюється, тому охарактеризуємо тут найвідоміші прикладні задачі, розв'язувані її інструментами.

Машинний переклад (Machine Translation) – найперший додаток КЛ, разом з яким виникла і розвивалася сама ця галузь. Перші програми перекладу були побудовані в середині минулого століття і були засновані на найпростішій стратегії послівного перекладу. Однак досить швидко було усвідомлено, що машинний переклад вимагає набагато повнішої лінгвістичної моделі. Таку модель було розроблено у системі ЕТАП, а також у кількох інших системах, що виконують переклад наукових текстів.

Нині існує цілий спектр комп'ютерних систем машинного перекладу (різної якості), від великих інтернаціональних дослідницьких проєктів до комерційних автоматичних перекладачів. Істотний інтерес становлять проєкти багатомовного перекладу, з використанням проміжної мови, якою кодується зміст фраз, що перекладаються. Сучасний напрям – статистична трансляція, що спирається на статистику перекладних пар слів і словосполучень. Незважаючи на багато десятиліть досліджень цього завдання, якість машинного перекладу ще далека до досконалості. Істотний прорив у цій царині пов'язують із використання машинного навчання та нейронних мереж (які виникли й досліджуються в рамках ШІ).

Ще один доволі старий застосунок комп'ютерної лінгвістики – це інформаційний пошук (Information Retrieval) і пов'язані з ним завдання індексування, реферування, класифікації та рубрикування документів.

Повнотекстовий пошук документів у великих базах текстових документів передбачає індексування текстів, що вимагає їхнього найпростішого лінгвістичного попереднього оброблення, і створення спеціальних індексних структур. Відомі кілька моделей інформаційного

пошуку, найвідомішою і найпоширенішою є векторна модель, за якої інформаційний запит подають як набір слів, а відповідні (релевантні) документи визначають на основі схожості запиту і вектора слів документа. Сучасні інтернет-пошуковики реалізують цю модель, виконуючи індексування текстів за вживаними в них словами і використовуючи для видачі релевантних документів вельми витончені процедури ранжирування. Актуальний напрям досліджень у сфері інформаційного пошуку – багатомовний пошук за документами.

Реферування тексту (Summarization) – скорочення його обсягу й одержання короткого викладу його змісту – реферату, що робить швидшим пошук у колекціях документів. Реферат може складатися також для кількох близьких за темою документів (наприклад, за кластером новинних документів). Основним методом автоматичного реферування досі є відбір найбільш значущих речень реферованого тексту на основі статистики слів і словосполучень, а також структурних і лінгвістичних особливостей текстів.

Близьке до реферування завдання – анотування тексту документа, тобто складання його анотації. У найпростішій формі анотація являє собою перелік основних (ключових) тем тексту, для виокремлення яких використовують статистичні та лінгвістичні критерії.

Під час опрацювання великих колекцій документів актуальними є завдання класифікації (Categorization) і кластеризації текстів (Text Clustering). Класифікація означає віднесення кожного документа до певного класу із заздалегідь відомими параметрами, а кластеризація – розбиття множини документів на кластери, тобто підмножини тематично близьких документів. Для розв'язання цих завдань застосовують методи машинного навчання, у зв'язку з чим ці прикладні завдання часто відносять до напряму Text Mining, який розглядають як частину наукової галузі Data Mining (інтелектуальний аналіз даних). Завдання класифікації набуває дедалі більшого поширення, його розв'язують, наприклад, під час розпізнавання спаму, класифікації SMS-повідомлень тощо.

Дуже близьким до класифікації є завдання рубрикування тексту (Text Classification) – віднесення тексту до однієї із заздалегідь відомих тематичних рубрик (зазвичай рубрики утворюють ієрархічне дерево тематик).

Відносно нове завдання, пов'язане з інформаційним пошуком – формування відповідей на запитання (Question Answering). Завдання розв'язують шляхом визначення типу запитання, пошуком текстів, що потенційно містять відповідь на це запитання (водночас зазвичай застосовують пошукові машини), і потім витяганням відповіді з виданих текстів.

Актуальне прикладне завдання, яке часто відносять до напряму Text Mining – це добування інформації з текстів (Information Extraction), що потрібне під час розв'язання завдань економічної та виробничої аналітики. Під час розв'язання цього завдання здійснюється виокремлення в тесті ПМ певних об'єктів – іменованих сутностей (імен персоналій, географічних назв, назв фірм тощо), їхніх стосунків і пов'язаних із ними подій. Як правило, це реалізується на основі часткового синтаксичного аналізу тексту, що дає змогу виконувати опрацювання великих масивів текстів, зокрема, потоків новин від інформаційних агентств. Виділені дані тим чи іншим чином структуруються або візуалізуються.

До напряму Text Mining належать і дві інші близькі задачі – виокремлення думок (Opinion Mining) та аналіз тональності текстів (Sentiment Analysis), що привертають увагу дедалі більшої кількості дослідників через свою актуальність. У першому завданні відбувається пошук (у блогах, форумах, інтернет–магазинах тощо) думок користувачів про товари та інші об'єкти, а також проводиться аналіз цих думок. Друга задача близька до класичної задачі контент-аналізу текстів масової комунікації, в ній оцінюється загальна тональність висловлювань і тексту загалом.

Ще одне прикладне завдання, яке виникло понад 50 років тому, і розвиток якого стимулювала поява мережі Інтернет – це підтримка діалогу ПМ. Раніше це завдання найчастіше розв'язували в рамках будь-якої інформаційної системи, зокрема, для опрацювання запитів на ПМ до спеціалізованої бази даних – у цьому разі мова запитів доволі обмежена (лексично і граматично), що дає змогу використовувати спрощені методи аналізу запитань, а відповіді будувати за шаблонами. Наразі дедалі більшого поширення в Інтернеті набувають чат-боти, що підтримують бесіду з людиною на певну тему і є спадкоємцями відомої системи ELIZA (розробленої в галузі ШІ в 70 рр.). Очевидний успіх цього напряму в тому, що з'явилися програми (наприклад, програма-співрозмовник «Євген Гусман»), які проходять відомий тест Тюрінга.

Зовсім інший прикладний напрям, який розвивається хоча й повільно, але стійко – це автоматизація підготовки та редагування текстів на ПМ. Одними з перших досягнень у цьому напрямі були програми автоматичного визначення переносів слів і програми орфографічної перевірки тексту (спелери, або автокоректори). Перевірка орфографії вже давно реалізована в комерційних системах, виявляються також досить частотні синтаксичні помилки (наприклад, помилки узгодження слів). Водночас в автокоректорах поки що не реалізовано розпізнавання більш складних помилок, зокрема, неправильного вживання прийменників та лексичних помилок, які виникають унаслідок друкарських помилок (правки замість довідки) або неправильного використання схожих слів (наприклад, ваговий замість вагомий). У сучасних дослідженнях КЛ розробляються методи автоматизованого виявлення та виправлення подібних помилок на основі статистики зустрічальності слів і словосполучень [21].

Ще одним прикладним завданням є навчання природної мови, у рамках цього напряму створюються комп'ютерні системи, що підтримують вивчення окремих аспектів (морфології, лексики, синтаксису) мови – англійської, української тощо. (подібні системи можна знайти в Інтернеті).

Розробляють також багатофункціональні комп'ютерні словники, які не мають текстових аналогів і орієнтовані на широке коло користувачів, наприклад, словник сполучуваності слів української мови Кросслексика, який додатково надає довідки щодо синонімів, антонімів та інших смислових зв'язків слів.

Наступний прикладний напрямок, який варто згадати, – це автоматична генерація текстів на ПМ [2]. У принципі, цю задачу можна вважати під задачею вже розглянутої вище задачі машинного перекладу, проте в рамках напряму є низка специфічних завдань. Таким завданням є багатомовна генерація, тобто автоматична побудова одразу кількома мовами спеціальних документів – патентних формул, інструкцій з експлуатації технічних виробів або програмних систем, виходячи з їхньої формальної специфікації.

Напрямком, що активно розвивається, є розпізнавання і синтез звучної мови. Помилки розпізнавання, що неминуче виникають, виправляються автоматичними методами на основі словників і морфологічних моделей, також застосовується машинне навчання.

1.2 Складнощі моделювання природної мови

Складність моделювання в КЛ пов'язана з тим, що ПМ – велика відкрита багаторівнева система знаків, що виникла для обміну інформацією в процесі практичної діяльності людини, і постійно змінюється у зв'язку з цією діяльністю.

Текст ЕМ складено з окремих одиниць (знаків), і можливе кілька способів розбиття (членування) тексту на одиниці, що належать до різних рівнів.

Загальновизнано існування таких рівнів:

– рівень речень (висловлювань) – синтаксичний рівень;

– рівень слів (словоформ – слів у певній граматичній формі, наприклад, ручка, дружба) – морфологічний рівень;

– рівень фонем (окремих звуків, за допомогою яких формуються та розрізняються слова) – фонологічний рівень.

Фонологічний рівень виокремлюється для усного мовлення, а для письмових текстів у мовах з алфавітним способом запису (зокрема, в європейських мовах) він відповідає рівню символів (фонеми приблизно відповідають буквам алфавіту).

Рівні, по суті, є підсистемами загальної системи ПМ (взаємопов'язані, але достатньою мірою автономні), і в них самих можуть бути виділені підсистеми. Так, морфологічний рівень включає також підрівень морфем. Морфема – це мінімальна значуща частина слова (корінь, префікс, суфікс, закінчення, постфікс).

Питання про кількість рівнів та їхній перелік у лінгвістиці досі залишається відкритим. Як окремий може бути виділено лексичний рівень – рівень лексем. Точніше, лексема – семантичний інваріант усіх словоформ. У тексті зустрічаються словоформи (лексеми в певній формі), а в словнику ПМ – лексеми, точніше, у словнику записується канонічна словоформа лексеми, яка називається також лемою (наприклад, для іменників це форма називного відмінка однини: лист).

У межах синтаксичного рівня можна виокремити підрівень словосполучень – синтаксично пов'язаних груп слів (бачив ліс, синя куля), і надрівень складного синтаксичного цілого, якому приблизно відповідає абзац тексту. Складне синтаксичне ціле, або надфразова єдність – це послідовність речень (висловлювань), об'єднаних смислом і лексико-граматичними засобами. До таких засобів належать передусім лексичні повтори й анафоричні посилання – посилання на попередні слова тексту, що реалізуються за допомогою займенників і займенникових слів (вони, цей, там же тощо).

Ієрархія рівнів проявляється в тому, що одиниці вищого рівня розкладаються на одиниці нижчого (наприклад, словоформи на морфи); вищий рівень великою мірою зумовлює організацію нижчого рівня – так, синтаксична структура речення значною мірою визначає, які мають бути обрані словоформи. Можна також говорити ще про один рівень – рівень дискурсу, під яким розуміється зв'язний текст у його комунікативній спрямованості.

Під дискурсом розуміють послідовність взаємопов'язаних одне з одним речень тексту, що має певну смислову цілісність, завдяки чому він виконує певне прагматичне завдання. У багатьох типах зв'язних текстів проявляється традиційна схематична (дискурсивна) структура, що організовує їхній загальний зміст, наприклад, певну структуру мають описи складних технічних систем, патентні формули, наукові статті, ділові листи тощо. Особливим є питання про рівень семантики.

Крім багаторівневості системи ПМ складність її моделювання пов'язана зі змінами, що постійно відбуваються в ній (що цілком відчутно після одного-двох десятиліть). Зміни стосуються не тільки словникового запасу мови (нові слова і нові смисли старих), а й синтаксису, морфології та фонетики. Як наслідок, принципово неможливо одного разу розробити формальну модель конкретної ПМ і побудувати відповідний лінгвістичний процесор. Потрібне постійне поповнення знань про мову на всіх її рівнях і корекція наявних моделей. Одним із наслідків довгого історичного розвитку ЕМ є нестандартна сполучуваність (синтактика) одиниць на кожному рівні мови. На відміну від штучних формальних мов (мов логіки, мов програмування), у яких сполучуваність знаків диктується їхньою семантикою та може бути зафіксована синтаксично (граматично), у природних мовах поєднання слів у реченнях лише частково може бути описане законами граматики.

1.3 Загальні етапи та модулі опрацювання текстів

Складність формального опису ПМ та її опрацювання веде до розбиття цього процесу на окремі етапи, що відповідають рівням мови. Більшість сучасних лінгвістичних процесорів належать до модульного типу, в якому кожному рівню/етапу аналізу або синтезу тексту відповідає окремий модуль процесора. У разі аналізу тексту окремі модулі ЛП виконують:

– графематичний аналіз (сегментація), тобто виділення в тексті речень і словоформ, точніше токенів (оскільки у тексті можуть бути не тільки слова) – перехід від символів до слів;

– морфологічний аналіз – перехід від словоформ до їхніх лемм (словникових форм лексем) або основ (ядерних частин слова, за вирахуванням словозмінювальних морфем);

– синтаксичний аналіз – виявлення синтаксичних зв'язків слів та граматичної структури речень;

– семантичний та прагматичний аналіз, під час якого визначають смисл фраз та відповідну реакцію системи, у межах якої працює ЛП.

Таким чином, лінгвістичний процесор можна розглядати як багатоетапний перетворювач, який перекладає у разі аналізу тексту кожне його речення у внутрішнє представлення його смислу і навпаки у разі синтезу.

Можливі різні схеми об'єднання і взаємодії модулів розглянутих етапів, проте окремі рівні – морфологія, синтаксис і семантика – зазвичай опрацьовуються різними механізмами. Під час розв'язання деяких прикладних завдань можна обійтися без представлення в процесорі всіх етапів/рівнів (наприклад, у ранніх експериментальних програмах КЛ опрацьовувані тексти належали до дуже вузьких проблемних сфер з

обмеженим набором слів, тож не був потрібен морфологічний і синтаксичний аналіз).

Модулі морфологічного аналізу словоформ розрізняються в основному за такими параметрами:

– результатом роботи – лемма або основа з набором морфологічних характеристик (рід, число, відмінок, вид, особа тощо) заданої словоформи;

– методом аналізу – з опорою на словник словоформ мови або на словник основ, або ж безсловесний метод;

– можливістю опрацювання словоформи лексеми, що не внесена до словника.

Під час морфологічного синтезу вихідними даними є лексема і конкретні морфологічні характеристики запитуваної словоформи цієї лексеми, можливий і запит на синтез усіх форм заданої лексеми (так званої парадигми слова). Результат як морфологічного аналізу, так і синтезу в загальному випадку неоднозначний.

1.4 Підходи до побудови модулів і систем КЛ

Нині для створення модулів лінгвістичних процесорів застосовують два головні підходи: заснований на правилах (rulebased), або інженерний, і заснований на машинному навчанні (machine learning).

Історично першим є підхід на правилах, який полягає в описі необхідної лінгвістичної інформації у вигляді формальних правил. У ранніх системах правила були вбудовані в програмний код, зараз же для запису правил використовують або вже готову формальну мову, або таку мову спеціально створюють для розроблюваного застосунку. Правила створюють лінгвісти або фахівці з проблемної області текстів, що обробляються.

У рамках підходу, заснованого на машинному навчанні, джерелом лінгвістичної інформації виступають не правила, а відібрані тексти проблемної області. Серед методів, що застосовуються в рамках підходу, виділяють методи навчання з учителем (supervised), методи навчання без учителя (unsupervised), методи часткового навчання з учителем (bootstrapping).

Найчастіше застосовується навчання з учителем, за якого відбувається побудова математичної та програмної моделі – машинного класифікатора, що вміє розпізнавати різні класи одиниць тексту (слів, словосполучень та інших конструкцій) або самих текстів. Побудова класифікатора відбувається на спеціально розміченому текстовому корпусі (навчальній вибірці), у якому одиницям, що розпізнаються (або самим текстам), приписують мітки, що кодують важливі ознаки одиниць/текстів, які розпізнають.

Сучасна тенденція – модульні, багатокомпонентні системи автоматичного опрацювання текстів (multi-component, pipelined systems), причому різні модулі можуть бути створені в рамках різних підходів, наприклад, модуль графематичного аналізу – на основі машинного навчання, а морфологічного – на основі правил.

Машинне навчання доволі часто застосовують для опрацювання колекцій текстових документів, з використанням ознакової моделі тексту, за якої ознаки визначено для кожного документа окремо. Ознаками можуть виступати різноманітні інформаційні характеристики тексту: як лінгвістичні, так і статистичні та структурні: наприклад, частота певних слів (або їхніх категорій) у документі, частота використання спецзнаків, співвідношення частин мови слів, наявність певних синтаксичних конструкцій або розділів тексту, дата створення тощо.

Різновидами ознакової моделі є модель BOW (bag of words – мішок слів), у якій текст характеризується набором своїх значущих слів (зазвичай це всі знаменні слова, точніше, їхні леми), а також векторна модель тексту,

у якій зазначений набір упорядковано. Векторну модель застосовують, наприклад, в інформаційному пошуку, водночас як ознаки частіше беруть не слова, а складніші характеристики, як-от показник TF-IDF для слів.

Окремо стоїть статистична мовна модель (Language Model), що характеризує мову загалом, а не окремий текст [12]. Класичну мовну модель будують за представницьким масивом текстів конкретної ПМ (наприклад, англійської) шляхом підрахунку частот N-грам слів (тобто слів, що стоять поруч). Найчастіше розглядають біграми ($N = 2$) і триграми ($N = 3$). Модель покликана давати відповідь на запитання, наскільки ймовірною є поява заданого слова, якщо безпосередньо перед ним зустрічалися певні слова. Ймовірності розраховуються на основі зібраної статистики. Таку модель застосовують, наприклад, для розв'язання лексичної неоднозначності. Різновиди моделі: N-грами частин мови слів тексту або N-грами літер тексту (можливі й інші моделі) застосовують для розв'язання морфологічної омонімії або для виявлення помилок у тексті відповідно.

1.5 Метрики якості сумаризації

Існують три групи методів оцінювання якості автоматичної сумаризації: експертна оцінка, автоматичні методи та напівавтоматичні методи.

1.5.1 Експертна оцінка

Експертна оцінка автоматично згенерованого реферату полягає в тому, що група експертів за заданими критеріями оцінює якість отриманого тексту. У тому числі можливе порівняння автоматично згенерованого тексту з текстом, написаним експертом. Не виключені такі випадки, що автоматично згенерований реферат може бути в чомусь кращим за текст експерта [2], і ручне оцінювання дає змогу виявити такі прецеденти.

1.5.2 ROUGE (автоматичні методи)

Серед методів автоматичного оцінювання якості сумаризації застосовують різні різновиди метрики ROUGE [16], [3], один із найпопулярніших різновидів – ROUGE-N. ROUGE-N – це частка n-грам з еталонного резюме–прикладу, що опинилася в автоматично згенерованому резюме. ROUGE-Nrecall представляє частку збігаючихся n-грам для згенерованого реферату і реферату-прикладу в загальному числі n-грам прикладу:

$$recall = \frac{count_{match}(gram_n)}{count_{ref.summ.}(gram_n)}. \quad (1.1)$$

ROUGE-N precision представляє частку збігаючихся n-грам для згенерованого реферату та реферату-прикладу в загальному числі n-грам згенерованого реферату:

$$precision = \frac{count_{match}(gram_n)}{count_{gen.summ.}(gram_n)}. \quad (1.2)$$

ROUGE-NF1 метрика – це середнє гармонійне між попередніми двома метриками:

$$F1_{score} = \frac{precision * recall}{precision + recall}. \quad (1.3)$$

Також існують такі метрики як: ROUGE-L – найдовша загальна послідовність, ROUGE-W – зважена найдовша загальна послідовність і ROUGE-S – статистика збігів за скіп–грамами.

1.5.3 PYRAMID (напівавтоматичні методи)

Метод оцінювання якості сумаризації PYRAMID [4] ґрунтується на гіпотезі про те, що існує еталонний стандарт реферату тексту. Хоча люди схильні резюмувати тексти по-різному, виокремлюючи різну інформацію з вихідного тексту, припускають, що можна зібрати еталонний реферат із безлічі рефератів, складених людьми, використовуючи інформацію про частоту появи тієї чи іншої інформації. За частотою можна судити про значущість інформації та оцінити якість сумаризації.

1.5.4 Автоматичні методи оцінювання читабельності

Зазначені вище автоматичні та напівавтоматичні методи дають змогу зрозуміти, зокрема, яку кількість важливої інформації було витягнуто, але є методи, які також дають змогу оцінити читабельність тексту. Існують такі методи автоматичної оцінки читабельності тексту:

– індекс читабельності Флеша–Кінкейда [5]:

$$FKGL = 0.39 * \left(\frac{\text{число слів}}{\text{число речень}}\right) + 11.8 * \left(\frac{\text{число слогів}}{\text{число слів}}\right) - 15.59; \quad (1.4)$$

– індекс туманності Ганнінга:

$$GFI = 0.4 * \left[\left(\frac{\text{число слів}}{\text{число речень}}\right) + 100 * \left(\frac{\text{число складних слів}}{\text{число слів}}\right)\right]; \quad (1.5)$$

– індекс Колман-Ліау:

$$CFI = 0.0588L - 0.296S - 15.8. \quad (1.6)$$

1.6 Важливість мовного моделювання

Метою мовного моделювання є вивчення розподілу ймовірностей над послідовностями лінгвістичних одиниць, які відносяться до мови. Під лінгвістичними одиницями можна розуміти будь-які природні одиниці, на які можна розбити лінгвістичні повідомлення, наприклад, символи, слова або словосполучення. Лінгвістичні одиниці, які бачить модель, складають словник моделі U.

$$P(S) = P(u_1, u_2, \dots, u_n), \tag{1.7}$$

де S – послідовності лінгвістичних одиниць, а u_i і-та одиниця, що зазвичай досягається наданням умовних ймовірностей p(u|c).

C – контекст лінгвістичної одиниці. Наприклад, ймовірність конкретної одиниці в послідовності:

$$P(u_i|u_{i-k}, u_{i-k+1}, \dots, u_{i+k+2}). \tag{1.8}$$

Більшість мовних моделей з фіксованим словниковим запасом використовують символ <unk>, що позначає всі одиниці, які не представлені у словниковому запасі U.

У цій роботі ми називаємо ММ в залежності від вивченої лінгвістичної одиниці, відповідно, ММ на рівні слів, підсловесному рівні та на рівні символів, оскільки вони виробляють ймовірність наступної мовної одиниці, мовну модель (ММ) можна розглядати як граматику мови, і вона відіграє ключову роль у традиційних завданнях НЛП, таких як автоматичне розпізнавання мовлення (Миколов та ін.), машинного перекладу (Schloss, 2010, Arisoyet ін., 2012), автоматичного розпізнавання мовлення.

1.7 Вивчення досліджень у галузі English Language Modeling

Англійська мова – найпоширеніша мова у світі, а отже, і більшість досліджень НЛП в основному розвивається навколо неї. Написано незліченну кількість праць, зібрано величезні лінгвістичні ресурси, задано численні орієнтири та протестовано підходи. Таким чином, можна стверджувати, що НЛП для англійської мови є найдосконалішим з-поміж природних мов, тож важливо почати з глибокого занурення в дослідження, що були проведені з англійської мови, включно з відповідними текстовими корпораціями та моделями.

Таким чином, отримані знання та досвід можна застосувати й для української мови.

1.7.1 RNN

Рекурентна нейронна мережа (RNN) – це різновид штучної нейронної мережі, що використовується в обробці природної мови (NLP). RNN спроектована таким чином, щоб обробляти послідовні входи даних, що дозволяє їй ефективно працювати з тимчасовими рядами та текстовими даними. Впроваджуючи архітектуру зі схованим станом \(h(t) \) і петлями, цей тип нейронних мереж дозволяє зберігати раніше оброблену інформацію і використовувати її для обробки поточного входу. Це робить RNN особливо корисними для задач, де контекст і порядок слів мають вирішальне значення, таких як машинний переклад та аналіз послідовностей. Загальна візуалізація такої моделі представлена на рисунку 1.1. Завдяки здатності зберігати інформацію з попередніх кроків, RNN може краще зрозуміти структуру і зміст довгих текстів.

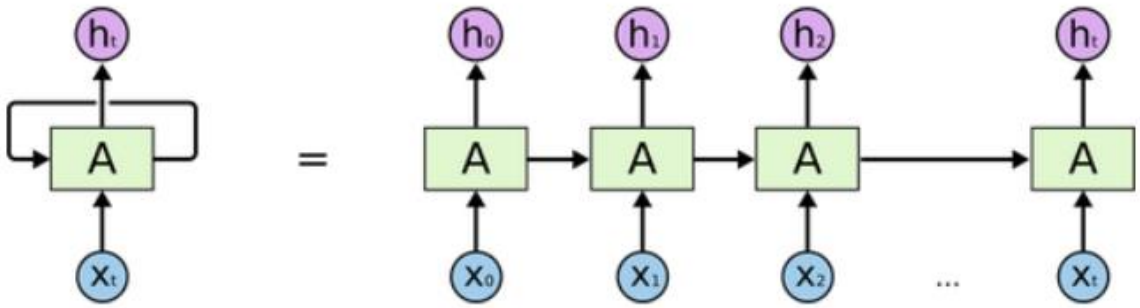


Рисунок 1.1 – Відображення структури RNN

Якщо припустити, що RNN використовується для прогнозування слів або символів, а вихід – дискретний, то математичне формулювання прямого поширення на кожному часовому кроці ванільної RNN виглядає наступним чином (Goodfellow, Bengio, and Courville, 2016):

$$a^{(t)} = b + Wh^{(t-1)} + Ux^{(t)}, \tag{1.9}$$

$$a^{(t)} = \tanh(a^t), \tag{1.10}$$

$$o^{(t)} = c + Vh^{(t)}, \tag{1.11}$$

$$\hat{y}^{(t)} = softmax(o^t), \tag{1.12}$$

де $x^{(t)}$ – вхідний вектор;

$y^{(t)}$ – вихідний вектор, що представляє розподіл ймовірностей за словниковим запасом на кроці часу t.

Параметрами моделі є вектори зсуву і матриці ваг U, V та W. Прихований стан $h^{(0)}$ ініціалізується на першій ітерації. Матриці U , V та W відповідно обчислюють зв'язки «вхід-приховане», «приховане».

Головною перевагою перед N-грамними LM, де можна було б враховувати лише кінцеве вікно з кінцевим розміром контексту, є можливість RNN обумовлювати модель усіма попередніми словами у реченні.

Навіть незважаючи на те, що попередні результати на невеликих корпораціях були багатообіцяючими, Mikolovet al., 2010 зауважили, що ванільний RNN, навчений градієнтним спусканням, не здатен вловлювати довгу контекстну інформацію.

Під час навчання дуже глибоких рекурентних нейронних мереж виникають дві поширені проблеми: закриття градієнтів та їхнє вичерпне кодування. Завдяки процесу backpropagation, який, по суті, є застосуванням ланцюгового правила, градієнт із найпізніших шарів множиться на матриці, щоб потім поширюватися на початок мережі. Якщо найбільші власні значення цих матриць менші за одиницю, то значення градієнта дорівнюватиме нулю, якщо в мережі багато шарів, що називається проблемою зникаючих градієнтів. З іншого боку, якщо найбільші власні значення більші за одиницю, то значення градієнта буде йти до нескінченності – це проблема вибухаючих градієнтів.

1.7.2 Transformer

Попередньо навчені мовні моделі стали ключовим елементом у досягненні state of the art результатів у багатьох завданнях обробки текстових даних. Ці моделі розширюють ідеї кодування слів, розглянуті раніше, головним чином завдяки вивченню й отриманню контекстуальних репрезентацій із великих наборів даних, використовуючи цільове завдання моделювання мови. Розглянуті моделі NVDM і ELMo є прикладами перших універсальних мовних моделей, застосування яких дало змогу отримати найкращі на свій час (2019 рік) результати з цілої низки завдань аналізу текстових даних. Однак наразі їхнє застосування виправдане лише в

рідкісних випадках (наприклад, у разі неможливості розпаралелювання аналізу даних із технічних причин), зважаючи на велику кількість переваг, які надають більш сучасні підходи.

Transformer був розроблений як архітектура перетворення послідовностей. Прикладами завдань, де вхідна послідовність перетворюється на вихідну, можуть слугувати машинний переклад, розпізнавання мови, сумаризації та багато інших областей. Традиційно ці завдання розв'язували рекурентні нейромережі [11] та їхні модифіковані версії на основі довгої короткострокової пам'яті (Long-ShortTermMemory) [12].

У базовій версії рекурентна мережа складається з двох компонентів: кодувальника і декодувальника. Обидві ці частини працюють як стандартні рекурентні мережі, тобто використовують на вході не тільки інформацію з введення, а й інформацію з попередніх виходів моделі. При цьому, на відміну від найпростішого випадку застосування рекурентної мережі, використання двох мереж дає змогу обійти обмеження нерівності довжини входу і виходу моделі та порядку слів, тобто немає необхідності в прямій відповідності вхідних даних і цільових значень. Кодувальник у цьому разі отримує вектор і прихований стан, який потім використовується спільно з вектором кодування наступного слова. Декодувальник, зі свого боку, отримує на вхід кодування і виводить послідовність слів. Основним недоліком цього підходу є неможливість паралелізації обчислень, оскільки для кожного наступного слова потрібен прихований стан кодувальника попереднього слова. Також великою проблемою є неможливість враховувати зв'язки слів у великих реченнях (алгоритм віддає перевагу найближчим попереднім словам). LSTM модифікує цю структуру за допомогою комірок стану.

Ці комірки дають змогу додавати або видаляти інформацію за допомогою, так званих, гейтів. Гейти дають змогу частково пропускати інформацію і складаються із сигмоїдного шару та операції поелементного

множення. Подібний механізм вибіркової пам'яті дає змогу краще вловлювати довгострокові зв'язки в реченнях, однак має всі ті самі проблеми з паралелізацією

Transformer модель вирішує проблеми стандартних підходів, використовуючи архітектуру, повністю побудовану на механізмі внутрішньої уваги (self-attention) [7]. На рисунку 1.2. представлено архітектуру цієї моделі. Ліворуч на цій схемі зображено кодувальник, а праворуч – декодувальник, при цьому ці блоки можуть дублюватися k разів (в оригінальній статті використовується 6 послідовних блоків).

Основними компонентами кодувальника і декодувальника є модулі внутрішньої уваги і шари прямого поширення (FeedForwardLayers). Входи і виходи (цільові пропозиції) спочатку вбудовуються в n–вимірний простір. При цьому важлива частина моделі позиційне кодування різних слів.

Оскільки ця модель не використовує рекурентні мережі, які можуть запам'ятати, як послідовності слів використовуються в моделі, виникає потреба присвоїти кожному слову відносне положення, оскільки послідовність залежить від порядку її елементів. Ці позиції додаються до вбудованого подання (n-вимірного вектора) кожного слова, що дозволяє моделі враховувати порядок слів у реченні.

Функцію внутрішньої уваги при цьому можна описати як відображення входу і набору пар ключ–значення на вихід, де запит, ключі, значення і вихідні дані є векторами. Вихідні дані обчислюються як зважена сума значень, де вагу, присвоєну кожному значенню, обчислює функція сумісності запиту з відповідним ключем. Цей механізм дозволяє моделі фокусуватися на релевантних частинах вхідного тексту, виділяючи найважливіші елементи для генерування виходу.

Таким чином, механізм уваги значно покращує здатність моделі до обробки складних залежностей у тексті та підвищує точність в задачах, таких як машинний переклад, реферування та питання-відповідь. Зокрема, використання multi-head attention дозволяє моделі паралельно аналізувати

різні аспекти вхідного тексту, що ще більше підвищує її ефективність та продуктивність (рисунок 1.2).

Основні переваги моделі:

– покращене моделювання далеких залежностей слів;

– на відміну від минулих підходів на основі RNN, можлива паралелізація методів.

Основні недоліки:

– складність перенавчання seq2seq моделей, оскільки архітектура складається з двох частин;

– механізм уваги працює з рядками фіксованої довжини, як наслідок вимагає розбиття, що в таких випадках веде до втрати контекстної інформації.

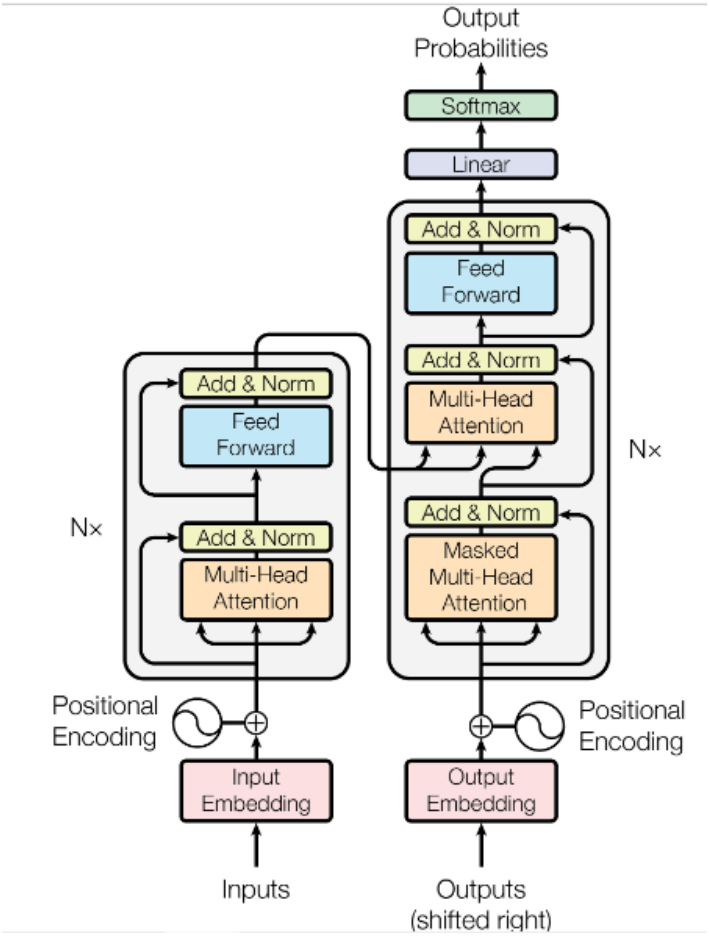


Рисунок 1.2 – Загальна структура архітектури Transformer моделі

Недоліком усіх перелічених вище методів є те, що вони призводять до втрати інформації. Кращою є реалізація, за якої вся послідовність обробляється за один прохід.

Однією з останніх тенденцій у побудові трансформерів для роботи з довгими послідовностями є одночасне використання механізмів глобальної та локальної пам'яті (рисунок 1.3).

У трансформерах із повним механізмом уваги збільшення зони уваги супроводжується збільшенням використовуваної пам'яті згідно з квадратичним законом n^2. Трансформери ETC, BigBird і Longformer використовують суміщений механізм уваги.

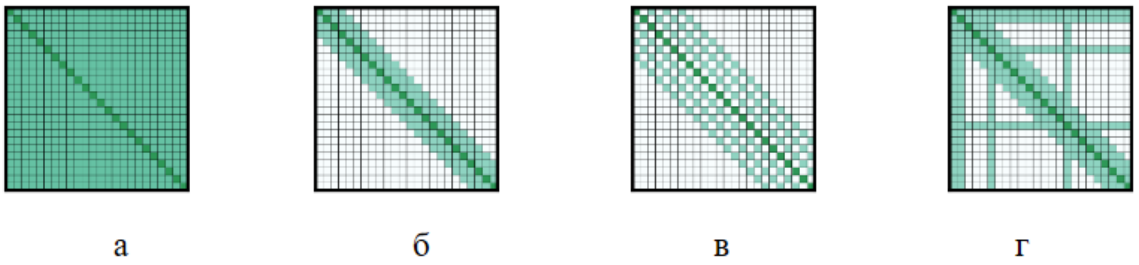


Рисунок 1.3 – Види уваги в Longformer

На даний момент найкращі результати з перерахованих моделей показують моделі сімейства Longformer.

2 ВИЛУЧЕННЯ ІНФОРМАЦІЇ З ТЕКСТІВ

2.1 Специфіка завдань, підходи до розв'язання, видобута інформація

Завдання вилучення інформації з текстів природною мовою можна віднести до інформаційного пошуку (Information Retrieval) в його найширшому розумінні, що передбачає пошук релевантної інформації. Однак у напряму Information Extraction є принципові особливості. На відміну від класичного пошуку, що виконується пошуковими машинами мережі Інтернет і видає користувачеві список відренжованих сніпетів, на виході IE-систем – структурована інформація, витягнута з колекції текстів (або одного великого тексту), що так чи інакше передбачає перетворення витягнутої інформації. Загалом, у рамках IE розв'язується завдання автоматичного вилучення з текстів даних, релевантних певній проблемі/питанню/темі, з неструктурованих текстів [19]. Такі тексти не мають жодної розмітки або метаданих, що допомагають ідентифікувати шукану інформацію. Для зручності подальшого опрацювання та застосування витягнуті дані структурують: у найпростішому випадку за допомогою тегів XML, у складніших вони перетворюються та зберігаються у формальному вигляді: у реляційних базах даних, таблицях, мережевих базах знань. Структуровані дані передаються засобам аналітичної обробки (OLAP, Data Mining) або ж візуалізуються для людини-аналітика у вигляді семантичних мереж, когнітивних карт тощо [5].

Перші прикладні дослідження в галузі вилучення інформації відносяться до початку 1980-х років, вони стосувалися обробки новинних і військових текстів з метою виділення в них певних подій [19]. Нині різні дані витягують також із художніх творів, науково-технічних статей, текстів мережі Інтернет (журналів, блогів та ін.). Таким чином, до розглянутої галузі стали відносити вилучення будь-яких семантично значущих даних із текстів різних функціональних стилів, у тому числі:

– імен персонажів і пов'язаних із ними подій із художніх творів;

– назв білків, генів, хвороб, ліків із текстів із медицини;

– термінів та їхніх смислових зв'язків зі спеціалізованих текстів;

– ключових слів і словосполучень з індексованих текстів;

– думок з приводу продуктів і послуг з інтернет-текстів відгуків.

Крім того, ведуться дослідження в галузі обробки мультимедійних документів: зображення, аудіо- та відеофайли автоматично обробляються, з них витягується вміст, на підставі якого потім для файлів складається опис.

Як дані, що витягуються з текстів, зазвичай виступають:

– значущий об'єкт: ім'я персоналії, назва компанії тощо для новинних повідомлень, термін предметної області спеціального тексту, посилання на літературу для науково-технічних документів тощо;

– атрибути об'єкта, що додатково характеризують його, наприклад, для компанії це юридична адреса, телефон, ім'я керівника тощо;

– відношення між об'єктами: наприклад, відношення «бути власником» пов'язує компанію і персону-власника, «бути частиною» з'єднує факультет і університет;

– подія/факт, що пов'язує кілька об'єктів, наприклад, подія «пройшла зустріч» включає учасників зустрічі, а також місце і час її проведення.

Згідно з видами інформації, що витягується, загальне завдання вилучення інформації з текстів включає такі основні підзадачі:

– розпізнавання та вилучення іменованих сутностей (named entities): І.Я. Франко, Запоріжжя, ПКО «Картографія» тощо;

– виокремлення атрибутів (attributes) об'єктів і семантичних відношень (relations) між ними: дати народження персони, відношення «працювати в» тощо;

– витяг фактів і подій (events), що охоплюють кілька їхніх параметрів (атрибутів), наприклад, подія «корабельна аварія» з атрибутами дата, час, місце тощо).

У нижченаведеному реченні:

«У жовтні 2005 року компанія eBay за 2,6 мільярда $ купила 30% акцій Skype»

Містяться кілька видів інформації, що витягується: подія «купити» і її параметри (атрибути): покупець, об'єкт покупки, час, ціна, причому параметри представлені іменованими сутностями.

Наразі найбільш дослідженим підзавданням є розпізнавання іменованих сутностей (NER: Named Entity Recognition).

Для розв'язання завдань розпізнавання та вилучення інформації з текстів використовують два головні підходи: заснований на правилах (rule-based), або інженерний, і заснований на машинному навчанні (machine learning). Зазначимо, що з'являється все більше гібридних методів, що враховують переваги обох підходів.

Перші IE-системи були побудовані в рамках інженерного підходу, найвідомішою з них була AutoSlog. Серед перших вітчизняних розробок варто згадати сімейство багатомовних систем вилучення інформації з ділових текстів OntosMiner, що забезпечували перехід від неструктурованої інформації до її семантичного представлення у форматі онтологій предметних галузей, закладених у систему (бізнес-події, судова тематика та поліцейські звіти).

Розроблення прикладних IE-систем є складним і трудомістким процесом, істотну допомогу в якому можуть надати інструментальні системи, що включають стандартні модулі аналізу тексту і навіть засоби збирання та налагодження додатків. Інструментальні системи, призначені для розроблення застосунків у межах інженерного підходу, мають зазвичай

вбудовану формальну мову для задання лінгвістичних правил і шаблонів - за їхньою допомогою стандартні програмні модулі налаштовують на розв'язання конкретного прикладного завдання. Інструментальні системи, що спираються на машинне навчання, дають змогу використовувати вже побудовані програмні моделі (класифікатори), а також навчати нові. Зрозуміло, що застосування готових моделей обмежується тією проблемною областю, для якої ці моделі були побудовані. Наприклад, у пакеті OpenNLP – це вилучення імен персоналій, географічних об'єктів, дат, часу тощо з новинних статей. Для отримання нової моделі необхідна навчальна вибірка, тобто розмічений за певними правилами текстовий корпус.

2.2 Розв'язання морфологічної омонімії

Для вирішення цієї проблеми існує три основні підходи:

– заснований на правилах;
– заснований на статистиці;
– заснований на машинному навчанні.

Метод, заснований на правилах, застосовується, наприклад, у роботі [5]. Тут були написані окремі модулі зняття омонімії, що дозволяють її тільки в певних випадках залежно від контексту і самих омонімічних слів, їхньої частини мови або набору параметрів. Суть методу зводиться до того, що в таких ситуаціях аналіз контексту допомагає зрозуміти синтаксичну структуру частини речення, а з її допомогою і форми слів.

Однак цей метод вимагає ручного складання правил, тобто тривалої і кропіткої роботи. Для кожного з правил потрібно написати самостійний програмний модуль. Поповнення системи правил стає дедалі важчим із кожним новим правилом. У зв'язку з цим подібні методи не набули широкого поширення.

Набагато частіше в сучасних морфологічних процесорах застосовуються статистичні методи і методи, засновані на машинному навчанні. Це пов'язано насамперед із наявністю відкритих, розмічених корпусів, об'єму яких достатньо для побудови досить точних моделей.

Підрахунок статистики різних варіантів розбору за корпусом є найпростішим способом зняття морфологічної омонімії. При цьому, за розміченим корпусом зі знятою омонімією.

Для вирішення омонімії може бути реалізовано метод простого підрахунку ймовірності $P(t|w)$ для кожного з набору тегів і слів корпусу (уніграмний метод):

$$P(t|w) = \frac{Fr(w,t)}{Fr(w)}, \quad (2.1)$$

де w – слов;

t – набір тегів;

$Fr(w)$ – скільки разів слово зустрілося в корпусі;

$Fr(w,t)$ – скільки разів слово зустрілося в корпусі з набором тегів t.

Фактично, у цьому методі розраховується апостеріорна ймовірність зустріти дану словоформу серед усіх варіантів вживання в тексті заданого токена.

Наприклад, якщо в нашому розміченому корпусі токен скло зустрівся 100 разів, при цьому у формі дієслова він зустрівся 5 разів, у формі називного відмінка іменника – 60 разів і у формі знахідного відмінка – 35 разів, то ймовірності відповідних форм дорівнюватимуть 0.05, 0.6 і 0.35. Тепер, зустрічаючи в новому, нерозміченому, тексті токен скло, ми будемо ухвалювати рішення, що він має бути називним відмінкам іменника (найімовірніше рішення). Підсумкова точність визначення леми буде приблизно 0.95 (один раз на двадцять уживань усе-таки потрапляє

дієслово), точність визначення набору граматичних параметрів становитиме 0.6 (ще 35 разів зі 100 ми пропустимо знахідний відмінок).

Точніші результати дає врахування контексту слова. Наприклад, якщо ми зустріли в тексті іменникову групу, що складається з кількох прикметників та іменника, то всі слова в ній мають бути узгоджені між собою. Ба більше, якщо ми зустріли прийменникову групу, то прикметники та іменник у ній не можуть перебувати в називному відмінку. Взагалі, відмінок слів у прийменниковій групі визначатиметься прийменником, що стоїть на початку. Тобто, якщо ми дивитимемося не на одне слово, а на його сусідів, то точність розпізнавання омонімії має підвищитися.

У цьому випадку використовують триграмну модель – аналіз слова та його контексту зі ще двох слів. Як показує практика, триграмна модель показує значно кращі результати, ніж уні або біграмні. Чотириграмна модель займає значно більше місця, але серйозного приросту в точності не дає [18].

Триграмна модель може використовуватися в різних варіантах. Можна вибрати словоформу з максимальною ймовірністю зустрічальності за умови попередніх двох слів:

$$w_i = argmaxP(w_i|w_{i-1}, w_{i-2}). \quad (2.2)$$

Взагалі, під триграмою можуть розумітися різні конструкції. По-перше, це може бути три словоформи, що йдуть підряд. Однак імовірність зустріти саме ці три словоформи в довільному тексті може виявитися дуже низькою. Але якщо відкинути леми, ймовірність зустріти подібну триграму стає значно вищою. Але й у цьому разі може вийти, що в тексті зустрінеться триграма, яка не зустрічалася раніше (наприклад, у зв'язку з тим, що тексти різного стилю мають різну частоту зустрічальності для частин мови або взагалі синтаксичних конструкцій). При цьому біграми, що її складають, у

корпусі вже зустрічалися. У такій ситуації можна використовувати комбінації біграм:

$$w_i = argmaxP\big(w_i|w_{i-1}) * P((w_i|w_{i-2})\big). \quad (2.3)$$

У загальному випадку можна використовувати згладжування: з різними ваговими коефіцієнтами береться інформація про триграму, біграму та уніграму на випадок, якщо якась частина інформації відсутня в аналізованому тексті:

$$w_i = argmax\big(\lambda_1 P(w_i|w_{i-1}, w_{i-2}) + \lambda_2 P(w_i|w_{i-2}) + \lambda_3 P(w_i)\big). \quad (2.4)$$

У всіх розглянутих варіантах ми намагалися передбачити поточне слово за попередніми. Однак на практиці може виявитися, що поточне слово пов'язане з кількома наступними. Також можливий варіант, коли слово буде пов'язане як із лівим, так і з правим контекстом. У такій ситуації можна використовувати статистичну інформацію не за однією, а за трьома триграмами, у яких поточне слово займатиме різні позиції:

$$w_i = P(w_i|w_{i-1}, w_{i-2}) - P(w_i|w_{i-1}, w_{i+1}) - P(w_i|w_{i+1}, w_{i+2}). \quad (2.5)$$

Крім статистичних методів, для зняття омонімії зараз використовують цілий спектр методів класифікації. Уявімо, що кожна словоформа, яка є результатом морфологічного аналізу обраного токена, належить до одного з двох класів: коректне пророкування і некоректне пророкування. За такої постановки завдання можна провести бінарну класифікацію словоформ. Параметрами класифікації можуть слугувати граматичні параметри даного або сусідніх слів у такому вікні, їхні леми, ознаки наявності розділових знаків тощо. Вибір методу класифікації багато в чому залежить від

уподобань розробника, для розв'язання якої використовують такі методи машинного навчання, як приховані марковські моделі, умовні випадкові поля, рекурентні нейронні мережі та ін. Для навчання методу класифікації також використовують розмічені корпуси, наприклад, НКУМ із вручну знятою омонімією.

Для контекстного зняття омонімії може використовуватися, наприклад, метод CRF (умовні випадкові поля). Цей метод добре зарекомендував себе в задачах комп'ютерної лінгвістики, таких як визначення частини мови, визначення іменованих сутностей тощо. CRF є дискримінативною ймовірнісною моделлю. Однією з головних переваг цієї моделі є те, що вона не вимагає моделювати ймовірнісні залежності між так званими спостережуваними змінними.

Зняття омонімії за всіма морфологічними характеристиками (тегами) є складним для навчання CRF-класифікатора через велику кількість тегів, що вимагає ускладнення моделі. Застосовуються чотири навчені CRF-класифікатори, які послідовно відсікають омонімічні варіанти. Першим працює CRF-класифікатор для частини мови, ознаки, які він використовує – словоформа і можливі частини мови (у вигляді бінарного вектора). Потім застосовується класифікатор для роду (ознаки: словоформа, вже визначена частина мови, можливі варіанти роду: чоловічий, жіночий, середній). Після цього аналогічним чином працюють CRFкласифікатори числа і відмінка. На рисунку 2.1 показано приклад набору ознак для класифікатора роду.

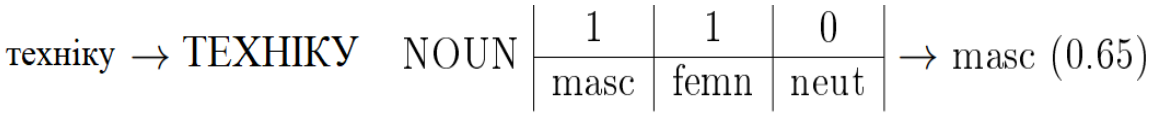


Рисунок 2.1 – Приклад набору класифікаторів

Класифікатори застосовуються послідовно, накопичуючи помилку попередніх етапів. Частковий приклад такого застосування показано на рисунку 2.2. Тут слово мила, що класифікується, проходить через усі чотири класифікатори: на першому етапі визначається частина мови (іменник), потім класифікатор роду обирає середній рід, потім класифікатор числа визначає, що слово належить до однини, а класифікатор відмінка обирає єдиний можливий варіант.

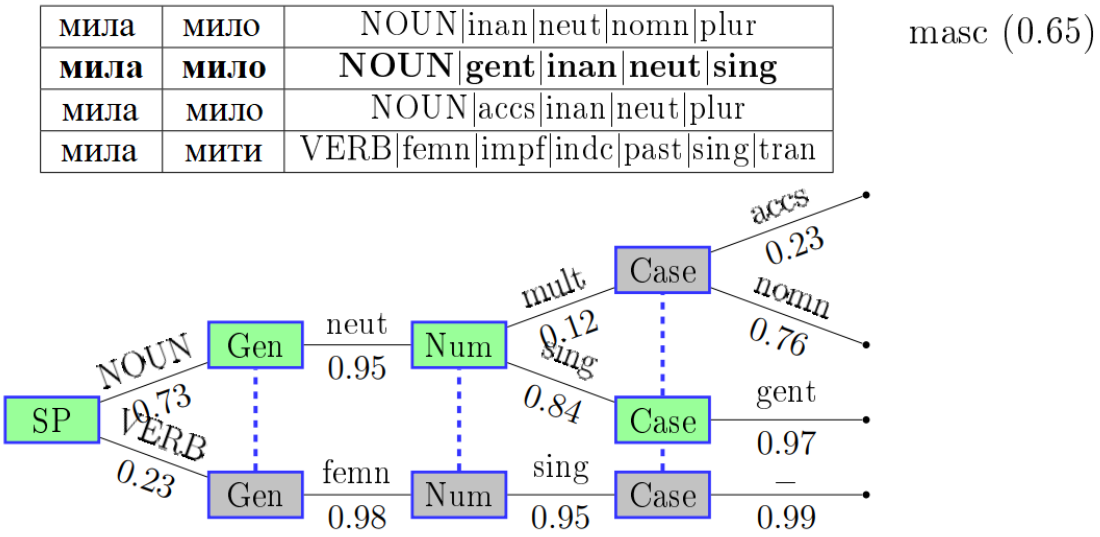


Рисунок 2.2 – Приклад помилки класифікатора

Зазначимо, що трапляються ситуації, коли після опрацювання словоформи всіма чотирма класифікаторами все одно зберігається морфологічна омонімія, наприклад, за лемою або натхненністю. У такому разі проводиться безконтекстне зняття на основі статистики за корпусом. У підсумку залишається єдиний варіант розбору

2.3 Автоматичне породження оціночних словників

Велику увагу приділяють автоматизації побудови словників оціночної лексики для конкретних мов або предметних областей. Словник оцінної

лексики для заданої природної мови може бути створено за допомогою перекладу та інтеграції оцінних словників, що існують іншими мовами .

Окремим напрямом досліджень є автоматичне вилучення з текстів оціночних слів і виразів, оскільки підкреслюється, що оціночні вирази, які використовуються, значною мірою залежать від предметної області та від типу оцінюваної сутності.

У класичній роботі виокремлення оціночних прикметників і визначення їхньої семантичної спрямованості ґрунтується на синтаксичних шаблонах і сполучниках І, АБО, АЛЕ. Передбачається, що, якщо два прикметники пов'язані сполучниками І або АБО, то вони обидва є або не є оціночними, а так само однаково семантично спрямовані. У разі сполучника АЛЕ семантичний напрям розрізняється. У результаті було побудовано класифікатор зв'язків, що працює з точністю 82%. На останньому кроці виконували кластеризацію слів, у результаті якої утворилося два кластери, більший із яких обирали позитивним. Точність кластеризації 92%.

Для отримання оціночних слів і обчислення їхньої спрямованості можуть використовуватися словники й тезауруси. Метод, передбачає використання тезауруса для збагачення, заданої вручну, еталонної множини оціночних слів. Основна ідея полягає в тому, що якщо слово оцінне, то його синоніми, гіпоніми також будуть оцінними й однаково семантично спрямовані, у разі антонімів – протилежно спрямовані.

Важливим завданням є створення словників оціночних слів і виразів для конкретних предметних галузей, оскільки такий словник є значною мірою залежним від предметної галузі: деякі оціночні вирази вживаються тільки в конкретних предметних галузях, інші є оціночними в одній галузі та не є оціночними в іншій.

Один із частих підходів вилучення словника оціночних слів для заданої предметної області полягає в завданні набору загальнозначущих оціночних слів, а потім поповнення цього набору на основі корпусу текстів.

У роботі [16] описано систему OPINE, яка слугує для вилучення з відгуків атрибутів описаних продуктів, а також оцінок за ними. OPINE виділяє такі атрибути продукту: властивості продукту, частини продукту, атрибути частин продукту, пов'язані сутності, властивості та частини пов'язаних сутностей. Передбачається, що оціночні фрази з'являються в безпосередній близькості від атрибутів об'єкта. Для вилучення оціночних слів використовується 10 правил, заснованих на синтаксичній структурі речення:

$$(M, NP = f) \Rightarrow \text{p}o = M\text{:}\,(expensive)scanner, \tag{2.6}$$

$$(S = f, P, O) \Rightarrow \text{p}o = O\text{:}\,(expensive)scanner, \tag{2.7}$$

$$(S, P, O = f) \Rightarrow \text{p}o = P\text{:}\,I\ (hate)this\ scanner, \tag{2.8}$$

$$(S = f, P, O) \Rightarrow \text{p}o = P\text{:}\,program\ (crashed), \tag{2.9}$$

де M – модифікатор;

NP – іменна група;

S – підмет;

P – предикат;

O – об'єкт;

f – ознака;

p_o – кандидат в оціночні слова.

Визначення семантичної орієнтації слів базується на низці чинників, включно з уживанням зі сполучниками, врахуванням словотворення, інформацією про синоніми та антоніми з тезаурусу WordNet.

У WordNet розглядається метод автоматичного вилучення оціночних слів на основі кількох корпусів текстів, які існують для багатьох предметних областей, а саме:

– корпус відгуків про сутності з оцінками, вручну проставленими споживачами;

– корпус нейтральних описів сутностей, наприклад, сюжети фільмів,

– нейтральний контрастний корпус загальнозначущих новин.

Із зазначених корпусів витягують списки слів; для кожного слова розраховують набір статистичних характеристик (частотності, відносні частотності між корпусами, різні кореляції появи слова та користувацької оцінки до відкликання), а також ураховують лінгвістичні чинники, як-от наявність префіксів або написання з великої літери. Далі використовуються методи машинного навчання для отримання якісного списку оціночних слів, характерних для даної предметної галузі.

Результат має вигляд упорядкованого списку слів, розташованих у порядку зниження ймовірності оціночності конкретного слова, передбаченого класифікатором. Цей метод не дає можливості проставити оцінку тональності слова, але зосереджує можливі оціночні слова ближче до початку списку, що полегшує їхній перегляд експертами для розмітки за тональністю.

Класифікатор для вилучення оціночних слів із таких корпусів було навчено на даних відгуків про фільми, а потім навчену модель було застосовано до інших предметних областей. Було показано, що модель добре переноситься на інші предметні області. Наприклад, оцінка застосування моделі в предметній області відгуків про книги показала, що в першій тисячі отриманого списку міститься понад 85% оціночних слів.

Наведемо приклад початку списку передбачуваних оціночних слів, витягнутого за описаною моделлю для предметної області відгуків про ресторани, із передбаченими ймовірностями їхньої оціночності:

– несмачний 0.970;
– несмачний 0.964;
– незатишний 0.956;
– неуважний 0.939;
– невимушений 0.937;
– пафосний 0.924;
– неквапливий 0.919.

Таким чином, можна виявляти конотації слів, виявляючи різницю частотності їхньої зустрічальності з позитивними і негативними оціночними словами.

Одним із відомих підходів для вилучення оціночних слів із текстів є підхід, у якому пропонувалося задати деяку множину вихідних позитивних і негативних слів, а для решти слів нараховувати спільну зустрічальність із заданими позитивними і негативними словами. Для оцінювання оцінної орієнтації слів (SO) було запропоновано використовувати формулу потокової взаємної інформації PMI у такий спосіб:

$$SO(w) = PIM(w, Pos) - PMI(w, Neg). \quad (2.10)$$

Потокова взаємна інформація визначається таким чином:

$$PMI(w_i, w_j) = log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}, \quad (2.11)$$

де $p(w_i)$ – це ймовірність зустріти слово в корпусі, зазвичай обчислюється як відношення кількості входжень слова $p(wi)$ до загальної кількості слів у корпусі.

Істотною проблемою цього простого методу була необхідність завдання вихідної множини позитивних і негативних слів, а також необхідність збору об'ємної текстової колекції. Надалі виявилося, що цей метод добре застосовний для вилучення оціночних слів і виразів із повідомлень Твіттера. При цьому самі повідомлення легко зібрати у великій кількості, користуючись API Твіттера. Крім того, не потрібно задавати безлічі вихідних оціночних слів, оскільки як такі слова використовуються позитивні й негативні хештеги та емотикони, проставлені самими користувачами. Відомо, що оцінна орієнтація хештега (або емотикона) далеко не завжди відповідає оцінній орієнтації повідомлення, якому цей хештег (емотикон) приписано, проте загалом такі дані можна використовувати цілком ефективно.

2.4 Відомі методи сумаризації тексту

2.4.1 LSA (Latent Semantic Analysis)

Латентно-семантичний аналіз (Latent Semantic Analysis, LSA) передбачає створення структурованих даних із колекції неструктурованих текстів. Перш ніж заглибитися в концепцію LSA, давайте швидко інтуїтивно зрозуміємо, що це за поняття. Коли ми пишемо щось на кшталт тексту, слова не вибираються випадковим чином зі словника.

Скоріше ми думаємо над темою (або темою), а потім підбираємо слова таким чином, щоб більш змістовно висловити свої думки іншим. Цю тему або тему зазвичай розглядають як латентний вимір.

Він латентний тому, що ми не бачимо цей вимір явно. Навпаки, ми розуміємо його лише після того, як пройдемо крізь текст. Це означає, що більшість слів семантично пов'язані з іншими словами, щоб висловити тему. Отже, якщо слова трапляються в колекції документів із різною частотою, це

має вказувати на те, як різні люди намагаються виразити себе, використовуючи різні слова та різні теми чи теми.

Іншими словами, частоти слів у різних документах відіграють ключову роль у вилученні латентних тем. LSA намагається витягти виміри за допомогою алгоритму машинного навчання, який називається розкладанням сингулярних значень (Singular Value Decomposition, SVD).

Розкладання сингулярних значень (Singular Value Decomposition, SVD) – це, по суті, техніка факторизації матриць. У цьому методі будь-яку матрицю можна розкласти на три частини, як показано нижче на рисунку 2.3.

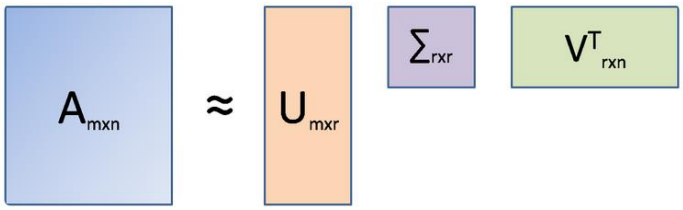


Рисунок 2.3 – Схема роботи Latent Semantic Analysis

Тут A – матриця документів-термінів (документи в рядках (m), унікальні слова в стовпчиках (n), а також частоти в точках перетину документів і слів). Слід пам'ятати, що в LSA вихідна матриця документ-термін апроксимується множенням трьох інших матриць, а саме: U, ∑ і VT. Тут r – кількість аспектів або тем. Після того, як ми зафіксуємо r (r<<n) і запустимо SVD, результат, який вийде, називається усіченим SVD, і LSA – це, по суті, лише усічений SVD.

SVD використовується в таких ситуаціях тому, що, на відміну від PCA, SVD не вимагає для розкладання ні кореляційної матриці, ні коваріаційної матриці. У цьому сенсі SVD вільна від будь-якого припущення про нормальність даних (розрахунок коваріації передбачає нормальний розподіл даних). Матриця U – матриця «документ-аспект», V –

матриця «слово-аспект», а $\sum$ – діагональна матриця сингулярних значень. Подібно до PCA, SVD також лінійно об'єднує стовпці вихідної матриці, щоб отримати матрицю U. Щоб отримати матрицю V, SVD лінійно об'єднує рядки вихідної матриці. Таким чином, із розрідженої матриці «документ-термін» можна отримати щільну матрицю «документ-об'єкт», яку можна використовувати як для кластеризації документів, так і для їхньої класифікації за допомогою наявних інструментів ML. Матриця V, навпаки, є матрицею вбудовування слів (тобто кожне слово виражається r числами з плаваючою комою), і цю матрицю можна використовувати в інших завданнях послідовного моделювання. Втім, для таких завдань доступні вектори Word2Vec і Glove, які є більш популярними.

Кластеризація документів корисна багатьма способами кластеризації документів на основі їхньої схожості між собою. Вони корисні в юридичних фірмах, сегрегації медичних записів, сегрегації книжок і в багатьох інших сценаріях. Алгоритми кластеризації, як правило, призначені для роботи з щільною матрицею, а не з розрідженою, яка утворюється при створенні матриці термінів документа. За допомогою LSA можна створити апроксимацію вихідної матриці з низьким ранговим коефіцієнтом рангів (з деякими інформаційними втратами, правда, з певною втратою інформації!), яку можна використати для нашої мети – кластеризації. У наступних кодах показано, як створюється матриця термінів документів і як LSA можна використовувати для кластеризації документів (рисунок 2.4).

На цій діаграмі лише новинна стаття, пов'язана з технологіями, виглядає так, що вона має значно ширший розкид, тоді як інші новинні статті виглядають доволі гарно кластеризованими. Це також свідчить про те, що LSA (або Truncated SVD) добре попрацював над текстовими даними, витягнувши 200 важливих вимірів, щоб відокремити новинні статті за різними темами. Слід розуміти, що TSNE є недетермінованим за своєю природою, і багаторазові прогони дадуть кілька репрезентацій, навіть якщо структура з більшою ймовірністю залишиться схожою, а то й однаковою.

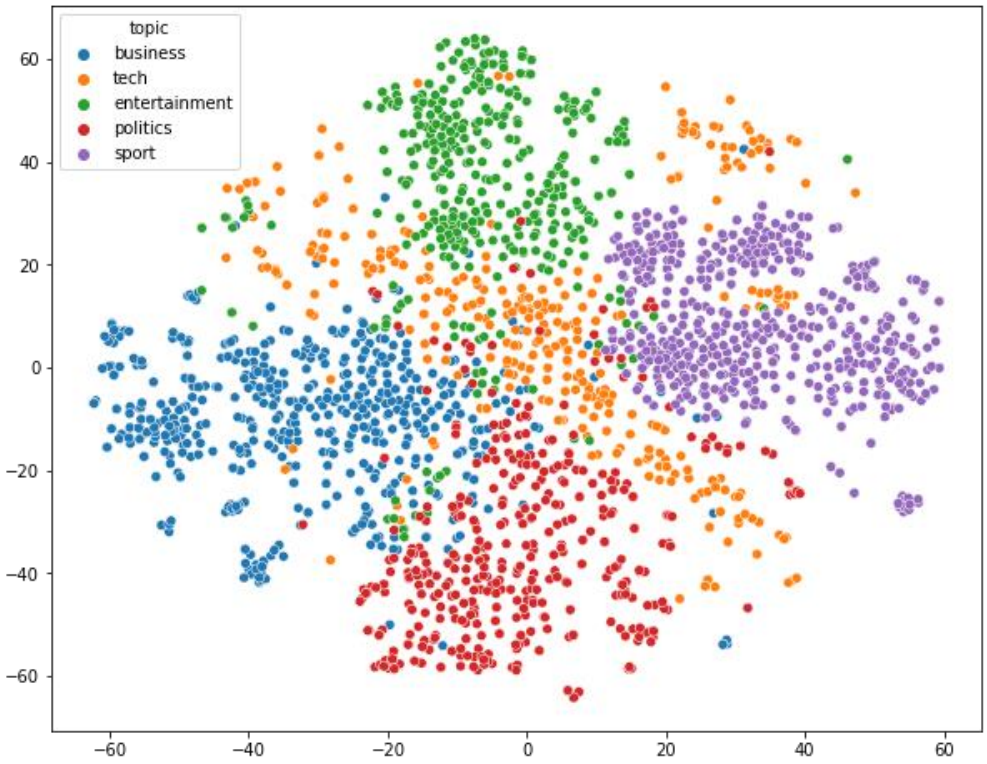


Рисунок 2.4 – Результат кластеризації з Latent Semantic Analysis

2.4.2 Трансформатори T5

Основна ідея, що лежить в основі роботи «Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer» [17], полягає в тому, щоб розглядати кожну проблему опрацювання тексту як проблему «перетворення тексту на текст», тобто приймати текст як вхідні дані і створювати новий текст у результаті роботи моделі. Подібний підхід зустрічався раніше в інших моделях з уніфікованими структурами для різних завдань НЛП. Прикладами можуть слугувати: фреймворк, що перетворює всі завдання опрацювання тексту на завдання відповідей на запитання [18] і мовне моделювання [19]. Важливо зазначити, що структура перетворення тексту в текст дає змогу безпосередньо застосовувати одну й ту саму модель, мету, процедуру навчання і процес декодування до кожної

задачі, що розглядається, що особливо актуально для розглянутої в даній роботі мети – всебічного аналізу корпусів користувацьких повідомлень.

Використання подібної гнучкої моделі дає змогу отримати результат і оцінити ефективність для широкого кола завдань НЛП, включно з узагальненням документів (abstarctive summarization) і класифікацією настроїв (sentiment analysis). На рисунку 2.5. показано схематичний приклад роботи такої моделі, що отримала назву Text-to-Text Transfer Transformer.

З цього прикладу можна побачити, як використання такої моделі не тільки спрощує ідеї, що лежать в основі різних завдань, а й дає змогу порівнювати ефективність різних цільових функцій трансферного навчання, отримувати якісні результати на даних, що не мають попередньої розмітки.

У ключових моментах, дана реалізація Трансформер-моделі кодувальника-декодера повністю відповідає його первісно запропонованій формі, описаній раніше в цьому розділі.

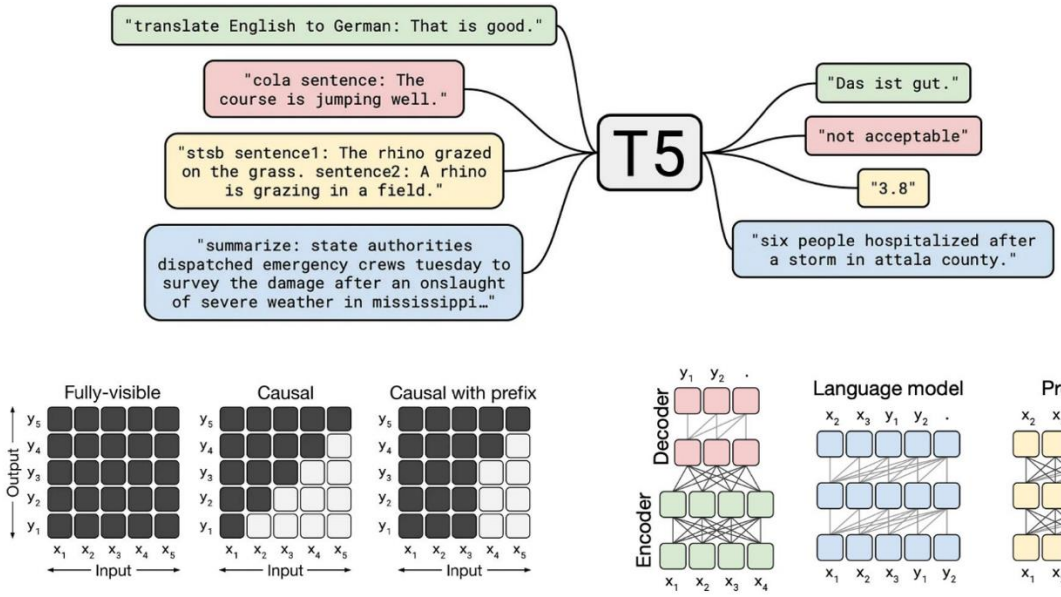


Рисунок 2.5 – Приклад роботи T5 моделі

Спочатку вхідна послідовність токенів кодується набором ембедингів, які потім передаються в кодувальник. Кодувальник складається зі

стека «блоків», кожен з яких складається з двох компонентів: рівня self-attention, за яким слідує невелика нейромережа прямого поширення. Нормалізація шару [20] застосовується до входу кожного компонента. Автори використовують спрощену версію нормалізації шарів, у якій активації лише масштабуються, без застосування адитивного зміщення. Після шару нормалізації Residual skip connection [21] додає вхід кожного підкомпонента до його вихідних даних.

Dropout застосовується в мережі з прямим зв'язком, для пропуску частини з'єднань, у вагах шару уваги, а також на вході та виході всієї структури. Декодер схожий за структурою на кодувальник, за винятком того, що він включає стандартний механізм уваги після кожного шару самоусвідомлення, який обслуговує вихідні дані кодера. Механізм самоусвідомлення в декодері також використовує форму авторегресії або причинного самоусвідомлення, що дає змогу моделі звертати увагу лише на минулі результати. Вихідні дані останнього блоку декодера передаються до щільного шару з виходом softmax, ваги якого розділяються з вхідним набором ембедингів. Усі механізми уваги в Transformer розділені на незалежні частини, виходи яких об'єднуються перед подальшим опрацюванням.

Одним із ключових досягнень авторів оригінальної статті став збір і підготовка даних для навчання моделі. Велика частина попередньої роботи з трансферного навчання для НЛП використовувала великі нерозмічені набори даних для навчання без вчителя. Щоб створити набори даних, автори використовували Common Crawl як джерело тексту, витягнутого з Інтернету. Common Crawl раніше використовували як джерело текстових даних у різних завданнях NLP, наприклад, для навчання мовної моделі n-грам, як навчальні дані в завданні «аналізу здорового глузду», для інтелектуального аналізу даних, збирання паралельних текстів у завданні машинного перекладу, як набір даних для попереднього навчання, і просто як великий текстовий корпус для тестування оптимізаторів.

Common Crawl – це загальнодоступний веб-архів, який надає витягнутий з Інтернету текст із видаленням мета-інформації та іншого нетекстового вмісту з очищених файлів HTML. Цей процес виробляє близько 20 ТБ очищених текстових даних щомісяця. На жаль, більша частина результуючого тексту не є природною мовою. Натомість він здебільшого складається з шаблонних текстів, таких як меню, повідомлення про помилки або тексти, що повторюються. У зв'язку з цим автори оригінальної статті проводять ретельний процес попереднього опрацювання, що включає в себе:

– видалення дублікатів;

– видалення нецензурних слів;

– видалення текстів, що належать до програмного коду;

– відбір текстів англійською мовою (за допомогою бібліотеки langdetect);

– видалення текстів з повідомленнями про помилки Javascript та іншими стандартними кодами помилок;

– відбір речень завдовжки понад 3 слова і текстів, що містять понад 5 речень.

Щоб зібрати базовий набір даних, було завантажено Common Crawl текст за квітень 2019 року з подальшим застосуванням описаної вище процедури фільтрації. У результаті виходить набір тексту, який не тільки на кілька порядків більший за більшість наборів даних, що використовуються для попереднього навчання (приблизно 750 ГБ), а й також містить досить чистий і природний текст.

Використання цієї системи збирання даних у сукупності з однією з найкращих архітектур навчання seq2seq моделей дало змогу авторам домогтися найвищих результатів на кількох бенчмарках, зокрема отримати результат, максимально наближений до людського, у системі оцінювання якості моделей розуміння мови SuperGLUE [27].

Основні переваги моделі:

– уніфікована структура для всіх seq2seq моделей;

– великий і якісний корпус, який використовується для попереднього навчання;

– зручна структура тюнінгу та отримання результатів моделі.

Ключові недоліки:

– модель не отримує найкращих результатів у завданні перекладу, тобто якість моделі певною мірою залежить від мови тексту;

– у разі завдання класифікації, наприклад, аналізу тональності, від моделі очікують виведення одного з варіантів (наприклад, «позитивний» або «негативний»), однак трапляються випадки, коли модель виводить виведення не з-поміж варіантів навчальної вибірки (такі випадки будуть помилкою моделі з огляду на її універсальну seq2sqe-структуру).

2.4.3 BERT

Однак архітектура кодер-декодер-трансформер – не єдиний варіант. У BERT (рисунок 2.6) використовується архітектура лише для кодерів, тоді як більшість сучасних моделей великих мов (LLM) базуються на трансформаторах, що працюють лише на декодерах.

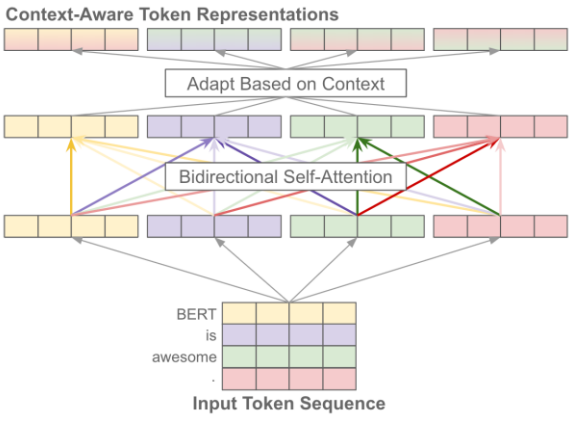


Рисунок 2.6 – Загальна архітектура BERT

Операція самоуваги бере на вхід послідовність лексем-векторів і створює на виході нову послідовність перетворених лексем-векторів тієї самої довжини, що й на виході; див. вище. Кожен запис цієї нової послідовності є середньозваженим середнім значенням векторів у вхідній послідовності. Зокрема, ми обчислюємо кожен лексемний вектор у вихідній послідовності наступним чином, де y_i та x_j – елементи вихідної та вхідної послідовностей відповідно:

$$y_i = \sum_j w_{i,j} x_j, \tag{2.12}$$

Вага w_{i, j}, наведена вище, є оцінкою уваги, яка створюється як функція x_i та x_j. Простіше кажучи, ця оцінка фіксує, наскільки поточний лексем повинен «звертати увагу» на інший лексем у послідовності під час обчислення його нового представлення (рисунок 2.7).

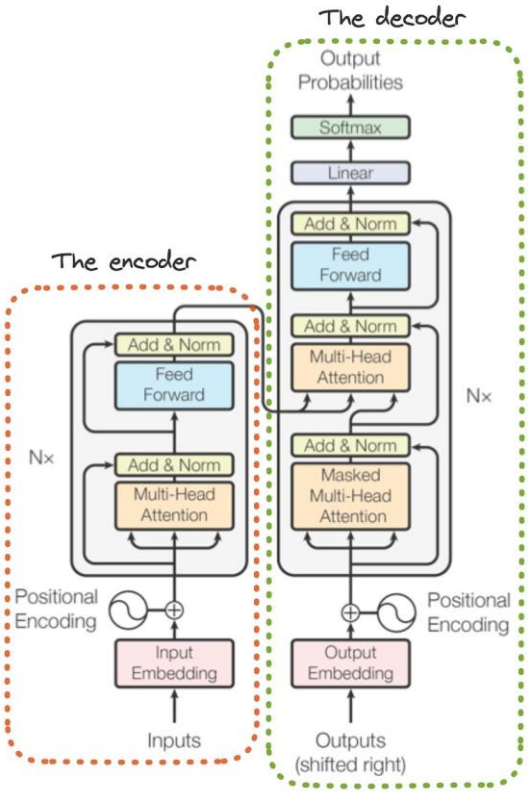


Рисунок 2.7 – Архітектура декодеру

Оригінальна архітектура трансформатора використовує два «стеки» трансформаторних шарів; див. вище. Перший стек (модуль кодера) складається з декількох блоків, які містять двонаправлену самоувагу та нейронну мережу з прямим спрямуванням. Другий стек (модуль декодера) досить схожий, але використовує масковану самоувагу, а також має додатковий механізм «перехресної уваги», який під час самоуваги враховує активації у відповідному кодерному шарі. Трансформатор спочатку використовувався для завдань «послідовність-послідовність» (наприклад, для перекладу мови), приклад на рисунку 2.8. Для інших завдань популярними стали одностекові трансформаторні моделі:

– у мовних моделях використовується архітектура лише декодера;

– моделі в стилі BERT використовують архітектуру, орієнтовану тільки на кодер.

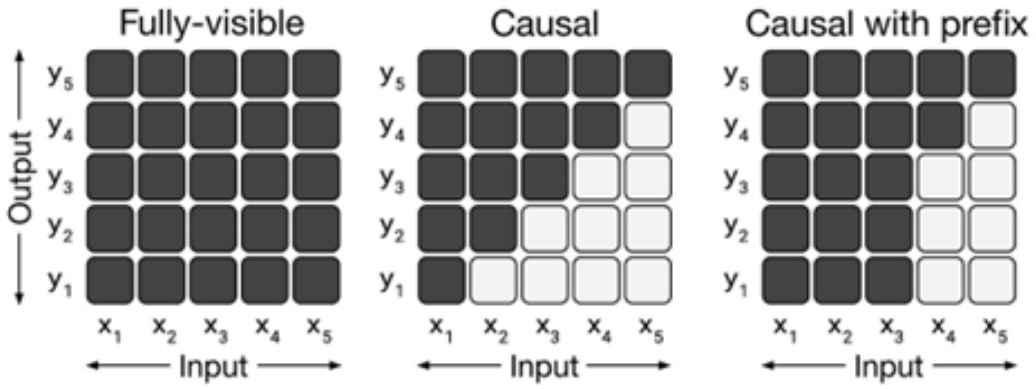


Рисунок 2.8 – Маски уваги

Атенційні маски. Різновиди архітектури трансформерів мають одну головну відмінність – тип маскування, що використовується в їхніх шарах уваги. Тут під словом «маскування» мається на увазі маскування (або ігнорування) певних лексем під час обчислення самоуваги. Простіше кажучи, певні лексеми можуть дивитися лише на вибрану частину інших

лексем у повній вхідній послідовності. На малюнку вище показано різні варіанти маскування для самоуваги.

Моделі, орієнтовані лише на кодер, використовують двонаправлену (або повністю видиму) самоувагу, яка під час самоуваги розглядає всі маркери в межах усієї послідовності. Кожне представлення лексеми в самоуважності обчислюється як середньозважене середнє значення всіх інших лексем у послідовності. На противагу цьому, в моделях, орієнтованих лише на декодер, використовується причинно-наслідкова самоуважність, коли кожна лексема враховує лише лексеми, які приходять до неї в послідовності, приклад на рисунку 2.9.

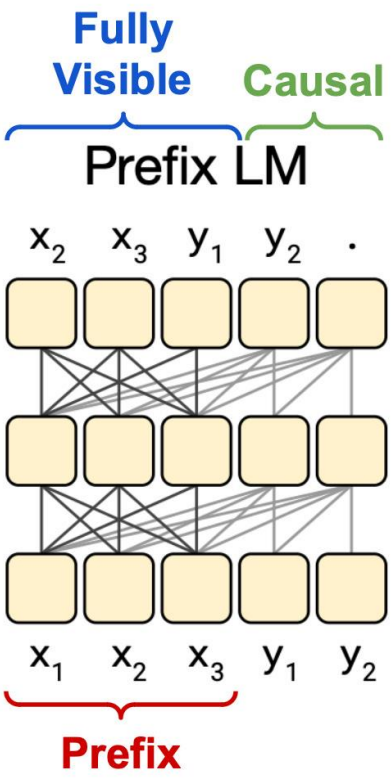


Рисунок 2.9 – Модель двонаправленої уваги

Ми також можемо застосувати гібридний підхід, визначивши «префікс». Більш конкретно, ми можемо виконувати двонаправлену самоувагу для групи лексем на початку послідовності (тобто

префікса), а потім виконувати причинно-наслідкову самоувагу для решти лексем послідовності. Повністю видима (або двонаправлена) самоувага корисна для уваги над префіксом або виконання завдань класифікації. Однак певні програми (наприклад, мовне моделювання) вимагають причинно-наслідкової самоуваги під час навчання, щоб запобігти тому, щоб трансформер не міг «зазирнути в майбутнє» (тобто просто скопіювати правильний лексем під час генерування вихідних даних).

Хоча в аналізі в [1] розглянуто багато трансформаторних архітектур, первинною моделлю, використаною для T5, є стандартна архітектура «кодер-декодер». За винятком кількох невеликих модифікацій, ця модель досить схожа на трансформатор у тому вигляді, в якому він був запропонований спочатку. Архітектури, призначені лише для кодерів, не розглядаються в [1], оскільки вони призначені для класифікації на рівні маркерів або послідовностей, а не для генеративних завдань, таких як переклад або узагальнення. T5 має на меті знайти уніфікований підхід (заснований на трансферному навчанні) для вирішення багатьох завдань з розуміння мови.

2.5 Нейромережеві моделі сумаризації тексту

2.5.1 Рекурентна нейронна мережа

Рекурентна нейронна мережа (RNN) – це різновид штучної нейронної мережі, що використовується в NLP. RNN спроектована таким чином, щоб виставляти послідовні входи даних. Впроваджуючи архітектуру зі схованим станомh(t) і петлями, цей тип нейронних мереж дозволяє зберігати раніше оброблену інформацію. Загальна візуалізація такої моделі представлена на рисунку 2.10.

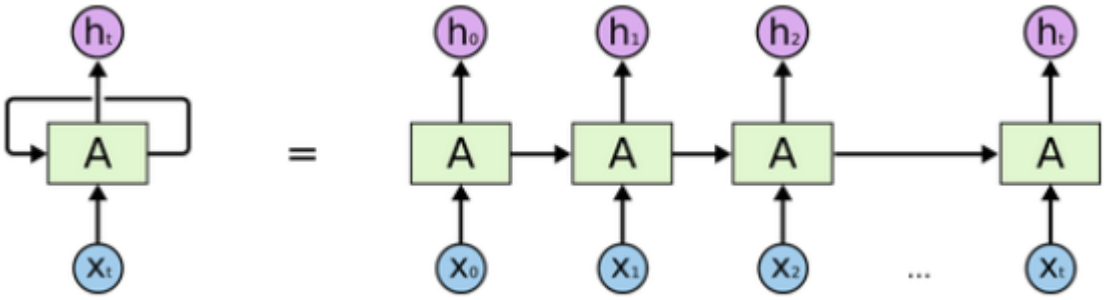


Рисунок 2.10 – Модель рекурентної нейронної мережі

RNN є потужним і надійним типом нейронної мережі і належать до найперспективніших алгоритмів, які використовуються, оскільки вони єдині з внутрішньою пам'яттю.

Як і багато інших алгоритмів глибокого навчання, рекурентні нейронні мережі відносно старі. Спочатку вони були створені в 1980-х роках, але лише в останні роки ми побачили їхній справжній потенціал. Збільшення обчислювальної потужності разом із величезними обсягами даних, з якими нам зараз доводиться працювати, і винахід довготривалої короткострокової пам'яті (LSTM) у 1990-х роках дійсно висунули RNN на перший план.

Завдяки своїй внутрішній пам'яті RNN можуть запам'ятовувати важливі речі про отримані дані, що дає їм змогу бути дуже точним у прогнозуванні того, що буде далі. Ось чому вони є найкращим алгоритмом для послідовних даних, таких як часові ряди, мова, текст, фінансові дані, аудіо, відео, погода та багато іншого. Рекурентні нейронні мережі можуть сформувати набагато глибше розуміння послідовності та її контексту порівняно з іншими алгоритмами.

RNN і нейронні мережі прямого зв'язку отримують свої назви від того, як вони передають інформацію.

У нейронній мережі з прямим зв'язком інформація рухається лише в одному напрямку – від вхідного шару через приховані шари до вихідного шару. Інформація переміщується прямо через мережу і ніколи не торкається вузла двічі.

Нейронні мережі з прямим зв'язком не пам'ятають вхідні дані, які вони отримують, і погано передбачають, що буде далі. Оскільки мережа прямого зв'язку враховує лише поточний вхід, вона не має поняття про порядок у часі. Вона просто не може згадати нічого про те, що відбувалося в минулому, крім свого навчання.

У RNN інформація проходить по циклу. Коли він приймає рішення, він враховує поточні вхідні дані, а також те, що він дізнався з вхідних даних, які він отримав раніше.

Таким чином, прямий перехід RNN може бути представлений наведеним нижче набором рівнянь:

$$a^{(t)} = b = Wh^{(t-1)} + Ux^{t}, \tag{2.13}$$

$$h^{(t)} = \tanh(a^{(t)}), \tag{2.14}$$

$$o^{(t)} = c + Vh^{(t)}, \tag{2.15}$$

$$\hat{y}^{(t)} = softmax(o^{(t)}). \tag{2.16}$$

Це приклад рекурентної мережі, яка відображає вхідну послідовність у вихідну послідовність такої ж довжини. Загальна втрата для даної послідовності значень x у парі з послідовністю значень y тоді буде просто сумою втрат за всі кроки часу. Ми припускаємо, що вихідні дані o(t) використовуються як аргумент функції softmax для отримання вектора ŷ ймовірностей над виходом. Ми також припускаємо, що втрата L є від'ємною

логарифмічною ймовірністю справжньої цілі y(t) з урахуванням вхідних даних.

2.5.2 Мережі довготривалої короткочасної пам'яті

Довготривала короткострокова пам'ять (Long Short Term Memory, LSTM) – це закрита RNN, розроблена Хохрейтером і Шмідхубером (Hochreiterand Schmidhuber, 1997), яка дозволяє пом'якшити проблеми зникаючих і вибухаючих градієнтів для довготривалих залежностей під час фази зворотного розповсюдження. Осередок пам'яті, який замінює прихований блок у ванільних RNN, об'єднує чотири головні елементи: вхідні ворота, забудьте ворота, вихідні ворота та самовідновлюваний рекурентний зв'язок, це відображено на рисунку 2.11.

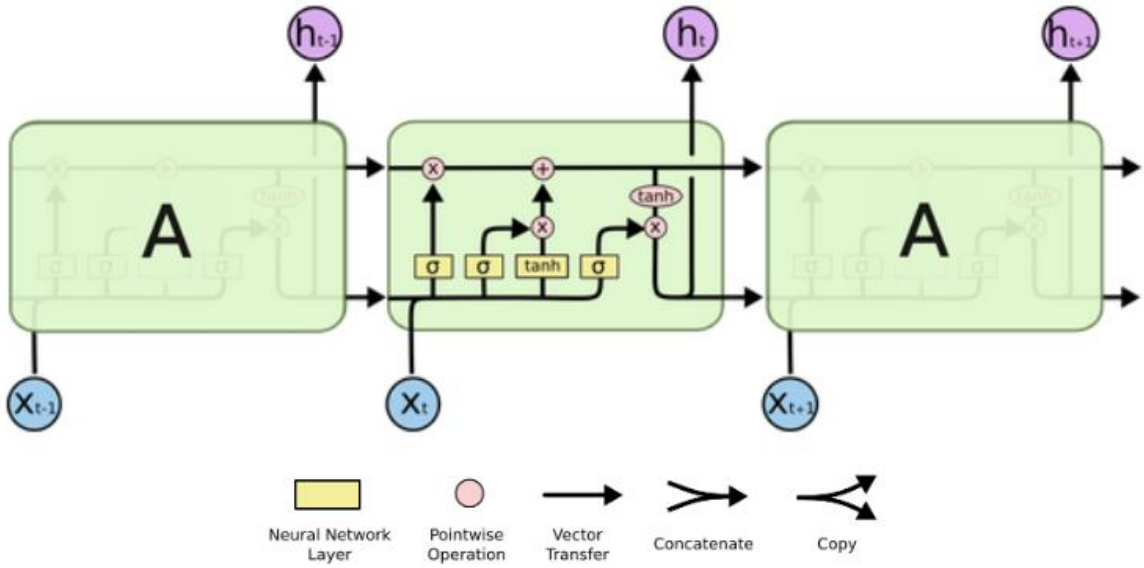


Рисунок 2.11 – Модель LSTM

Наступні рівняння визначають пряме поширення для кожного часового кроку t:

$$i^t = \sigma\left(W^{(i)}x^{(t)} + U^{(i)}h^{(t-1)} + b^{(i)}\right), \quad (2.17)$$

$$f^{(t)} = \sigma\left(W^{(f)}x^{(t)} + U^{(f)}h^{(t-1)} + b^{(f)}\right), \quad (2.18)$$

$$o^{(t)} = \sigma\left(W^{(o)}x^{(t)} + U^{(o)}h^{(t-1)} + b^{(o)}\right), \quad (2.19)$$

$$\widehat{c^{(t)}} = tanh\left(W^{(c)}x^{(t)} + U^{(c)}h^{(t-1)} + b^{(c)}\right), \quad (2.20)$$

$$c^{(t)} = tanh\left(f^{(t)}o\ c^{(t-1)} + i^{(t)}o\ c^{(t)}\right), \quad (2.21)$$

$$h^{(t)} = o^{(t)}o\ \tanh\left(c^{(t)}\right), \quad (2.22)$$

де x^t – вхідний вектор;

h^t – оновлений прихований стан на кроці часу t;

Σ – функція нелінійності.

Вектори зміщення $b^{(i)}$, $b^{(f)}$, $b^{(o)}$ і $b^{(c)}$ та вагові матриці $U^{(i)}$, $U^{(f)}$, $U^{(o)}$, $U^{(i)}$, $W^{(i)}$, $W^{(f)}$, $W^{(o)}$ і $W^{(c)}$ – параметри моделі.

LSTM лежать в основі еволюції мовного моделювання. Незважаючи на те, що вони розв'язують проблему зникаючого градієнта, проблема вибухаючих градієнтів все ще переважає. Незважаючи на запровадження таких методик, як градієнтне відсікання (Pascanu, Mikolov, and Bengio, 2013) і випадаюче-регулювання (Dropout-регуляризація) (Заремба, Суцкевер, і Вінялс, 2014), їх, мабуть, недостатньо, щоб адекватно розв'язати цю проблему. З точки зору мовного моделювання запропоновано багато варіантів базового LSTM. Зокрема, експерименти з воротами (Melis, Kociský, Blunsom, 2020) та інкорпоруванням багатоклітинних LSTM – Multi-cell LSTM (Cherian, Badola, Padmanabhan, 2018).

Наразі найкращий показник perplexity 2,20 на бенчмарках PTB та WikiText-2 досягається модифікацією LSTM Mogrifier (Melis, Koˇcisk ́y, and

Blunsom,2020). Отже, навіть після появи трансформерів поважні LSTM все ще можна вважати найсучаснішими.

2.5.3 Закритий рекурентний блок

Gated Recurrent Unit (GRU) – це простіша версія LSTM, яку представили Cho et al., 2014. GRU має лише двоє воріт (скидання й оновлення) і не має комірки пам'яті, а рівняння 2.23-2.26 описують, як клітинка GRU оновлюється за кожним часовим кроком t.

$$z^{(t)} = \sigma\left(W^{(z)}x^{(t)} + U^{(z)}h^{(t-1)} + b^{(z)}\right), \quad (2.23)$$

$$r^{(t)} = \sigma\left(W^{(r)}x^{(t)} + U^{(r)}h^{(t-1)} + b^{(r)}\right), \quad (2.24)$$

$$\widehat{h^t} = \tanh\left(r^{(t)} o U^{(h)}h^{(t-1)} + W^{(h)}x^{(t)} + b^{(h)}\right), \quad (2.25)$$

$$h^{(t)} = \left(1 - z^{(t)}\right) o \widehat{h^{(t)}} + z^{(t)} o h^{(t-1)}. \quad (2.26)$$

Вектори зміщення $b^{(z)}$, $b^{(r)}$, $b^{(h)}$ та вагові матриці $U^{(z)}$, $U^{(r)}$, $U^{(h)}$, $W^{(z)}$, $W^{(r)}$, та $W^{(h)}$, є параметрами моделі. Σ – функція нелінійності. GRU менш відома, ніж LSTM, але, маючи меншу кількість параметрів, демонструє порівняно хороші результати при розпізнаванні мовлення LM Irie et al., 2016 та послідовному генеруванні даних Chung et al., 2014, приклад на рисунку 2.12.

Традиційно продуктивність LM вимірюється такими внутрішніми метриками, як перплекситивність, крос-ентропія та біт на символ (BPC). Усі ці три показники сильно корелюють між собою.

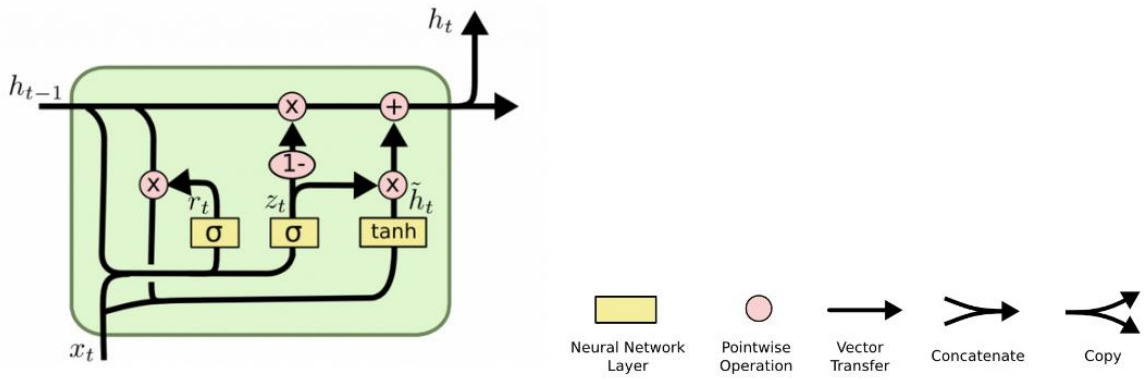


Рисунок 2.12 – Схематична модель GRU

Але не завжди доцільно порівнювати їх між ЛМ з різним розміром словникового запасу, мовними одиницями та оперуванням OOV-словами. Теорія походить від оцінки «ентропії» англійської мови, запропонованої Шенноном (Shannon, 1948), що вимірює середній обсяг інформації, переданої в повідомленні. Мовна модель має на меті дізнатися невідомий розподіл мовних одиниць у мові за зразком тексту. Таким чином, мінімізуючи крос-ентропію H(P,Q) засвоєного розподілу Q по відношенню до емпіричного розподілу P мови.

2.6 Основні підходи до сумаризації тексту

2.6.1 Екстрактивна сумаризація

Процес екстрактивного узагальнення передбачає відбір найбільш релевантних речень зі статті та їх систематичну організацію. Речення, що складають резюме, беруться дослівно з вихідного матеріалу. Системи екстрактивного узагальнення в тому вигляді, в якому ми їх знаємо зараз, базуються на трьох фундаментальних операціях.

Побудова проміжного представлення вхідного тексту.

Прикладами методів, що базуються на репрезентаціях, є репрезентація тем і репрезентація індикаторів. Щоб зрозуміти згадану в тексті тему, тематичне представлення перетворює текст на проміжне представлення.

За допомогою цього методу можна знайти у вхідному документі терміни, пов'язані з темою. Значимість речення можна обчислити двома способами: по-перше, як функцію кількості тематичних підписів, які в ньому містяться; по-друге, як частку тематичних підписів, які в ньому містяться.

Частотно-залежні підходи. За допомогою цього методу словам надається відносна важливість. Якщо термін відповідає темі, він отримує 1 бал, інакше – дорівнює нулю. Залежно від способу реалізації ваги можуть бути безперервними. Тематичні репрезентації можуть бути досягнуті за допомогою одного з двох методів:

– ймовірність слова: Для того, щоб визначити його значущість, потрібна лише частота слова. Щоб обчислити ймовірність слова w, ділимо частоту, з якою воно трапляється, f (w) на загальну кількість слів N;

$$P(w) = \frac{f(w)}{N}. \quad (2.27)$$

Середнє значення значущості слів у реченні дає важливість речення при використанні ймовірностей слів.

– TFIDF (Term Frequency Inverse Document Frequency): цей метод є вдосконаленням словесно-імовірнісного підходу. Тут вагові коефіцієнти визначаються за допомогою підходу TF-IDF. Методика Term Frequency Inverse Document Frequency (TFIDF) надає меншого значення термінам, які часто зустрічаються в більшості документів. Вага кожного слова w у документі d обчислюється так:

$$q(w) = f_d(w) * log \frac{|D|}{f_D(w)}, \quad (2.28)$$

де f_d (w) – частота терміна слова w у документі d;

f_d (w) – кількість документів, які містять слово w;

а |D| – кількість документів у колекції D.

Латентно-семантичний аналіз. Латентно-семантичний аналіз (LSA) – це непідконтрольний метод вилучення репрезентації семантики тексту, що ґрунтується на спостережуваних словах. Процес LSA починається з побудови матриці термін-умова (n на m), де кожен рядок являє собою слово з вхідних даних (n-слів), а кожен стовпчик – речення (m речень). У матриці вага слова і у реченні j визначається записом aij. За методикою TFIDF кожному слову в реченні присвоюється певна вага, причому нульовий показник присвоюється термінам, які не входять до речення.

Оцінювання речень на основі репрезентації.

Під час генерування проміжної репрезентації кожному реченню присвоюється оцінка значущості. У разі використання методу, який ґрунтується на тематичному репрезентативному представленні, оцінка речення відображає, наскільки ефективно воно розкриває критичні концепції в тексті. У репрезентації індикаторів оцінка обчислюється шляхом агрегування доказів від різних зважених індикаторів.

Підбір резюме, що складається з кількох речень.

Щоб сформувати резюме, програмне забезпечення для узагальнення вибирає найкращі k речень. Наприклад, деякі методи використовують жадібні алгоритми, щоб вибрати найбільш релевантні речення, тоді як інші можуть перетворити відбір речень на оптимізаційну задачу, коли набір речень відбирають з таким умовою, щоб максимізувати загальну важливість і зв'язність і при цьому звести до мінімуму кількість надлишкової інформації (рисунок 2.13).

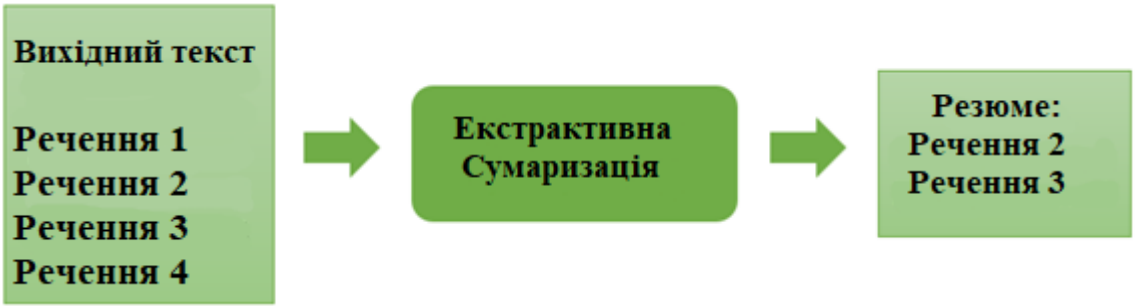


Рисунок 2.13 – Процес екстрактивної сумаризації

2.6.2 Абстрактивна сумаризація

На противагу екстрактивному узагальненню, абстрактне узагальнення є більш ефективним методом. Здатність створювати унікальні речення, які передають життєво важливу інформацію з текстових джерел, сприяла зростанню популярності цього методу. Абстрактне узагальнення подає матеріал у логічному, добре організованому та граматично грамотному вигляді. Якість резюме можна суттєво підвищити, зробивши його більш читабельним або покращивши його лінгвістичну якість. (включити зображення).

Існує два підходи: Структурований підхід та Семантичний підхід.

Структурно-орієнтований підхід – у структурно-першочерговому методі найбільш важлива інформація з документу (документів) кодується за допомогою психологічних схем ознак, таких як шаблони, правила видобування, а також альтернативні структури, зокрема дерев'яну, онтологічну, основну і основну, правильну, а також структуру на основі графів. Далі ми прочитаємо про дПМкі з кількох методів, які інтегровані в цю стратегію (рисунок 2.14).

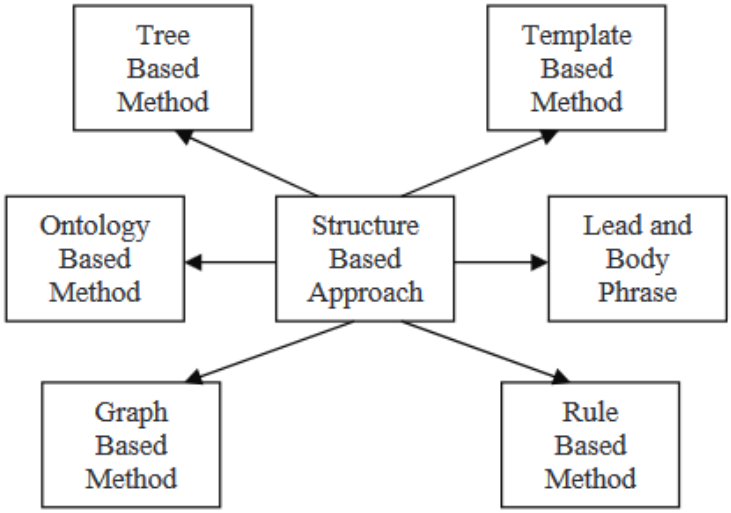


Рисунок 2.14 – Схема структурно-орієнтованого підходу

Деревоподібні методи – у цьому методі вміст документа представляється у вигляді дерева залежностей. Відбір контенту для конспекту можна здійснити за допомогою кількох інших методів, як-от алгоритмічна програма для перетину тем або програма, що використовує нативне вирівнювання у проаналізованих реченнях. У цьому підході для генерації конспектів використовується або генератор мов, або алгоритм асоційованого ступеня.

У цьому методі в якості вхідних даних використовується набір документів, які обробляються за допомогою алгоритму відбору тем для вилучення центральної теми, а потім за допомогою алгоритму кластеризації ранжируються фрази за ступенем важливості, приклад на рисунку 2.15. Структурований метод кодує найважливіші дані з документа (документів), використовуючи психологічні схеми ознак, такі як шаблони, правила видобутку, а також альтернативні структури, як-от: дерево, онтологія, ведуча і основна частина, правило, а також графові структури.

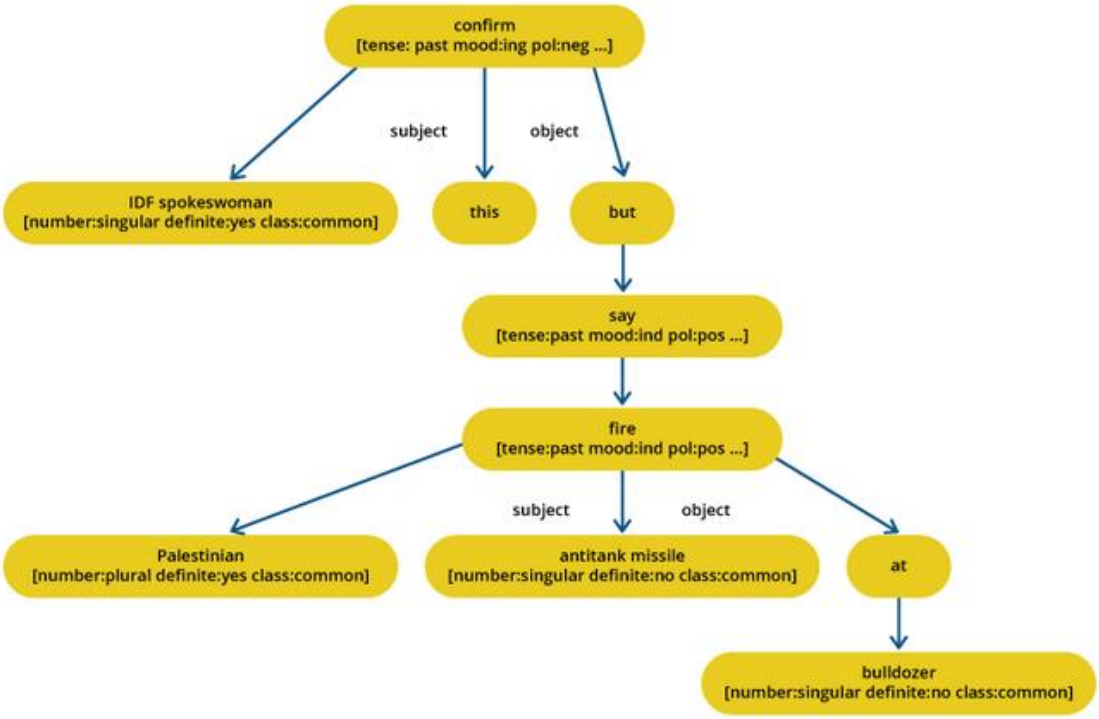


Рисунок 2.15 – Деревоподібний метод

Методи на основі шаблонів – у цьому методі використовується путівник, який представляє весь документ. Лінгвістичні закономірності або критерії вилучення порівнюються з метою виявлення текстових фрагментів, які можна зіставити зі слотами путівника. Ці текстові фрагменти є індикаторами одиниці площі вмісту контуру, приклад на рисунку 2.16.

Реалізована для вилучення інформації, GISTEXTER – це система узагальнення, яка ідентифікує інформацію, пов'язану з темою, у вхідному тексті та перетворює її на записи в базі даних, після чого речення додаються до зведеної інформації залежно від запитів користувачів.

Методи, засновані на онтологіях – багато дослідників намагалися підвищити ефективність резюме, використовуючи онтологію (базу знань). Більшість інтернет-документів мають спільний домен, що означає, що всі вони стосуються однієї і тієї ж загальної теми. Онтологія – це потужне представлення унікальної інформаційної структури кожного домену. У цій роботі пропонується використовувати нечітку онтологію, яка моделює

невизначеність і точно описує доменні знання, для узагальнення китайських новин. У цьому методі експерти-доменники спочатку визначають онтологію домену для новинних подій, а потім на етапі підготовки документів витягують семантичні слова з корпусу новин і китайського словника новин.

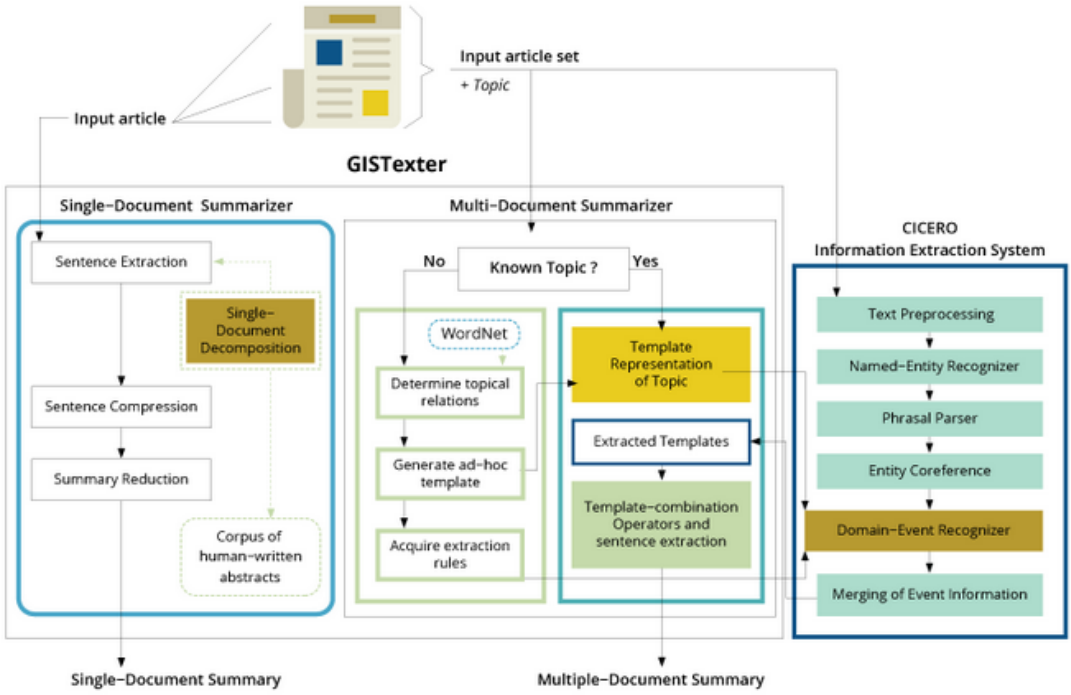


Рисунок 2.16 – Методи на основі шаблонів

Метод головної та тілесної фраз – цей підхід передбачає переписування головного речення, виконуючи операції над фразами (вставка та підстановка), у яких у головній та тілевій частинах речення є однакові синтаксичні головні фрагменти.

3 РЕАЛІЗАЦІЯ ЗАДАЧІ СУМАРИЗАЦІЇ ТЕКСТУ УКРАЇНСЬКОЮ МОВОЮ

3.1 Особливості налаштування системи для української мови

Маючи всю вищезазначену роботу, виконану для англійської мови, передбачувано, що підходи, які демонструють хороші результати, будуть порівняно добре працювати і для інших мов. Дійсно, як зазначають Cotterell (2018), «Більшість методів переносні в наступному розумінні: за умови наявності належним чином аннотованих даних їх, в принципі, можна тренувати на будь-якій мові. Однак, незважаючи на цю грубу крос-лінгвістичну сумісність, малоймовірно, що всі мови однаково легкі або що наші методи однаково добре працюють на всіх мовах».

З точки зору статистичного мовного моделювання є три важливі відмінності між українською та англійською мовами.

По-перше, українська – слов'янська мова, яка славиться багатими флексіями. Як зазначає Павлюк (2018), кількість флексій в українській мові значно перевищує їхню кількість в англійській, оскільки кожна номінальна частина мови має безліч закінчень. Ці закінчення виражають відмінок і рід іменних частин мови (іменників, прикметників, числівників, займенників), а також час, аспект, особу, число, глас і настрій дієслів. Також в українській мові багато частин мови можуть утворювати зменшувальні форми слова, тоді як в англійській мові така можливість є лише в іменників.

По-друге, у більшості узгоджувальних словосполучень (коли форма головного слова визначає форму залежного слова) в англійській мові немає узгодження. Натомість в українській мові залежні слова узгоджуються за числом, родом і відмінками, якщо головним словом є іменник, а сполучними – прикметники та займенники.

По-третє, українські слова часто демонструють чіткіші морфологічні закономірності, ніж англійські слова. Спрощену модель слова можна

визначити як основу, що відповідає за лексичне значення, і флексію, прикріплену до основи. В ідеалізованому прикладі українських слів, основа – це, по суті, складова комбінація, що складається з трьох головних частин: корінь, який відповідає за основне лексичне значення слова, до якого можуть бути приєднані нульовий або більший похідний префікс (префіксів) і нульовий або більший суфікс (суфіксів). Наявність цих афіксів, як правило, робить внесок у зміну лексичного значення кореня.

По-перше, англійська мова накладає жорсткі обмеження на відносний порядок слів у реченні. Натомість в українській мові суб'єкт і об'єкт речення можуть визначатися не тільки порядком самих слів, але й флексією слова, а також його узгодженням із дієсловом.

Ці відмінності суттєво вплинули на постановку проблеми дослідження та подальше ухвалення рішення щодо вибору набору експериментів.

3.2 Дані для аналізу

Як зазначають Jozefowicz et al., 2016: «Моделі, які здатні точно розміщувати розподіли над реченнями, кодують не лише складнощі мови, як-от граматичні структури, але й розподіляють чималу кількість інформації щодо знань, які можуть міститися в корпусі».

Таким чином, для нашого дослідження було дуже важливим скласти корпус, який можна порівняти з тими, що представлені для англійської мови, який би добре репрезентував українську лексику та граматичні закономірності.

Як правило, для data-science-завдань наявні дані ділять на навчальні, валідаційні та тестові розділи відповідно до частки 90%, 5% і 5% відповідно. Щоб іншим дослідникам було простіше порівнювати свої результати з нашими, ми вирішили обрати в якості базового тестового набору добре збалансований український корпус Брауна з відкритим вихідним кодом.

Щодо навчальної частини, то через брак таких даних ми не могли відповідати тим самим обмеженням і вимогам щодо різноманітності, а також щодо добре сформованого та вичитаного масиву даних.

Перейдемо до тренувальних даних із ресурсу «Korrespondent» це 2.1Gb даних завантажених з каналу новин «Korrespondent». Статті представляють соціальні, національні, політичні, наукові, спортивні, lifestyle&fashion, ділові, шоу-бізнес та інші новини. Крім прозових текстів, у ньому також містяться прогнози погоди, спортивні новини, кіно і театральні програми, включно з таблицями, порівняннями, переліками метрик та індексів. Ми попередньо обробили (описано далі в цій статті) цей масив даних вручну і створили корпус, достатній як навчальний набір для наших експериментів. Після препроцесування він складається з 8 640 598 речень, відокремлених за допомогою інструмента Tokenize Text від API NLP UK.Collection української художньої літератури (рисунок 3.1).

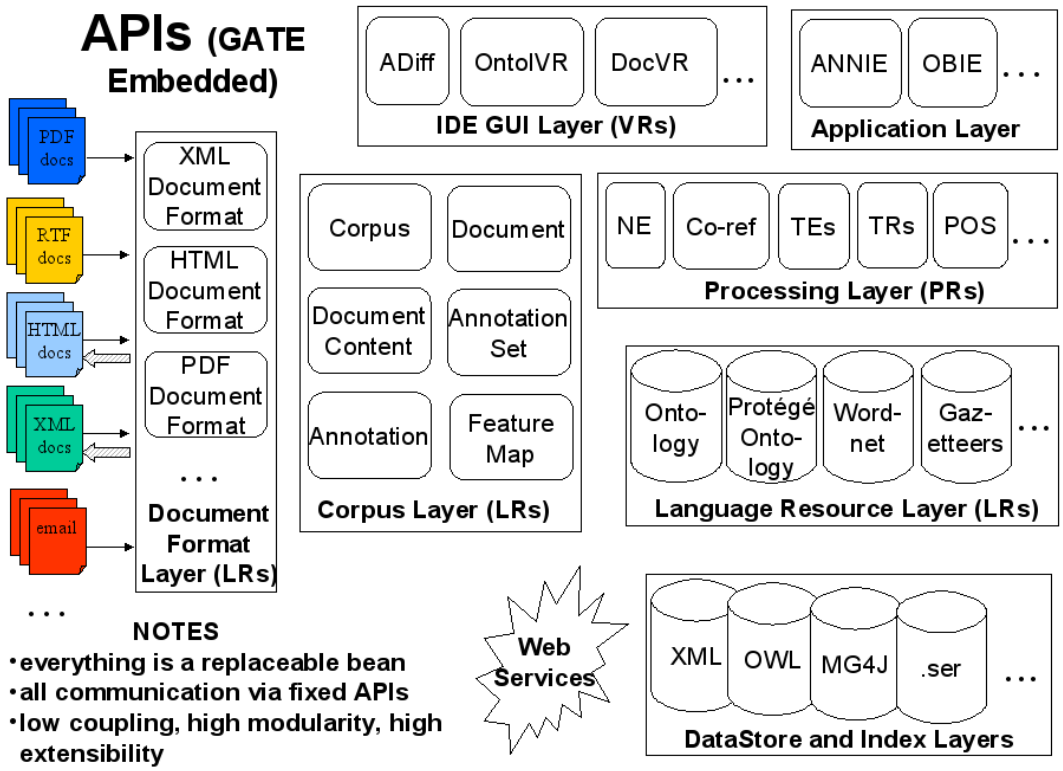


Рисунок 3.1 – API NLP UK

3.2.1 Виріб для оцінки Evaluation set

BrUk золотий стандартний тестовий корпус для всіх експериментів у цій магістерській дисертації ми обрали Український коричневий корпус BrUk – це добре збалансована та вивірена колекція оригінальних українських текстів із різних регіонів України. Ці тексти були опубліковані в період між 2010 і 2018 роками. Він містить лише прозові тексти, які не належать до жодної конкретної діалектної групи, і містить дані з таких сфер, як:

- новинні ЗМІ;
- релігійні ЗМІ;
- професійну літературу;
- естетично-інформаційна література;
- адміністративні документи;
- науково-популярна наука;
- наукова література;
- освітня література;
- написання художньої літератури.

Нами було проведено описовий аналіз «хорошої» (вичитані фрагменти без помилок) і «задовільний» (вичитані фрагменти з деякими незначними помилками) частин цього корпусу перед попередньою обробкою. Він складається з 924 текстів довжиною від 80 до 2195 слів (не лексеми).

Загалом він налічує 600810 слів і 38728 унікальних лем (лематизація здійснювалася за допомогою Language Tool API NLP UK), що відображено у таблиці 3.1.

Таблиця 3.1 – Аналіз відсоткового розподілу лемм

Домен	Відсоток у корпусі	Число слів	Число речень	Число унікальних лем
Новини	24%	144276	933	18293
Релігія	3.3%	19548	1220	4710
Проф.література	5.3%	32150	2393	7356
Художня літ.	7.9%	47491	2820	10858
Адмін.документи	2.0%	11974	563	2272
Наукого-поп.	5.4%	32545	2137	7404
Наукова літ.	11.3%	67690	3114	9252
Навчальна літ.	13.5%	81245	4913	11010
Рукописи	27.3%	163891	13527	21405

3.2.2 Попереднє опрацювання даних

Ми витратили значну кількість часу на попередню обробку та підготовку даних, особливо під час формування навчального набору. По-перше, ми видалили велику кількість украхнських та інших мовних цитат, діалогів і віршів або замінили їх на маркер _#foreign_ там, де вони були лише частиною речення. Для некириличних мов ми також замінювали окремі іноземні слова на цей лексем. З кириличними символами зробити це було складніше, тож наші тренувальні дані все ще містять невеликі вставки з неукраїнських мов.

По-друге, ми замислюємося над питанням заміни цифр. Щоб представити статистичний розподіл слів у мові, зазвичай цифри замінюються якимось токеном, а не включаються до словникового запасу окремо. Наприклад, корпус банку Penn Tree, будучи текстом-джерелом для новинних ЗМІ, частина «Корреспондента» в нашому корпусі містить багато цифр, що позначають певні табличні значення, списки метрик та індексів.

Ми вирішили замінити цифри, що позначають роки, дати, час, тегами _#year_, _#date_ та _#time_ відповідно. У письмовій мові цифри іноді пишуться замість числівників і в складі речення можуть виконувати ту саму функцію, що й іменник (кардинальне та збірне числівники), прикметник (порядкові та множительні числівники) і прислівники (множительні числівники та дистрибутивні числівники).

Числівники, що належать до дескриптивного класу слів, у словосполученнях з іншими частинами мови змінюють морфологічні закінчення. В українській мові числівники мають числівникову, відмінкову та частково родову категорії. Кардинальні числівники «один» і «два» мають три гендерні розрізнення, інші – загальну форму чоловічого і жіночого роду та індивідуальну форму середнього роду. Крім того, в урядовому типі словосполучень з іменниками числівники «два», «три», «чотири» набувають певної граматичної форми модифікованого слова (іменника), що є винятком зі схеми відмінних від схеми падежів, які регламентуються числівниками «п'ять» і вище.

Тому ми вирішили замінити цифри 1, 2, 3 і 4 там, де це було доречно, відповідним словом у правильній граматичній формі. Усі інші цифри ми замінили на лексеми _#number_ (нерозривні цифри замінювали на один лексем) і _#float_ відповідно.

Майже всі римські цифри також були замінені на маркер _#number_. Частина корпусу «Кореспондент» містила багато посилань на інші медіаджерела. 70 найпоширеніших з них ми замінили на маркер _#media_. Як і в будь-якій іншій попередній обробці тексту, ми також відшліфували навчальний і тестовий корпуси, видаливши зашумлені невідомі символи, спецсимволи, HTML-теги, а також перетворивши розділові знаки на єдину схему (рисунок 3.2).

За кордоном , з усіх дітей , які мають особливі потреби , від _#number_ до _#number_% залучено до інклюзивного навчання .
Україна має цей показник , за різними даними , від чотирьох до _#number_%.

Постановка проблеми Впровадження інклюзивного навчання в Україні розпочалось Канадсько-Українським проектом , який тривав протягом _#year_-_#year_ рр. , у _#year_ році розпочався всеукраїнський експериментальний проект з розроблення інклюзивного середовища , яким до _#year_ р. має бути охоплена уся Україна .

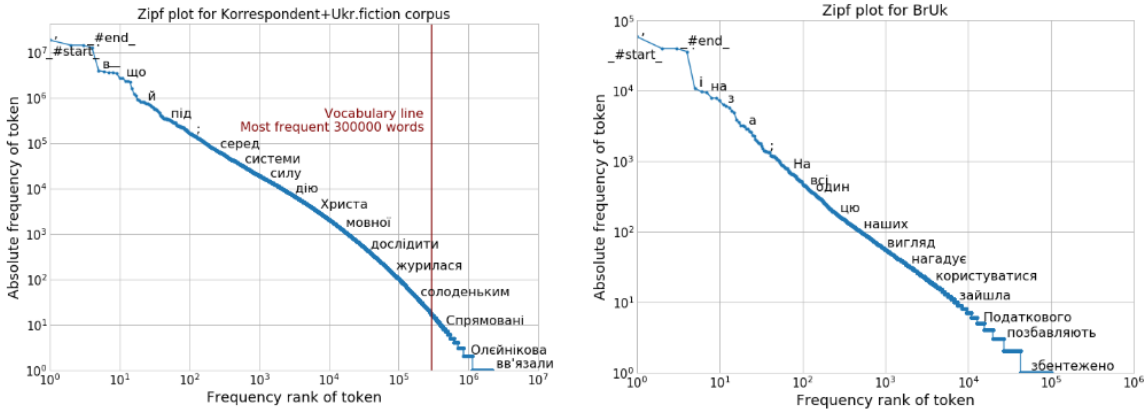
Дом'є колись сказав : "Людина — це жест ".

Він відчував , що усередині закипає лість .
"Марічко , не бійся , дитино , я тебе не здам , не здам ".
— Я не буду розповідати щось людині , яка не розуміє й крихти того , що відбувається .
— Чому ж не розумію , — посміхнувся військовий , — добре розумію .

До "Плуга "входили Дмитро Бедзик , Володимир Гжицький , Андрій Головко , Наталя Забіла , Василь Минко , Іван Сенченко , Павло Усенко та інші . Видавали журнал "_#media_ "(_#year_-_#year_) , а також часопис "Плуг "(_#year_-_#year_) .

Рисунок 3.2 – Приклад посилань на речення

На наступному рисунку 3.3 ми відтворили діаграму Зіпфіана на навчальному корпусі художньої літератури «Кореспондент+Українська художня література» та на українському корпусі «Браун».



(А) (В)

Рисунок 3.3 – Діаграма розподілення

Тут осями слугують log (порядок ранжування в таблиці частот) і log (частота лексем), а речення лекенізовано так, як було описано раніше. Так, окремими лексемами є слова, знаки пунктуації та спеціальні лексеми. Також ми вилучили речення, що перевищують 60 лексем (112 119 речень), щоб на наступних кроках навчати нейромережеві LM зі складними архітектурами.

Загальна кількість унікальних лексем у корпусі поїзда становить 2189477. Як стверджує закон Зіпфа і як видно з діаграми, в корпусі є кілька дуже високочастотних лексем, на які припадає більша частина лексем у тексті («_#start_», «-», «і» і т.п.), а також безліч низькочастотних слів (прізвища, помилкові слова й інші неординарні слова). Навчити нейронну мережу над усіма словами, які трапляються хоча б один раз, було б обчислювально неможливо, і ми вирішили зменшити словниковий запас до розміру 300000 найчастіших лексем.

3.3 Налаштування моделі рішення

3.3.1 KenLM

Kneser-Ney LM було розраховано як орієнтир для порівняння більш складних LM. Основна перевага модифікованого N-gram LM полягає в тому, що за порівняно невеликий проміжок часу він оцінює статистичний розподіл над усіма occurrences n-gram у навчальному тексті.

За допомогою бібліотеки KenLM з відкритим кодом і команди *bin/lmplz* оцінили файл ARPA на навчальному корпусі. Файл ARPA представляє собою модель і містить n-грами та їхню статистику. Потім за допомогою команди *bin/query Model.arpa < BrUk.txt* на тестових даних BrUk ми зробили запит до моделі, щоб обчислити недолугість.

Бібліотека KenLM автоматично включає лексеми <s>, </s> та <unk>, тож ми не додавали лексеми _#start_ та _#end_ до наших речень вручну.

Крім того, themodel не бореться з оцінкою довших речень, тож речення, довші за 60 лексем, не видалялися перед цією серією експериментів.

Обрізка – це методика, яка використовується для зменшення розміру файлу ARPA, без істотної погіршення моделі. Це параметр команди оцінювання в бібліотеці KenLM. Встановлюючи pruning = 0 0 1 1 1 1, ми вказуємо оцінювачу видаляти одиничні синглетони порядку 3 і вище, а встановлюючи pruning = 0 0 0 0 0 0 0 1, ми видаляємо лише одиничні синглетони порядку 6.

Ми експериментували з різними N-грамами та різною обрізкою , рисунок 3.4. Беручи до уваги компроміс між розміром збірки ARPA-файлів і складністю тестового масиву, пропонуємо розглядати 6-граммний KenLM з обрізаннями = 0 0 0 0 0 1 1 в якості базового LM для корпусу BrUk.

Kenlm N-gram	Pruning	Training time (minutes)	Size of ARPA file	Perplexity including OOV	Perplexity excluding OOV
3-gram	-	03m:55	6.4G	814.93	697.82
6-gram	0 1 1 1 1 1	07m:24	5.0G	871.97	749.91
6-gram	0 0 1 1 1 1	08m:45	6.0G	800.39	686.53
6-gram	0 0 0 0 1 1	11m:51	18.2G	748.81	641.11
6-gram	0 0 0 0 0 1	23m:25	29.4G	742.42	635.54
6-gram	-	40m:05	41.1G	736.83	630.69

Рисунок 3.4 – Результат розміру збірки

3.3.2 RNNLM

Наші рекурентні нейронні мережеві LM дотримуються загального шаблону:

– речення представлене у вигляді послідовності цілих чисел, де лексема u_i ідентифікується за її індексом у словниковому запасі V ($u_i \in \{1, ...N\}$,N=розмір словникового запасу);

– шар вбудовування кодує N цілих чисел у N унікальних векторів ознак розміруembedding;

– повністю зв'язаний шар застосовує лінійне перетворення, щоб отримати результуючий вектор розміром, що дорівнює розміру словника. Над результуючим вектором обчислюється softmax-функція, яка присвоює ймовірності того, що кожна лексема у словнику буде наступною лексемою в послідовності;

– функція втрат обчислюється за тією самою функцією, що й перплекситет.

Усі моделі навчали за допомогою бібліотеки машинного навчання PyTorch (версія 1.3.1).

Моделі навчалися на корпусі «Кореспондент+Українська художня література», сформованому з розміром словникового запасу 300000 і обрізаними реченнями, що перевищують 60 токенів. Речення оброблялися в модель пакетами розміром 40, упакованими за допомогою *torch.nn.utils.rnn.pack_sequence*. Перед кожною епохою речення випадковим чином перемішувалися, і відбиралася валідаційна множина (з 4%).

Перебирати по сітці різні набори гіперпараметрів моделей було б надзвичайно трудомістким процесом. Тому ми спочатку вибирали гіперпараметри випадковим чином, а потім на основі попередніх результатів. Крім моделей, наведених у описі ми також провели низку експериментів із моделями Simple RNN з різними параметрами відсікання, відпадання та швидкості навчання. *У* багатьох експериментах виникали складнощі *в f* наборах для тренування та валідації, як і в першій моделі – через зникаючі або вибухонебезпечні градієнти, що вибухають. Також архітектура бажаної моделі сильно обмежувалася доступними обчислювальними та пам'ятними ресурсами.

Мотивуючись Bojanowski et al., 2017, та їхніми результатами щодо мовного моделювання слов'янських мов, ми поставили собі за мету оцінити працездатність мішків символьних n-грам 2.1.4 на українській мові.

Попередньо навчені 300-вимірні вбудовані слова FastText для 157 мов (включно з українською) були отримані та оприлюднені Facebook AI Research у 2018 році (Grave et al., 2018). Модель тренували за допомогою CBOW (Mikolov et al., 2013) з позиційними вагами та пакетами символьних n-грамм. Публічний словник FastText включає 2 млн лексем. На жаль, при токенізації українських текстів було допущено кілька помилок (наприклад, багато слів взагалі не відокремлено, апострофи розглядаються як символи відокремлення і т.д.). Загалом, у лексиконі FastText було представлено 763578 унікальних лексем з нашого поїзного корпусу. Хоча вектори вбудовування FastText є сумою символьних n-граммних репрезентацій, а вектори для позасловесних слів можна легко отримати за допомогою інструментів FastText, у цьому експерименті ми вирішили використати лише словник розміром 300 тис. найпоширеніших попередньо підготовлених вбудованих FastText-векторних вбудовувань. Ми замінили всі лексеми OOV на лексему _#unknown_ і вирахували її вбудовування як середнє значення всіх векторів, які не використовуються в нашому словнику. Також ми вирахували вбудовування для лексеми _#number_ як середнє значення всіх числових лексем у словнику FastText.

Крім того, ми використовували 300 тис. попередньо навчених FastText-векторів замість шару вбудовування в нашій LSTM LM і заморозили його. За такого підходу ми можемо приблизно порівняти отримані результати з результатами попередньо навчених моделей, оскільки обидві підтримують точно такий самий набір лексем.

BPE поетапно замінює словник набором лексем таким чином, щоб загальна кількість лексем для кодування тексту була мінімізована. Ми розрахували сегментацію з різною кількістю злиттів BPE (Крок 2: 2.1.4), щоб побачити, як розмір кодованого корпусу залежить від кількості кроків

злиття BPE (рисунок 3.5). Розмір словникового запасу, сформульований кодуванням BPE, дорівнює кількості унікальних символів у тексті плюс кількість операцій злиття.

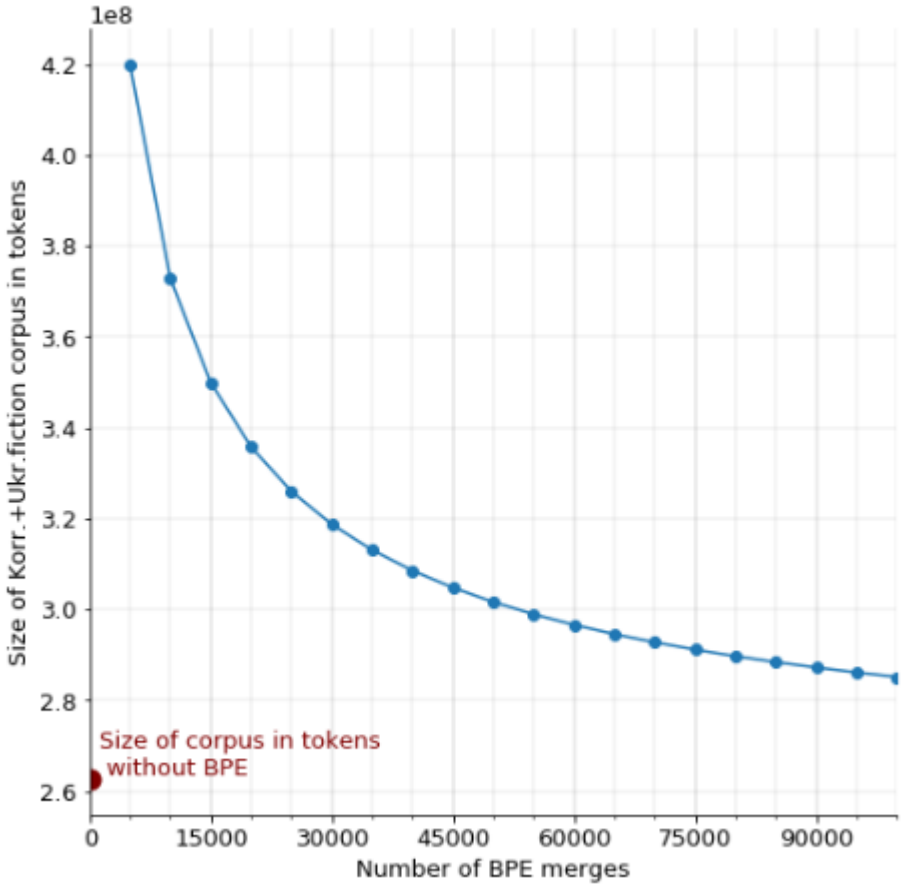


Рисунок 3.5 – Розмір словникового запасу

Для нашого завдання важлива не тільки кількість лексем для кодування тексту та розмір словника, а й значимість морфологічної інформації, закодованої в окремих відокремлених лексемах. У цьому експерименті ми хотіли спостерігати за впливом розбивки слів на послідовність кореня, афіксів і закінчення. Загалом БПЕ не придумано спеціально для цього. Він просто дає гарний баланс між об'ємом словникового запасу та ефективністю декодування. Але зазвичай часто зустрічаються частини слова, які виявляються афіксами. Тож із отриманих

словників вручну відбираємо той, у якому не надто багато цілих слів уже відібрано, але значна частина словникового запасу складається з невеликих частин слів. Потім округлюємо розмір цього словника до 20 тис. лексем.

Сегментацію ми реалізували за допомогою бібліотеки Google SentencePiece, яка також є офіційним кодом роботи Kudo and Richardson, 2018. SentencePiece задає остаточний розмір словника як вхідний параметр для алгоритму токенізації BPE (ми вибрали 20K). Крім того, інструмент резервує ідентифікатори словника для спеціальних метасимволів, які ми легко змінили на відповідні нам токени (невідомий символ _#unknown_, початок речення _#start_ і кінець речення _#end_). Ще однією чудовою особливістю SentencePiece є можливість обробляти користувацькі символи як один фрагмент у будь-якому контексті. Ми вказали наші спеціальні лексеми (_#foreign_, _#year_, _#date_, _#time_, _#media_, _#number_, _#float_) як користувацькі символи.

Далі ми просто кодуємо наш текст серією ідентифікаторів, що відповідають унікальним лексемам у згенерованому словниковому запасі, і подаємо упаковану послідовність в RNN так само, як це було зроблено для словесного рівня.

Ми вирішили обчислювати складність для LM на рівні підслів'я, нормалізуючи LM не за кількістю лексем у передбачуваній послідовності, а за кількістю слів, як було запропоновано у попередніх пунктах.

На рисунку 3.6, 3.7 наведено результати навчання RNN та методу екстрактивної сумаризації із сформульованим на основі BPE лексиконом із 20 тис. лексем і різними настройками гіперпараметрів.

Загалом, порівняно з RNN словесного рівня, значно зменшуються розмір моделі, кількість параметрів, що навчаються, і, як наслідок, вимоги до обчислювальних ресурсів. Незважаючи на те, що числові значення метрики складності вищі, ми не вважаємо, що це свідчить про те, що результати гірші.

```
  Input Words: міська влада виключає центрі міста явитися щонайменше підземні паркінги михайлівською європейською площами украї
нській правді київ недавно розповів радник мера києва павло рябікін очевидно хочуть <UNK> божевільну ідею відреагує громадськіс
ть рябікін бачить шляхи вирішення проблем паркінгами впорядкувати наявний паркувальний простір створити додаткові паркувальні п
отужності числі центрі міста способи зменшити кількість автівок радник кличка чомусь згадує насправді таких альтернатив уже наб
ереться добрий десяток платний їзд центр обмеження парними непарними номерними знаками ускладнення їзду великих авто наприклад
джипів пільги <UNK> електромобілів тощо різному вигляді практики чудово зарекомендували багатьох європейських містах лондоні па
рижі амстердамі берліні пекіні десятках міст авто парковок легальні законні паркінги стоять напівпорожні дурень захоче платити
стати тротуарі дивно влада <UNK> цілі площі центрі києва перетворюються стихійні парковки михайлівській софійській площі щодня
паркуються десятки автомобілів псують туристичну привабливість міста платять жодної гривні міський бюджет щойно новорічні гулян
ня михайлівській софійській площах закінчилися машини одразу повернулися місце теорія розбитих вікон дії являється ряд автівок
їхніх власників штрафують являються ряди д <UNK> виявиться простір розбитого шлагбауму понад місяць нікому діла софійська площа
4 лютого праворуч видно ряд якому автомобілів тиждень 12 лютого ряд уже заповнений фото новорічного дива софіївський майдан пар
ковки перетворився місце гуляють відпочивають люди новорічні свята єдиний площі використовують люди автівки скріншоти відео рад
іо свободи подібна ситуація михайлівській площі водії сприймають лінії утворює чорна плитка розмітку паркінгу <UNK> <UNK> лінію
середині ілюстрації влітку 2014 місяців обрання посаду кличко піарився михайлівській нема машин півтора такими дрібничками уже
ніхто <UNK> вибори

Summary
  Word Ids:       [39837, 90995, 4978, 16912, 8020, 17869]
  Response Words: ворожості паркові чоловіками пустого опустилися підказати
```

Рисунок 3.6 – Результат сумаризації тексту

```
Summarized Text:
 Чимало нинішніх керівників  підприємств – індустріальних монстрів часів СРСР – у своїй поведінці нагадують героя серіалу «Чорн
обиль» Анатолія Дятлова, інженера ЧАЕС
Один із сильних моментів фільму: діалог солдата й бабці, що не збирається залишати свою оселю та згадує усі трагедії, які пережи
ла Україна в XX столітті
```

Рисунок 3.7 – Результат сумаризації тексту

Передусім через дві важливі причини:

– модель рівня підслів'я є відкрито-словесною, і вона намагається передбачити з усього теоретично можливого набору слів. Таким чином, вона розкидає свою ймовірнісну масу тонко на майже нескінченну кількість випадків, на противагу порівняно невеликому розміру словесного словникового запасу (300 тис. К);

– якщо модель словесного рівня коректно передбачає, що вона раніше не бачила _#невідомої_ лексеми, то її збентеження зменшується, тоді як у моделі підсловесного рівня такої можливості немає, оскільки _#невідома_ лексема представляє собою окремі символи, які ніколи не траплялись у тренувальному корпусі. Але в добре складених корпусах таких символів не повинно існувати.

ВИСНОВКИ

У рамках цієї роботи було проведено порівняльний аналіз різних моделей сумаризації текстових даних і способів їхнього спільного застосування з методами визначення тональності. Було проведено ретельний аналіз сучасних підходів до вирішення завдань кодування текстових даних, сумаризації, тематичного моделювання та аналізу тональності. Додатково було проведено аналіз наявних комерційних інструментів, який показав, що методи, які використовують сервіси, часто є застарілими, ґрунтуються на евристичних підходах, підходах з використанням машинного навчання. При цьому аналіз літератури показує, що найякісніші результати для розв'язання всіх розглянутих завдань досягаються з використанням нейромережевих підходів, а саме з використанням архітектури Transformer і підходу трансферного навчання. Для розв'язання практичних завдань було розроблено архітектуру, що зачіпає всі основні етапи аналізу даних, починаючи від збирання та попереднього оброблення, закінчуючи візуалізацією отриманих результатів. Описується процес збору та первинного аналізу двох наборів даних, отриманих із різних соцмереж. Проведено низку практичних тестів, що показують приклади отримання значущих репрезентацій з вихідних корпусів. Як основні висновки можна відзначити:

– різні дані вимагають підбору моделей, краще пристосованих до роботи з тими чи іншими специфічними обмеженнями. Прикладами таких обмежень можуть бути;

– багатомовність вихідних даних, великий розмір корпусів, використання коротких текстів;

– методи кластерного аналізу, за правильного їх застосування разом із сучасними методами кодування даних, дають змогу отримувати якісні подання документів. Використання додаткових статистичних

прийомів дає змогу отримувати словникові репрезентації кластерів, приводячи задачу до вигляду, схожого із завданням тематичного моделювання.

Моделі мають низку ключових переваг перед методами тематичного моделювання:

– гнучка архітектура (можливість використання різних кодувань, включно з багатомовними), можливість автоматичного визначення кількості кластерів щільнісними алгоритмами;

– класичні та сучасні методи тематичного моделювання мають перед методами кластеризації низку переваг, основними з яких є міцне теоретичне підґрунтя та можливість кодування документів одночасно кількома темами.

При цьому перевага кількох тем на документ часто виявляється незатребуваною в завданні аналізу текстів, отриманих із соцмереж. У таких випадках достатнім виявляється використання однієї (найпопулярнішої) теми на повідомлення, зважаючи на короткий розмір текстів.

Підходи абстрактної сумаризації дають змогу отримувати якісно інші репрезентації текстових даних. Узагальнення, що генеруються такими методами, здатні якісно передавати сенс великої кількості постів і коментарів у стислій формі. Надалі такі репрезентації можуть бути використані фахівцями профільних галузей (соціологами, журналістами, політологами та маркетологами).

Методи аналізу тональності можуть використовуватися як на етапі агрегації даних (з подальшим аналізом негативних або позитивних записів), так і на етапі фіналізації для отримання додаткових візуальних уявлень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1) Aster R. C., Borchers B., Thurber C. H. Iterative Methods. *Parameter Estimation and Inverse Problems*. 2019. P. 151–179.

2) An automated essay evaluation system using natural language processing and sentiment analysi / V. S. Sadanand et al. *International Journal of Electrical and Computer Engineering (IJECE)*. 2022. Vol. 12, no. 6. P. 6585.

3) Attention based GRU-LSTM for software defect prediction / H. S. Munir et al. *PLOS ONE*. 2021. Vol. 16, no. 3. P. e0247444

4) Kalbfleisch J. D., Lawless J. F., Vollmer W. M. Estimation in Markov Models from Aggregate Data. *Biometrics*. 1983. Vol. 39, no. 4. P. 907.

5) Bousquet P.-M., Bonastre J.-F., Matrouf D. Exploring some limits of Gaussian PLDA modeling for i-vector distributions. *The Speaker and Language Recognition Workshop (Odyssey 2014)*. ISCA, 2014.

6) Text summarization using NLP / C. Varagantham et al. *International Journal Of Trendy Research In Engineering And Technology*. 2022. Vol. 06, no. 04. P. 26–30.

7) Papers with Code - Attention Is All You Need. The latest in Machine Learning | Papers With Code. URL: https://paperswithcode.com/paper/attention-is-all-you-need (date of access: 20.05.2024).

8) Semantic similarity metrics for evaluating source code summarization / S. Haque et al. *ICPC '22: 30th International Conference on Program Comprehension*, Virtual Event. New York, NY, USA, 2022.

9) ACL Anthology - ACL Anthology. URL: https://aclanthology.org/K19-1038.pdf (date of access: 22.05.2024).

10) Thesaurus for IR (IEKO). ISKO: International Society for Knowledge Organization. URL: https://www.isko.org/cyclo/thesaurus (date of access: 27.05.2024).

11) Сковпень Д. В. Інформаційний пошук в мережі інтернет : thesis. 2021. URL: https://er.nau.edu.ua/handle/NAU/54414 (дата звернення: 29.05.2024).

12) Павлов А. С., Добров Б. В. Метод виявлення масово породжених неприродних текстів на основі аналізу тематичної структури. *Обчислювальні методи і програмування: нові обчислювальні технології*. 2011. Т. 12. С. 58–72.

13) DHQ: Digital Humanities Quarterly: Exploratory Search Through Visual Analysis of Topic Models. Alliance of Digital Humanities Organizations – A Global Coalition of Digital Humanities Organizations.

14) Erosheva E. A., Fienberg S. E., Joutard C., Love T., Shringarpure S. Реконцептуалізація класифікації статей PNAS. *Proceedings of The National Academy of Sciences*. 2010. Vol. 107. Pp. 20899-20904.

15) Andrzejewski D., Buttler D. Latent topic feedback for information retrieval. *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '11. 2011. Pp. 600–608.

16) Andrzejewski D., Zhu X. Latent Dirichlet allocation with topic-in-set knowledge. *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*. SemiSupLearn '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009. Pp. 43–48.

17) Apishev M., Koltcov S., Koltsova O., Nikolenko S., Vorontsov K. Майнінг етнічного контенту в онлайні за допомогою адитивно регуляризованих тематичних моделей. *Computacion in Sistemas*. 2016. Vol. 20, no. 3. P. 387–403.

18) Asuncion A., Welling M., Smyth P., Teh Y. W. Про згладжування та виведення для тематичних моделей. *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*. 2009. P. 27–34.

19) Balikas G., Amini M., Clausel M. On a topic model for sentences. *Proceedings of the 39th International ACM SIGIR Conference on Research and*

Development in Information Retrieval. SIGIR '16. New York, NY, USA: ACM, 2016. P. 921– 924.

20) Bassiou N., Kotropoulos C. Online PLSA: Batch updating techniques including out-of-vocabulary words. *Neural Networks and Learning Systems, IEEE Transactions on*. Nov 2014. Vol. 25, no. 11. P. 1953–1966.

21) Blei D., Lafferty J. Кореляційна тематична модель науки. *Annals of Applied Statistics*. 2007. Vol. 1. P. 17–35.