МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет радіоелектроніки

Факультет Комп`ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів вилучення інформації з неструктурованих текстів на природній мові
(тема)

Виконав:
Студент 2 курсу, групи ІПЗм-19-2
Загоруйко А.В
(прізвище, ініціали)

Спеціальність 121- Інженерія програмного забезпечення
(код і повна назва спеціальності)

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

Керівник доц. Валенда Н.А.
(посада, прізвище)

Допускається до захисту

Зав. кафедри _____ З.В. Дудар
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Комп`ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 121- Інженерія програмного забезпечення
(код і повна назва спеціальності)

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

Освітня програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:
Зав.кафедри _____
(підпис)
« 26 » березня 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента Загоруйка Андрія Володимировича
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів вилучення інформації з неструктурованих текстів на природній мові

затверджена наказом університету від 26.03.2021 № 386 Ст

2. Термін подання роботи до екзаменаційної комісії 11 травня 2021р.

3. Вихідні дані до роботи *природня мова, вилучення інформації, дослідження методів вилучення інформації, пояснювальна записка*

4. Перелік питань, що потрібно опрацювати в роботі *Вступ, аналіз стану проблеми і постановка задачі, огляд методів вилучення інформації, дослідження існуючих алгоритмів, опис проведених теоритичних досліджень, технічна імплементація, висновки, перелік джерел посилання.*

5. Перелік графічного матеріалу із зазначенням креслеників, схем, слайдів, ілюстрацій *Схема узагальненої системи текстомайнінгу, приклади інтерфейсу програм аналогів, діаграма архітектури Watson: DeepQA, діаграма алгоритму виявлення причинно-наслідкових зв'язків, схема процесу побудови текстового класифікатора, приклади стемінга та лематизації, приклади роботи додатку.*

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина	доц. Валенда Н.А.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання індивідуального завдання	25.01.2021	*Виконано*
2	Аналіз предметної області	25.01.2021 – 01.02.2021	*Виконано*
3	Постановка задачі	01.02.2021– 08.02.2021	*Виконано*
4	Аналіз предметної галузі	08.02.2021 – 15.02.2021	*Виконано*
5	Дослідження існуючих методів	15.02.2021 – 22.02.2021	*Виконано*
6	Дослідження існуючих інструментів та додатків	24.02.2021 – 05.03.2021	*Виконано*
7	Написання програмної реалізації	19.03.2021 – 12.04.2021	*Виконано*
8	Підготовка пояснювальної записки	21.04.2021 – 06.06.2021	*Виконано*
9	Підготовка презентації та доповіді	06.06.2021 – 08.05.2021	*Виконано*
10	Нормоконтроль	10.05.2021	*Виконано*
11	Рецензування	11.05.2021	*Виконано*
12	Занесення диплома в електронний архів	12.05.2021	*Виконано*
13	Попередній захист	12.05.2021	*Виконано*
14	Захист звіту	14.05.2021	*Виконано*

Дата видачі завдання 25 січня 2021р.

Студент гр. ІПЗм-19-2 _____ Загоруйко А.В.
(підпис)

Керівник роботи _____ доц. Валенда Н.А.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 89 с., 27 рис., 26 джер.

КОМП'ЮТЕРНА ЛІНГВІСТИКА, ШТУЧНИЙ ІНТЕЛЕКТ, ПРИРОДНЯ МОВА, TEXT MINING, НЕЙРОМЕРЕЖІ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, ВИЛУЧЕННЯ ІНФОРМАЦІЇ, PYTHON, IBM WATSON

Об’єктом дослідження є задачі інтелектуального аналізу текстів на природній мові, для вилучення інформації.

Метою роботи є аналіз та підвищення ефективності існуючих методів вилучення інформації з текстів на природній мові.

Методи розробки базуються на таких технологіях, як Python, Anaconda.

В результаті роботи було досліджено задачі інтелектуального аналізу текстів на природній мові, проведено аналіз та моделювання предметної області, проведено дослідження існуючих засобів та інструментів для інтелектуального аналізу тексту, було обрано інструменти та проведено експеримент.

COMPUTER LINGUISTICS, ARTIFICIAL INTELLIGENCE, NATURAL LANGUAGE, TEXT MINING, NEURAL NETWORKS, INTELLECTUAL DATA ANALYSIS, EXTRACTION OF INFORMATION, PYTHON, IBM WATSON

The object of research is the task of intellectual analysis of texts in natural language, to extract information.

The aim of the work is to analyze and increase the efficiency of existing methods of extracting information from texts in natural language.

Development methods are based on technologies such as Python, Anaconda.

As a result of the work the problems of intellectual analysis of texts in natural language were investigated, the analysis and modeling of the subject area were carried out, the research of existing means and tools for intellectual analysis of the text was carried out, the tools were selected and the experiment was carried out.

Я, Загоруйко Андрій Володимирович, студент гр. ІПЗм-19-2, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів вилучення інформації з неструктурованих текстів на природній мові», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElAr KhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП 7
1 АНАЛІЗ СТАНУ ПРОБЛЕМИ 9
1.1 Загальна характеристика компютерної лінгвістики 9
1.1.1 Додатки комп'ютерної лінгвістики 9
1.1.2 Складності моделювання природної мови 112
1.1.3 Загальні етапи і модулі обробки текстів 13
1.2 Огляд існуючих текстомайнінгових програмних продуктів 18
1.2.1 TextAnalyst 20
1.2.2 Businessobjects Text Analysis 23
1.2.3 AeroText 25
1.2.4 Attensity suite 26
1.2.5 STATISTICA Text Miner 27
1.3 Постановка задачі 28
2 ОПИС ПРОВЕДЕНИХ ТЕОРИТИЧНИХ ДОСЛІДЖЕНЬ 30
2.1 Огляд існуючих сучасних підходів до проектування 30
2.1.1 Механізми визначення причинно-наслідкових зв'язків 31
2.1.2 Методи вилучення інформації 34
2.1.3 Інструментальні засоби підтримки интеллектуального аналізу даних 35
2.1.4 Метод виявлення причинно-наслідкових зв'язків 36
2.2 Опис методів та засобів реалізації тематичної класифікації тексту 40
2.2.1 Алгоритми класифікації з вчителем 41
2.2.2 Представлення даних у задачах класифікації текстів 41
2.2.3 Відбір термінів для класифікації 43
2.2.4 Алгоритм наївної байєсівської класифікації 46
2.2.5 Етапи автоматичної класифікації текстової інформації 49
2.2.6 Опис методів побудови ознак 51
2.3 Огляд використовуваних засобів програмування 55

2.3.1 Огляд мови програмування Python 56
2.3.2 Огляд програмного середовища Anaconda 57
2.3.3 Існуючі бібліотеки для обробки природної мови 56
3. ОПИС ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ 58
ВИСНОВОК 70
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 71
ДОДАТОК А Перелік джерел посилання за науковими напрямами керівника та науковців кафедри програмної інженерії 74
ДОДАТОК Б Сертифікати за участь у науково-технічній конференції 75
ДОДАТОК В Звіт результатів перевірки кваліфікаційної роботи на унікальність тексту 76
ДОДАТОК Г Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 77
ДОДАТОК Ґ Слайди презентації 78

ВСТУП

Комп'ютерна лінгвістика це застосування обчислювальної техніки для аналізу, синтезу і розуміння писемного та усного мовлення. Обчислювальна лінгвістика використовується в режимах миттєвого машинного перекладу, системах розпізнавання мовлення, синтезаторах тексту в мову, системах інтерактивної голосової відповіді, пошукових системах, текстових редакторах та навчальних матеріалах з мови. Міждисциплінарна галузь дослідження вимагає досвіду машинного навчання, глибокого навчання, штучного інтелекту, когнітивних обчислень та нейронауки [1].

У сучасному світі значна частина інформаційних ресурсів представлена у вигляді неструктурованого тексту на природній мові. Це матеріали різних статей, документів, збережені в відкритих джерелах веб-сторінки, текстові файли. Для структурованих даних механізми вибірки досить добре специфіковані. А ось для неструктурованих текстів на природній мові постає завдання інтелектуального аналізу тексту.

Обчислювальне розуміння мови забезпечує людям розуміння мислення та інтелекту. Комп’ютери, які володіють лінгвістичною компетенцією, не тільки сприяють взаємодії людини з машинами та програмним забезпеченням, а й роблять текстові та інші ресурси Інтернету легко доступними різними мовами.

Дещо спрощено завдання комп'ютерної лінгвістики може бути сформульована як розробка методів і засобів побудови лінгвістичних процесорів для різних прикладних задач по автоматичної обробці текстів на ПМ. Розробка лінгвістичного процесору для деякої прикладної задачі передбачає формальне опис лінгвістичних властивостей оброблюваного тексту (хоча б найпростіше), яке може розглядатися як модель тексту (або модель мови) [2].

Метою атестаційної роботи є дослідження існуючих методів засобів та інструментів для інтелектуального аналізу текстів на природній мові, та створення

алгоритму, призначеного для вилучення інформації з текстів, а саме тематичної класифікації текстів, на прикладі сайтів новин.

Об’єктом дослідження є процес обробки неструктурованого тексту на природній мові, вилучення з нього інформації та тематичної класифікації текстових даних в україномовному варіанті.

Предметом дослідження є методи інтелектуального аналізу текстів на природній мові, класифікації текстових даних для тематичної класифікації.

Методами дослідження є аналіз існуючих методів та інструментів, огляд їх переваг та недоліків, а також методи машинного навчання, які базуються на методах text mining. Одним із методів дослідження є розробка додатку для здійснення класифікації текстових даних за допомогою обраних засобів – мови програмування Python, бібліотек Mystem, SnowballStemmer та програмного середовища Anaconda.

У результат проведеного дослідження було отримано програмний застосунок, який може бути використаний як основа для створення інших додатків, це можуть бути різноманітні пошуковими, системи для роботи з текстовими файлами, застосунки для автоматичної перевірки робот на плагіат та інше.

Далі у розробленому програмному застосунку, для вдосконалення, може буду підтримана більша кількість.

За результатами кваліфікаційної роботи магістра було розроблено презентацію (див. додаток В).

Результат теоретичних досліджень опубліковано в роботі «Research of methods for extracting information from unstructured natural language texts» в рамах науково-технічної конференції «Priority directions of science and technology development».

1 АНАЛІЗ СТАНУ ПРОБЛЕМИ

1.1 Загальна характеристика комп’ютерної лінгвістики

Комп'ютерна лінгвістика (КЛ) — міждисциплінарна область, яка виникла на стику таких наук, як інформатика, математика, лінгвістика, штучний інтелект. У своєму розвитку вона продовжує вбирати і застосовувати (при необхідності адаптуючи) розроблені в цих науках методи і інструменти.

В комп'ютерній лінгвістиці об'єктом обробки виступають тексти природньої мови, її розвиток неможливий без базових знань в області загальної лінгвістики (мовознавства) [3]. Лінгвістика вивчає загальні закони природної мови — його структуру і функціонування, і включає такі області: фонологія, морфологія, синтаксис, семантика і прагматика, лексикографія описує лексикон конкретної ПМ.

1.1.1 Додатки комп'ютерної лінгвістики

Найбільш тісно комп'ютерна лінгвістика пов'язана з областю штучного інтелекту (ШІ) [4], в рамках якої розробляються програмні моделі окремих інтелектуальних функцій. Незважаючи на очевидне перетинання досліджень в області комп'ютерної лінгвістики та штучного інтелекту (ШІ) (оскільки володіння мовою відноситься до інтелектуальних функцій), ШІ не поглинає всю КЛ, оскільки вона має свій теоретичний базис і методологію.

Область додатків КЛ постійно розширюється, тому охарактеризуємо тут найбільш відомі прикладні завдання, які вирішуються її інструментами.

В даний час існує цілий спектр комп'ютерних систем машинного перекладу (різної якості), від великих міжнародних дослідницьких проектів до комерційних автоматичних перекладачів.

Ще один досить старий додаток комп’ютерної лінгвістики — це інформаційний пошук (Information Retrieval) [5] і пов'язані з ним завдання індексування, реферування, класифікації та рубрикування документів.

Дуже близька до класифікації завдання рубрикування тексту (Text Classification) — віднесення тексту до однієї з заздалегідь відомих тематичних рубрик (зазвичай рубрики утворюють ієрархічне дерево тематик).

Щодо нове завдання, пов'язана з інформаційним пошуком — формування відповідей на питання (Question Answering) [6]. Завдання вирішується шляхом визначення типу питання, пошуком текстів, потенційно містять відповідь на це питання (при цьому зазвичай застосовуються пошукові машини), і потім витяганням відповіді з виданих текстів.

Актуальна прикладна задача, яка часто відноситься до напрямку Text Mining — це вилучення інформації з текстів (Information Extraction), що потребує і вирішення завдань економічної та виробничої аналітики. При вирішенні цього завдання здійснюється виділення в тесті ПМ певних об'єктів — іменованих сутностей, їх стосунків і пов'язаних з ними подій.

До напрямку Text Mining відносяться і дві інші близькі завдання — виділення думок (Opinion Mining) і аналіз тональності текстів (Sentiment Analysis) [7], що привертають увагу все більшого числа дослідників в силу своєї актуальності.

Ще одна прикладна задача, яка виникла більше 50 років тому і розвиток якої стимулювало появу мережі Інтернет, — це підтримка діалогу на ПМ.

Зовсім інше прикладний напрямок, який розвивається хоча і повільно, але стійко — це автоматизація підготовки та редагування текстів на ПМ. У сучасних дослідженнях КЛ розробляються методи автоматизованого виявлення і виправлення подібних помилок на основі статистики зустрічальності слів і словосполучень [8].

Активно напрямком, що розвивається, є розпізнавання і синтез звуковій мові. Неминуче виникаючі помилки розпізнавання виправляються автоматичними методами на основі словників і морфологічних моделей, також застосовується машинне навчання.

1.1.2 Складності 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 в зв'язку з цією діяльністю.

Текст на ПМ складений з окремих одиниць (знаків), і можливо кілька способів розбиття (членування) тексту на одиниці, що відносяться до різних рівнів.

Загальновизнано існування наступних рівнів [8]:

– рівень пропозицій (висловлювань) — синтаксичний рівень;
– рівень слів (словоформ — слів в певній граматичній формі, наприклад, ручка, дружбою) — морфологічний рівень;
– рівень фону (окремих звуків) — фонологічний рівень.

Фонологічний рівень виділяється 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 підсистеми загальної системи ПМ (взаємопов’язані, але в достатній мірі автономні), і в них самих можуть бути виділені підсистеми. Так, морфологічний рівень включає також підрівень морфем (корінь, префікс, суфікс, закінчення, постфікси).

Питання про кількість рівнів і їх переліку в лінгвістиці досі залишається відкритим. Як окремий може бути виділений лексичний рівень — рівень лексем. Лексема — це слово, як сукупність всіх його конкретних граматичних форм (наприклад, лексему лист утворюють форми лист, листа, листу, листом). Точніше, лексема — семантичний інваріант всіх словоформ. У тексті зустрічаються словоформи (лексеми в певній формі), а в словнику ПМ — лексеми, точніше, в словнику записується канонічна словоформа лексеми, звана також лемою (наприклад, для іменників це форма називного відмінка однини: лист).

В рамках синтаксичного рівня може бути виділений 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 пропозицій, об'єднаних змістом і лексико-граматичними засобами [8].

Ієрархія рівнів проявляється в тому, що одиниці вищого рівня розкладені на одиниці нижчого (наприклад, словоформи на морфи); вищий рівень великою мірою зумовлює організацію нижчого рівня — так, синтаксична структура речення значною мірою визначає, які повинні бути обрані словоформи.

Можна також говорити ще про одне рівні — рівні дискурсу, під яким розуміється зв'язний текст в його комунікативної спрямованості. Під дискурсом розуміється послідовність взаємопов'язаних між собою речень тексту, що володіє певною смисловою цілісністю, за рахунок чого він виконує певну прагматичну задачу.

Особливим є питання про рівень семантики. В принципі, сенс є всюди, де є знакові одиниці мови (морфеми, слова, речення). Підтвердженням самостійності рівня семантики вважається те, що людина зазвичай запам'ятовує сенс висловлювання, а не його конкретну мовну форму. До сих пір не ясна організація цього рівня, пропонується, що існує універсальний набір елементарних семантичних одиниць (званих семами), приблизно 2 тисячі, за допомогою яких можна висловити сенс будь-якого висловлювання.

Крім багаторівневості системи ПМ складність його моделювання пов'язана з постійно відбуваються в ньому змінами (що цілком відчутно після одного-двох десятиліть). Зміни стосуються не тільки словникового запасу мови (нові слова і нові смисли старих), але також синтаксису, морфології і фонетики. Як наслідок, принципово неможливо раз розробити формальну модель конкретного ПМ і побудувати відповідний лінгвістичний процесор. Потрібне постійне поповнення знань про мову на всіх його рівнях і корекція існуючих моделей [8].

Одним із наслідків тривалого історичного розвитку ПМ є нестандартна сполучуваність (синтактика) одиниць на кожному рівні мові. На відміну від штучних формальних мов (мов логіки, мов програмування), в яких сполучуваність знаків диктується їх семантикою і може бути зафіксована синтаксично

(граматично), в природних мовах з'єднання слів у реченнях лише частково може бути описана законами граматики.

Однією з найбільших складнощів при обробці текстів на ПМ є неоднозначність (багатозначність) його одиниць, що виявляється на всіх його рівнях, що виражається в явищах полісемії, омонімії, синонімії.

Полісемія — наявність у однієї одиниці мови кількох пов'язаних між собою значень, зокрема, полісемія слів, наприклад: земля — суша, грунт, конкретна планета. Синонімія — повний або частковий збіг значень різних одиниць, наприклад: синонімія слів: негідник і негідник, синонімія приставок (морфів) пре- і пере- (прекрасний, пересохлий). Омонімія — це збіг за формою двох різних за змістом одиниць (на відміну від полісемії немає смислового зв'язку між збіглися за формою одиницями). Розрізняють такі види омонімії.

- лексична омонімія означає однакові за звучанням та написом слова, що не мають спільних елементів сенсу, наприклад, рожа — обличчя і вид хвороби;
- морфологічна омонімія — збіг форм одного і того ж слова (лексеми), наприклад, словоформа олівець відповідає називному і знахідному відмінках;
- лексико-морфологічна омонімія виникає при збігу словоформ двох різних лексем, наприклад, вірш — два омоніми: дієслово в однині чоловічого роду і іменник в однині, називному відмінку;
- синтаксична омонімія означає неоднозначність синтаксичної структури, що призводить до виникнення кількох інтерпретацій.

1.1.3 Загальні етапи і модулі обробки текстів

Складність формального опису ПМ і його обробки веде до розбиття цього процесу на окремі етапи, відповідні рівням мови. Більшість сучасних лінгвістичних

процесорів відносяться до модульного типу, в якому кожному рівню / етапу аналізу або синтеза тексту відповідає окремий модуль процесора. У разі аналізу тексту окремі модулі ЛП виконують [8]:

- графематичний аналіз (сегментація), тобто виділення в тексті пропозицій і словоформ, точніше токенів (тобто в тексті можуть бути не тільки слова);
- морфологічний аналіз — перехід від словоформ до їх лем (словникових форм лексем) або основ (ядерних частин слова, за вирахуванням словозмінних морфем);
- синтаксичний аналіз — виявлення синтаксичних зв'язків слів та граматичної структури речень;
- семантичний і прагматичний аналіз, при якому визначається зміст фраз і відповідна реакція системи, в рамках якої працює ЛП.

Таким чином, лінгвістичний процесор можна розглядати як багатоетапний перетворювач, що перекладає в разі аналізу тексту кожну його пропозицію у внутрішнє представлення його сенсу і навпаки в разі синтезу.

Можливі різні схеми об'єднання і взаємодії модулів розглянутих етапів, проте окремі рівні — морфологія, синтаксис і семантика зазвичай обробляються різними механізмами. Під час вирішення деяких прикладних задач можна обійтися без подання в процесорі всіх етапів/рівнів (наприклад, в ранніх експериментальних програмах КЛ оброблювані тексти ставилися до дуже вузьких проблемних областей з обмеженим набором слів, так що не був потрібен морфологічний і синтаксичний аналіз) .

Модулі морфологічного аналізу словоформ розрізняються переважно за такими параметрами:

- результату роботи — лема або основа з набором характеристик (рід, число, відмінок, вид, особа і т.п.) заданої словоформи;
- методу аналізу — з опорою на словник словоформ мови або на словник основ, або ж безсловниковий метод;
- можливості обробки словоформи лексеми, не включеної до словнику.

Для реалізації синтаксичного етапу в рамках КЛ запропоновано велике число різних методів, що відрізняються описом синтаксису мови, використання цієї інформації при аналізі, а також способом подання синтаксичної структури пропозиції [8]. Виділяються три основні підходи: генеративний підхід, висхідний до ідей граматик, що породжують Н. Хомського [9]; підхід, висхідний до ідей І. Мельчука і представлений в лінгвістичної моделі «Сенс-Текст», а також підхід, в рамках якого робляться спроби подолати недоліки перших двох підходів, наприклад, теорія синтаксичних груп.

В рамках генеративного підходу синтаксичний аналіз здійснюється, як правило, на основі формальної контекстно-вільної граматиці, яка описує фразову структуру пропозиції, або ж на основі деякого розширення контекстно-вільної граматики. Ці граматиці виходять з послідовного лінійного членування пропозиції на фрази (різні словосполучення) і відображають тому одночасно як його синтаксичну, так і лінійну структури. Отримана внаслідок ієрархічна синтаксична структура пропозиції ПМ описується деревом складових, в листі якого знаходяться слова речення, піддерева відповідають входять в пропозицію синтаксичним конструкціям (фразам), а дуги висловлюють відносини вкладення конструкцій. Даний підхід був значно розвинений в ряді робіт, зокрема, в [8].

В рамках другого підходу для подання синтаксичної структури пропозиції використовується більш наочний спосіб — дерева залежностей [9].

Дерева складових більше підходять для опису мов з жорстким порядком слів, подання з їх допомогою розірваних і непроективних конструкцій вимагає розширення використовуваного граматичного формалізму. Зате в рамках цього підходу більш природно описуються конструкції з непідрядними відносинами. У той же час загальні труднощі для обох підходів — уявлення однорідних членів речення.

В рамках генеративного підходу валентності слів (перш всього, дієслів) описуються переважно у вигляді спеціальних фреймів (subcategorization frames) [8], а в рамках підходу, заснованого на деревах залежностей — як моделі управління.

Модулі синтаксичного аналізу в обох розглянутих підходах спираються на граматики ПМ. Загальна кількість правил граматики може бути від декількох десятків до декількох сотень, в залежності від використовуваного словника: чим більше інформації представлено в словнику, тим коротше може бути граматика і навпаки. Так, в моделі «Сенс-Текст» [5] акцент робиться на словник, а не на граматику; в використовуваному словнику зберігається інформація, що відноситься до різних рівнів мови, зокрема, про моделі управління слів і нестандартної сполучуваності слів.

Етап семантичного аналізу тексту найменш опрацьований в рамках КЛ. Для локального семантичного аналізу, тобто аналізу пропозицій були запропоновані так звані відмінкові граматики і семантичні відмінки (валентності) [8], на базі яких семантика пропозиції описується через зв'язку головного слова (зазвичай дієслова) з його семантичними актантами, тобто через семантичні відмінки.

Для подання семантики всього тексту зазвичай використовуються два формалізму (обидва вони детально описані в рамках штучного інтелекту):

– формули обчислення предикатів, що виражають властивості, процеси, стани, дії і відносини;
– семантичні мережі —графи, де вершини відповідають поняттям, а дуги — відносинам.

Мало досліджений у КЛ рівень прагматики і дискурсу, до якого аналіз текст в цілому. Переважно розроблені методи аналізу локальної зв'язності тексту, в першу чергу, дозвіл анафоричних посилань. Серед робіт, ідеї яких все частіше застосовуються, слід указати теорію риторичних структур; в роботі [19] запропонована модель синтезу дискурсивної структури описових текстів.

Використовувані в комп'ютерній лінгвістиці моделі ПМ зазвичай будуються з урахуванням лінгвістичних теорій і моделей; виділимо особливості саме моделей КЛ [8]:

– формальність і, в кінцевому рахунку, алгоритмизованість;
– функціональність (відтворення функцій мови як «чорного ящику», без побудови точної моделі синтезу та аналізу мови людиною);

– опора на лінгвістичні ресурси;
– експериментальна обгрунтованість, яка передбачає тестування моделі на різних текстах.

Розробка і застосування лінгвістичних процесорів спирається на використання тих чи інших лінгвістичних ресурсів: лексичних (словарних) і текстових. До лексичних ресурсів належать словники, тезауруси, онтології.

Словники є найбільш традиційною формою представлення лексичної інформації; вони розрізняються своїми одиницями (зазвичай слова або словосполучення), структурою, охопленням лексики (словники термінів конкретної проблемної області, словники загальної лексики, словники синонімів або паронімів і т.п.). Одиниця словника називається словниковою статтею, в ній є інформація про лексему. Лексичні омоніми зазвичай представляються в різних словникових статтях.

До лексичних ресурсів належать бази словосполучень, до яких відбираються найбільш типові словосполучення конкретного мови.

Більш складними видами лексичних ресурсів є тезауруси і онтології. Тезаурус — це семантичний словник, тобто ловник, в якому представлені смислові зв'язки слів — синонімічні, відношення Рід-Вид (іноді звані ставленням Вище-Нижче), Частина-Ціле, асоціації.

З поняттям тезауруса тісно пов'язане поняття онтології [11]. Онтологія — набір понять, сутностей певній галузі знань, орієнтований на багаторазове використання для різних завдань. Онтології можуть створюватися на базі існуючої в мові лексики — в цьому випадку вони називаються лінгвістичними.

Подібної лінгвістичної онтологією вважається система WordNet [8] — великий лексичний ресурс, в якому зібрані слова англійської мови: іменники, прикметники, дієслова і прислівники і представлені їх смислові зв'язки декількох типів. Для кожної із зазначених частин мови слова згруповані в групи синонімів (сінсети), між якими встановлені відносини антонімії, гіпонімії (відношення рід-вид), меронімії (відношення частина-ціле), тропонімії.

Корпус текстів — це представницький масив текстів, зібраний за певним принципом (по жанру, авторської приналежності і т.п.) і володіє лінгвістичної розміткою — морфологічної, акцентної, синтаксичної, дискурсивної або ін. [12]. В даний час відомо кілька сотень різних корпусів (для різних ПМ і з різною разміткою).

Розмічені корпуси створюються зазвичай експертами-лінгвістами та використовуються як для лінгвістичних досліджень, так і для налаштування (навчання) лінгвістичних процесорів на основі методів машинного навчання.

Зауважимо, що оскільки корпусу і колекції текстів завжди обмежені за представленими в них мовних явищ, як більш повного джерела зразків сучасної мови можуть розглядатися тексти мережі Інтернет.

1.2 Огляд існуючих текстомайнінгових програмних продуктів

Текстомайнінг (text mining) часто називають також текстовим дейтамайнінгом (text data mining), що частково розкриває взаємозв'язок двох цих технологій. Якщо дейтамайнінг дозволяє витягувати нові знання (факти, приховані закономірності, невідомі взаємозв'язку і т.п.) з великих обсягів структурованої інформації, то текстомайнінг — знаходити знання в неструктурованих текстових масивах.

Текстомайнінг додає до дейтамайнінга додатковий етап — переведення неструктурованих текстових масивів до структурованих. Після чого дані можуть оброблятися за допомогою стандартних методів дейтамайнінгу [13].

Обсяг інформації, що накопичується в численних текстових базах, що зберігаються в особистих ПК, локальних і глобальних мережах стрімко збільшується, тому все більш актуальним стає рішення текстомайнінга.

Актуальність текстомайнінга зростає в міру того, як людям різних професій доводиться приймати рішення на основі аналізу великого обсягу слабоструктурованих або зовсім неструктурованих текстів (рис. 1.1).

Рисунок 1.1 — Очікуване зниження / зростання даних різного ступеня структурованості в найближчі три роки [13]

Одним з нових напрямків текстомайнінгу є Opinion Mining (OM) — технологія, яка концентрується не стільки на змісті документа, скільки на думці, що він висловлює.

Opinion Mining Systems широко використовуються для автоматичної оцінки (позитивної, негативної, нейтральної) новинних подій, продуктів, персоналій, організацій, країн світу і т.д., що надходять в режимі реального часу з повідомлень е-ЗМІ (електронних засобів масової інформації), повідомлень блогерів, форумів тощо в Інтернеті, тобто всього того, що позначається загальним терміном Social Media і, зокрема, Social Media Monitoring. Умовно систему текстомайнінгу можна поділити на чотири блоки (рис. 1.2).

У наступному блоці перерахований набір необхідних користувачеві завдань, кожна з яких вимагає свого технологічного рішення. До набору завдань слід віднести: кластеризацію, класифікацію, побудова семантичних мереж, витяг фактів, понять, витяг думок, анотування, сумарізація, тематичне індексування,

пошук за ключовими словами, відповідь на запит, створення таксономій і тезаурусів.

Рисунок 1.2 — Структура узагальненої системи текстомайнінгу [15]

На даний час пропонується досить багато інструментів текстомайнінгу — від простих програм, що спираються на аналіз термінів у текстах, наприклад WordStat, до складних застосувань типу Aerotext і Businessobjects Text Analysis.

1.2.1 TextAnalyst

TextAnalyst (рис. 1.3) — це засіб семантичного аналізу, навігації та пошуку в неструктурованих текстах. У продукті реалізована синергія від використання технологій лінгвістичного аналізу і нейромереж.

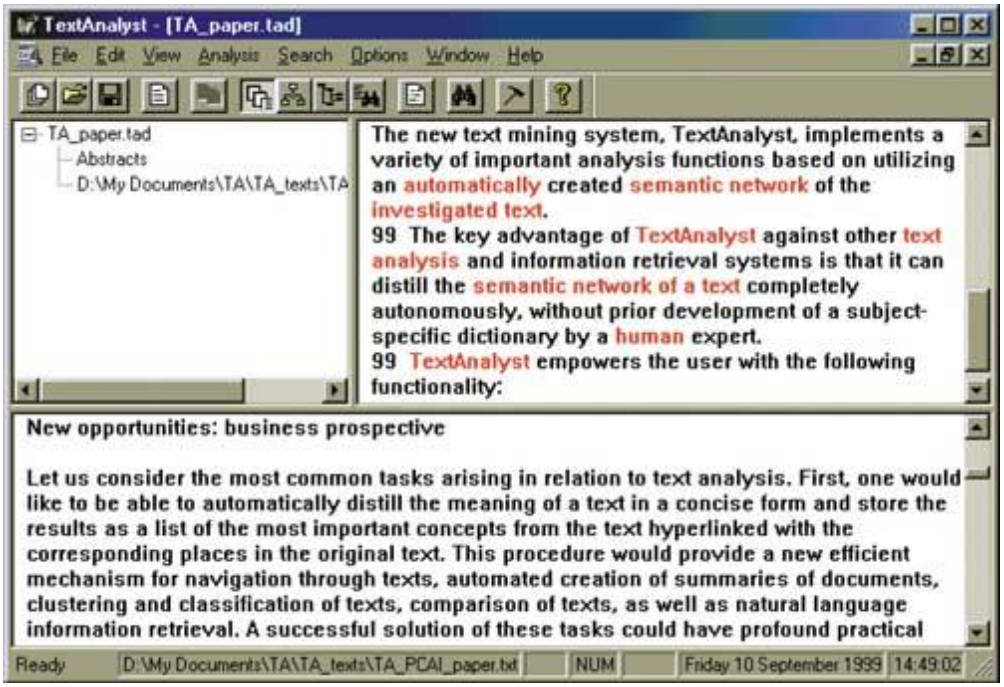


Рисунок 1.3 — Інтерфейс програми TextAnalyst [14]

Система TextAnalyst допоможе швидко резюмувати, ефективно управляти і об'єднувати в групи документи в текстовій базі. Вона полегшує пошук семантичної інформації або може сфокусувати вивчення тексту на якомусь певному предметі.

Продукт забезпечує вирішення таких завдань, як складання резюме об'ємного тексту, дає уявлення про що текст, дозволяє ефективно здійснювати навігацію по великим текстових документів і пошук інформації за допомогою запитів на природній мові.

Основні можливості програми відображені на рис. 1.4.

Рисунок 1.4— Основні можливості програми TextAnalyst

Продукт існує як автономне і як вбудовуване рішення. Одна з можливостей системи — це побудова мережі семантичних зв'язків тексту (Semantic Network). Отримана семантична мережа є основою для подальшого смислового аналізу тексту. Семантична мережа — це набір найбільш важливих понять, витягнутих з тексту, і взаємозв'язків між ними, оцінених на основі їх відносної важливості (рис. 1.4).

Ефективна навігація по текстовим масивів здійснюється на основі гіперпосилань за ключовими словами (поняттями) в семантичній мережі на ті пропозиції в документі, які містять необхідні комбінації слів. Окремі пропозиції можуть мати, в свою чергу, гіперпосилання на ті місця в початковому тексті, де вони були виявлені.

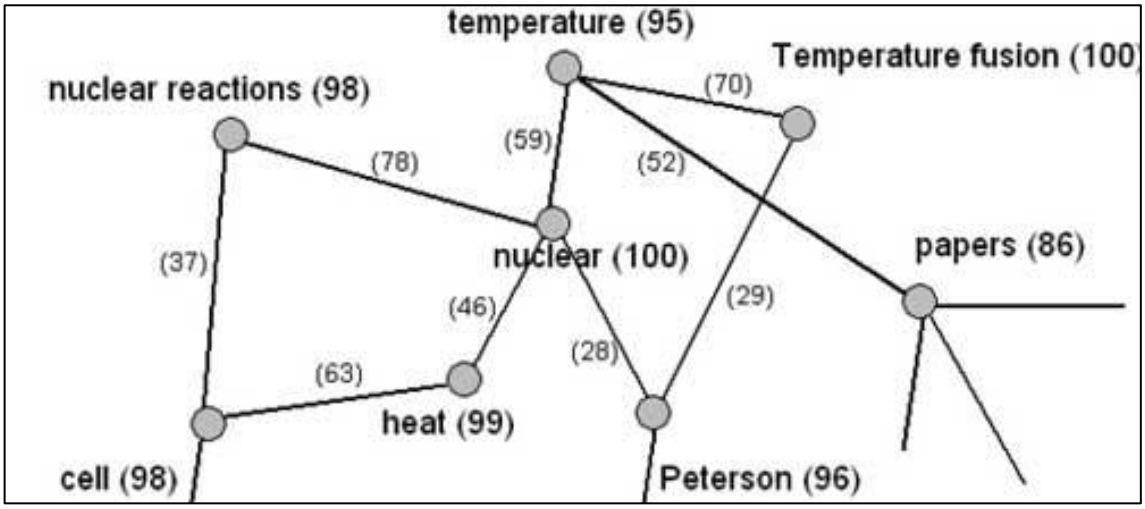


Рисунок 1.4 — Приклад фрагмента семантичної мережі

Продукт забезпечує можливість виявлення тематичної структури тексту — програма дозволяє автоматично генерувати древообразную тематичну структуру досліджуваного тексту. Чим більш суттєвими є теми в тексті, тим ближче вони розташовуються до кореню деревовидної структури.

За допомогою підключення словників (слів, що включаються і виключаються) програма дозволяє досліднику сконцентруватися на досліджуваному предметі.

Кластеризація текстів базується на видаленні слабких посилань в семантичної мережі, що призводить до розбиття тексту на семантично однорідні кластери.

Процедура розгляду заяв про природною мовою здійснюється на основі аналізу на наявність семантично значущих слів в досліджуваній базі і повернення релевантних пропозицій з вихідної текстової бази. Додатково формується так зване піддерево понять, що відносяться до запиту, що також допомагає вдосконалити пошук.

1.2.2 Businessobjects Text Analysis

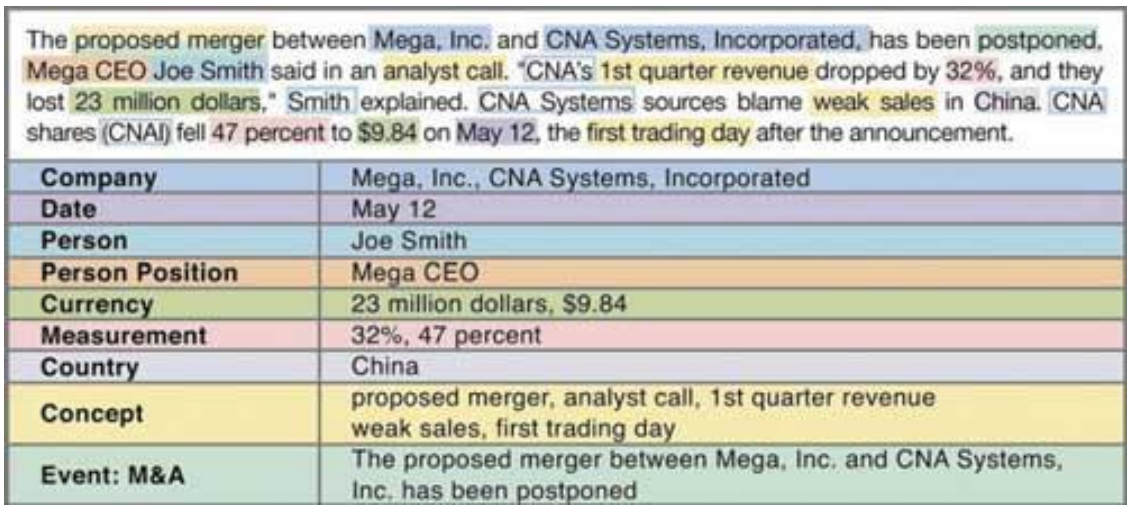
BusinessObjects Text Analysis володіє потужними лінгвістичними можливостями з читання і розуміння документів на 30 мовах, базуючись на розвиненому NLP-апараті, і дозволяє обробляти дані на базі 220 файлових форматів. Аналіз тексту виконується не на рівні слів і частоти їх появи в тексті — програма йде від розуміння побудови речень в природних мовах.

Дані можливості доповнюються категоризацією, що дозволяє застосовувати для користувача таксономії при аналізі тексту для наступної класифікації, реферування і побудови пов'язаних вичавок тексту.

Програма дозволяє витягувати інформацію по 35 типам об'єктів і подій, включаючи людей, географічні місця, компанії, дати, грошові суми, email-адреси, і виявляти взаємозв'язки між ними.

Потужний інструмент дозволяє обробляти величезні масиви інформації, визначаючи шукані об'єкти (рис. 1.5).

На основі структури природних мов програма може розпізнавати інформацію, пов'язану з заданими користувачем об'єктами, такими як назви проектів, аналізувати взаємозв'язки між подіями і конкретні фрази на предмет сентимент-аналізу.



The proposed merger between Mega, Inc. and CNA Systems, Incorporated, has been postponed. Mega CEO Joe Smith said in an analyst call. "CNA's 1st quarter revenue dropped by 32%, and they lost 23 million dollars," Smith explained. CNA Systems sources blame weak sales in China. CNA shares (CNAI) fell 47 percent to $9.84 on May 12, the first trading day after the announcement.

Company	Mega, Inc., CNA Systems, Incorporated
Date	May 12
Person	Joe Smith
Person Position	Mega CEO
Currency	23 million dollars, $9.84
Measurement	32%, 47 percent
Country	China
Concept	proposed merger, analyst call, 1st quarter revenue weak sales, first trading day
Event: M&A	The proposed merger between Mega, Inc. and CNA Systems, Inc. has been postponed

Рисунок 1.5 — Приклад роботи програми [15]

BusinessObjects Text Analysis надає можливість класифікації документів за представленими категоріям, які можуть явно і не бути присутнім у вихідному документі. Наприклад, ваш документ може бути віднесений до категорії «скарга користувача» навіть в тому випадку, якщо слово «скарга» ніде в ньому не зустрічається. Програма сама виявить незадоволеність клієнта і віднесе документ до цієї категорії автоматично.

Реферування здійснюється на базі вилучення найбільш релевантних пропозицій, що характеризують смисловий зміст документа.

1.2.3 AeroText

AeroText — це текстомайнінговий додаток, що використовується для контент-аналізу, який може застосовуватися на різних мовах. Воно розроблялося в підрозділі Integrated Systems and Solutions корпорації Lockheed Martin Corporation для потреб оборонного відомства США (U.S. Intelligence Community (Department of Defense)). Згодом це рішення стало одним з провідних в області текстомайнінга, інтелектуальний модуль AeroText інтегрований і в інші продукти. AeroText

забезпечує вилучення інформації та аналіз взаємозв'язків між витягнутими одиницями інформації (рис. 1.6).

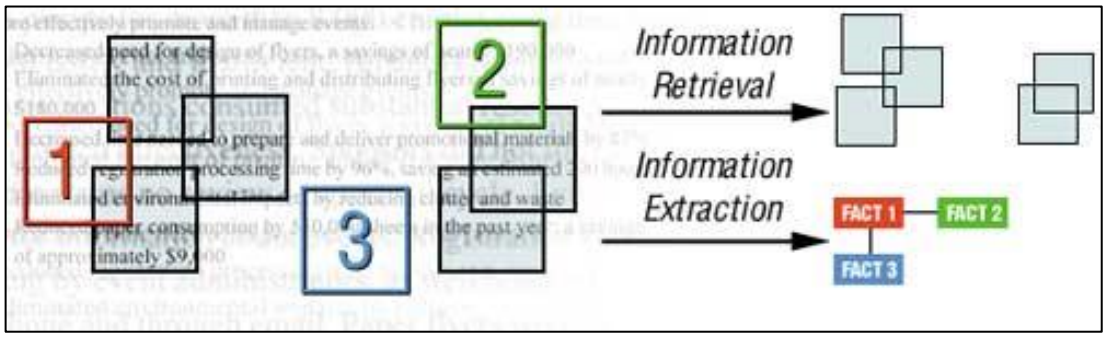


Рисунок 1.6 — Схема роботи програми AeroText [16]

AeroText — це ПЗ, яке дозволяє вирішувати проблему інформаційного перевантаження на базі вилучення елементів аналізу інформації, таких як сутності (entities), взаємини (relationships) і події (events), в неструктурованих текстах. Програма також дозволяє виявляти приховані взаємозв'язки та події в текстах. Додаток може бути інтегровано з іншими інструментами управління знаннями (knowledge management tools), володіє засобами індивідуальної настройки під досліджувану середу і підтримує вилучення даних на різних мовах [19].

AeroText — це рішення data-independent, тобто рішення, яке не залежить від типу документа, тематики і типу мови. За допомогою цієї технології можуть вирішуватися такі завдання, як побудова бази даних, маршрутизація документів, броузінг, підготовка реферату (вичавки тексту), побудова повнотекстових пошукових індексів і т.п. Версія AeroText 5.x існує у вигляді набору компонентів. Програма дозволяє здійснювати вилучення інформації, пов'язаної з конкретними об'єктами (персони, організації, географічні об'єкти і т.п.), ключові фрази (вказівка на конкретний час, обсяги грошей) і т.п. Рішення також аналізує взаємозв'язки між предметами, дозволяючи вирішити проблему множинних значень одного і того ж предмета, здійснює ідентифікацію взаємин між предметами, витяг подій (хто, де, коли), категоризацію тем (предмет, його визначення), визначення часового проміжку, коли мало місце подія, визначення місця, яке може бути прив'язане до карти.

1.2.4 Attensity suite

Attensity — це набір текстомайнінгових рішень, що базуються на статистичних і NLP-технологіях.

Технології Attensity — це результат десятирічних досліджень в області комп'ютерної лінгвістики, які привели до створення ПО, що дозволяє витягати знання з неструктурованих текстів. Програму відрізняють широкий набір технологій вилучення — від ключових слів до подій, відкрита архітектура і зручний інтерфейс (рис. 1.7). Програма Attensity пропонує багатий набір інструментів для аналізу текстів, який включає засоби інтеграції, інтелектуальний модуль, масштабовану серверну платформу, використовує запатентовані засоби добування інформації і дозволяє створювати бізнес-рішення «під ключ». Технологія дає користувачам можливість отримувати і аналізувати наступні факти: хто, що, де, коли і чому робив, — і згодом уточнювати, хто, в яких місцях і в яких подіях брав участь і як вони між собою пов'язані.

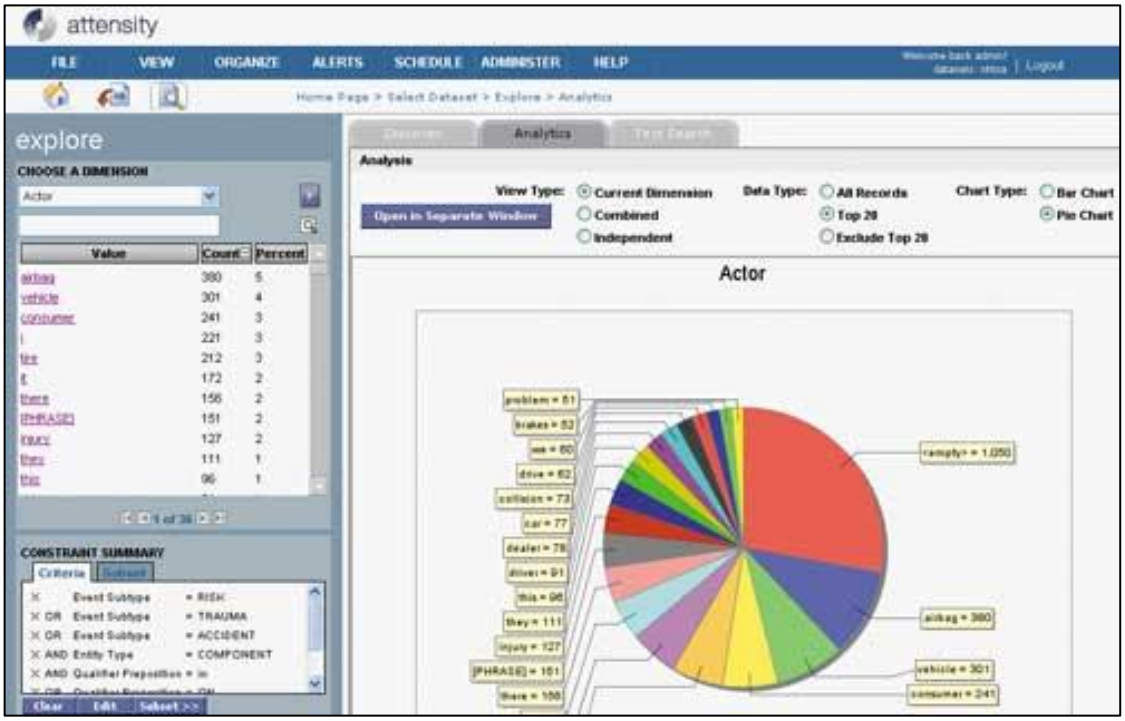


Рисунок 1.7 — Інтерфейс Attensity [17]

В основі Attensity Text Analytics suite лежить технологія добування інформації з неструктурованих текстів. Вона дозволяє витягувати інформацію, захована в неструктурованому тексті, і переводити її в структуровані дані, які мають зв'язки, які можуть бути проаналізовані тими ж методами, що і інші види структурованих даних. Витяг інформації як з неструктурованих, так і з структурованих джерел дає додаткові можливості.

Програма може працювати навіть з текстами, що містять граматичні помилки, що важливо в тому випадку, коли доводиться обробляти повідомлення електронної пошти, особисті записи, скарги клієнтів і т.п.

1.2.5 STATISTICA Text Miner

STATISTICA Text Miner — це додаткове розширення програми STATISTICA Data Miner, призначене для перекладу неструктурованих текстових даних в інформацію, придатну для прийняття рішень. STATISTICA Text Miner дозволяє витягувати з тексту необхідні дані, структурувати їх і представляти інформацію в графічному вигляді (рис. 1.8).

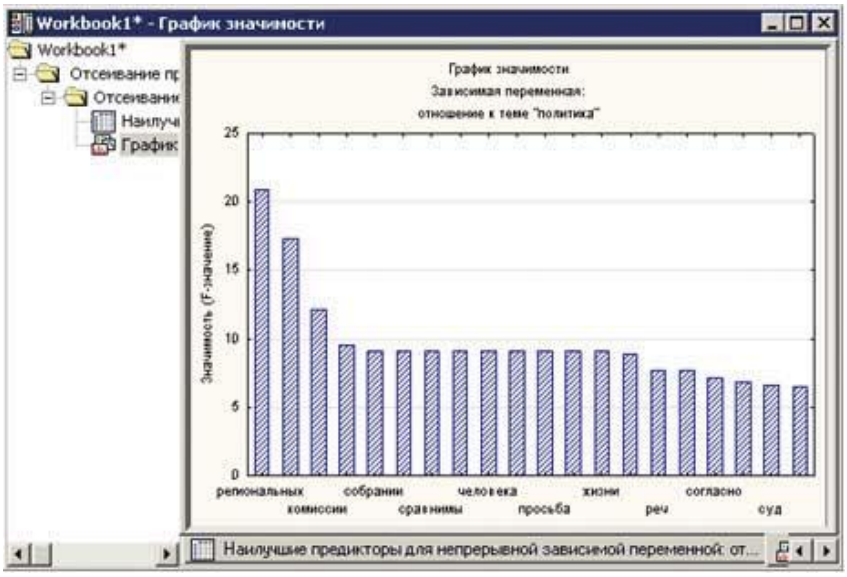


Рисунок 1.8 — Інтерфейс STATISTICA Text Miner

В якості вхідних даних можна використовувати не тільки текстові документи або веб-сторінки, але і файли інших типів. Програма забезпечує доступ до текстових документів в різних форматах, включаючи TXT, PDF, PS, HTML, XML, RTF і ін.

Документи можуть бути оброблені, перш ніж вони будуть проіндексовані (фактично ці процеси відбуваються одночасно). Програма написана таким чином, що підтримка додаткових мов здійснюється з мінімумом витрат. Засоби аналізу дозволяють отримати кількісний звіт по досліджуваного тексту. Шляхом статистичного аналізу можна оцінювати ступінь схожості документів. На основі зіставлення документів по частоті появи в них різних слів можна встановити приналежність документа до тієї чи іншої смислової категорії. Кластерний аналіз дозволяє ідентифікувати групи схожих за змістом документів [18]. Прогнозны методи добування даних дозволяють встановлювати зв'язки між отриманими чисельними характеристиками документів з іншими індикаторами (наприклад, оцінити намір ввести в оману, медичний діагноз і т.д.).

STATISTICA Text Miner має відкриту архітектуру. Програмне забезпечення для текстомайнінга може бути інтегровано з будь-яким ПЗ з лінійки продуктів STATISTICA: STATISTICA Data Miner workspace, WebSTATISTICA або зі звичайними додатками STATISTICA.

1.3 Постановка задачі

В даному розділі розглянута загальна характеристика комп’ютерної лінгвістики, описано задачі, які вирішуються у цій галузі, складності їх вирішення, загальні етапи та модулі обробки природної мови. А також зроблено огляд існуючих додатків, та конкретно текстомайнінгових програмних продуктів.

Метою дослідження є створення алгоритму, призначеного для тематичної класифікації текстів, пов’язаних з новинами.

Для досягнення поставленої мети необхідно виконати наступні завдання:

– здійснити аналіз методів та алгоритмів машинного навчання для вирішення задачі автоматичної класифікації неструктурованих текстів на природній мові;
– вибрати модель для представлення текстової інформації в класифікаторі;
– відповідно до обраної моделі представлення текстової інформації, розробити алгоритм попередньої обробки текстів;
– розробити модифікований метод класифікації текстової інформації;
– зробити програмну реалізацію розробленого алгоритму автоматичної класифікації текстової інформації;
– розробити графічний інтерфейс керування програмним продуктом;
– підготувати дані для оцінки якості класифікації;
– провести дослідження ефективності розробленої інформаційної технології.

2 ОПИС ПРОВЕДЕНИХ ТЕОРИТИЧНИХ ДОСЛІДЖЕНЬ

2.1 Огляд існуючих сучасних підходів до проектування

Інтелектуальний аналіз даних (Text Mining) являє собою набір технологій і методів, мета яких полягає в отриманні ключовий і найбільш значимої інформації з неструктурованого тексту. До області знання Text Mining входять такі напрямки комп'ютерної лінгвістики, як:

– інтелектуальний аналіз даних — Data mining;
– аналіз даних у мережі Інтернет — Web mining;
– пошук інформації — Information Retrieval;
– витяг інформації — Information Extraction;
– обробка тексту на природній мові — Natural Language Processing.

Технологія інтелектуального аналізу даних (ІАД) досить точно визначається в [19]: «Data Mining — це процес виявлення в великих сирих даних раніше невідомих, об'єктивних, корисних на практиці знань, необхідних для прийняття будь-яких рішень».

Основу математичних методів Data Mining слід поділити на дві групи: статистичні методи (аналіз часових рядів, кореляційний і регресійний аналіз, дискримінантний аналіз і ін.) та методи комп'ютерної математики (дерева рішень, нечітка логіка, системи обробки експертних знань, штучні нейронні мережі та ін.).

Починаючи з початку 2000-х років парадигма вирішення проблеми автоматичного вилучення інформацій з неструктурованих даних стала тяжіти до статистики і машинного навчання (ML) [19].

Технології ML в завданні вилучення інформації з текстів формується як рішення задачі класифікації з використанням статистичних моделей. Методи машинного навчання, які використовуються для вирішення завдання, поділяються на кілька етапів.

– навчання «з вчителем»: навчання на основі навчальної колекції, що включає явно специфіковані (вручну) іменовані суті. Методи цієї групи

оцінюють параметри для позитивно певних прикладів корпусу і при роботі з новим корпусом використовують значення цих параметрів. Сюди відноситься Байєсівський класифікатор, приховані марковські моделі, принцип максимуму ентропії, дерева прийняття рішень, метод опорних векторів, умовні випадкові поля та інше;

– часткове навчання «з вчителем»: від попереднього підходу відрізняється тим, що вихідна навчальна колекція містить дуже маленький набір початкових даних. За допомогою реалізації методу бутстреппінга здійснюється ітеративне навчання класифікатора;
– навчання «без вчителя»: для вирішення завдання не вимагають попереднього створення корпусу прикладів. Такі методи здатні зробити висновок по сирому текстового матеріалу.

Завдання Text Mining можна розуміти в когнітивному контексті. Це обумовлено тим, що критерій якості виділення інформації з інформації визначається людиною. В цілому когнітивні методи і процеси допомагають перетворити неявне знання до явного.

На етапі когнітивного дослідження виконується формування зв'язків в інформаційній структурі та визначення напрямку зв'язків. В цьому відношенні основною моделлю представлення знань можуть бути причинно-наслідкові зв'язки.

Сучасні підходи вилучення причинно-наслідкових зв'язків припускають наявність засобів автоматичного вилучення даних зв'язків в процесі пошуку інформації [19], підтримки прийняття рішень, або прогнозування майбутніх подій.

2.1.1 Механізми визначення причинно-наслідкових зв'язків

Під причинно-наслідковим зв'язком розуміють зв'язок між явищами, при якій одне явище тягне за собою інше явище. Перше явище називається причиною, за наявності певних умов породжує інше явище, що має назву слідство. Одним з

найпростіших способів вираження причинно-наслідкових зв'язків між двома подіями є пропозиції типу «подія А викликано подією Б» або «з події А слід подія Б». Причинність може бути виражена з використанням множини різних типів речень і мати різноманітні синтаксичні уявлення. Існують такі способи вираження явних причинно-наслідкових зв'язків англійською мовою:

- за допомогою причинних спілок для з'єднання двох фраз або пропозицій;
- за допомогою причинних дієслів;
- за допомогою результуючих конструкцій, в яких після дієслова слід фраза, що описує стан об'єкта в результаті дії;
- за допомогою умовних виразів «якщо-то»;
- з допомогою причинних прикметників або прислівників.

Широко відомим проектом є проект IBM Watson [19], що використовує когнітивний підхід для виконання аналізу великої кількості різних зовнішніх джерел інформації, виявлення неочевидних залежностей між різними видами даних, що зберігаються. Цей проект дозволяє, враховуючи специфіку даних, досить оперативно давати релевантну відповідь на отриманий запит. При цьому Watson справляється із завданням, у багатьох випадках, навіть краще людини і, більш того, обробка даних йде набагато швидше, робота ведеться з набагато більшими обсягами.

Для того, щоб навчити систему аналізувати складні смислові конструкції, з урахуванням емоцій і інших чинників, фахівці використовували глибоку обробку природної мови. А саме — питально-відповідну систему тематичної аналітики DeepQA (Deep Question-Answering) [19]. На рисунку 2.1 приведено архітектуру DeepQA. Спрощений алгоритм роботи Watson при відповіді на питання, поставлене на природній мові, виглядає наступним чином:

- у кожному питанні проводиться синтаксичний аналіз для виділення основних особливостей питання;
- система генерує ряд гіпотез (варіанти відповіді), аналізуючи базу даних з фразами, в яких з певною часткою ймовірності можуть містяться правильну відповідь;

– система виконує глибоке порівняння мови, на якому було поставлено питання, з мовами, на яких міститься кожен один з можливих варіантів відповіді, застосовуючи різні алгоритми визначення логічних зв'язків;
– далі, кожен логічний алгоритм виставляє одну або кілька оцінок, що показують ступінь відповідності знайденого відповіді заданому питанню;
– кожній оцінці присвоюється певний ваговий коефіцієнт. При цьому використовується статистична модель, яка фіксує успішність роботи кожного алгоритму при виявленні логічних зв'язків. Згодом дана модель використовується для визначення загального рівня впевненості Watson в тому, що знайдена відповідь дійсно є вірним;
– пункти 3-5 повторюються до тих пір, поки Watson не знайде відповіді, які будуть мати найбільші шанси опинитися правильними.

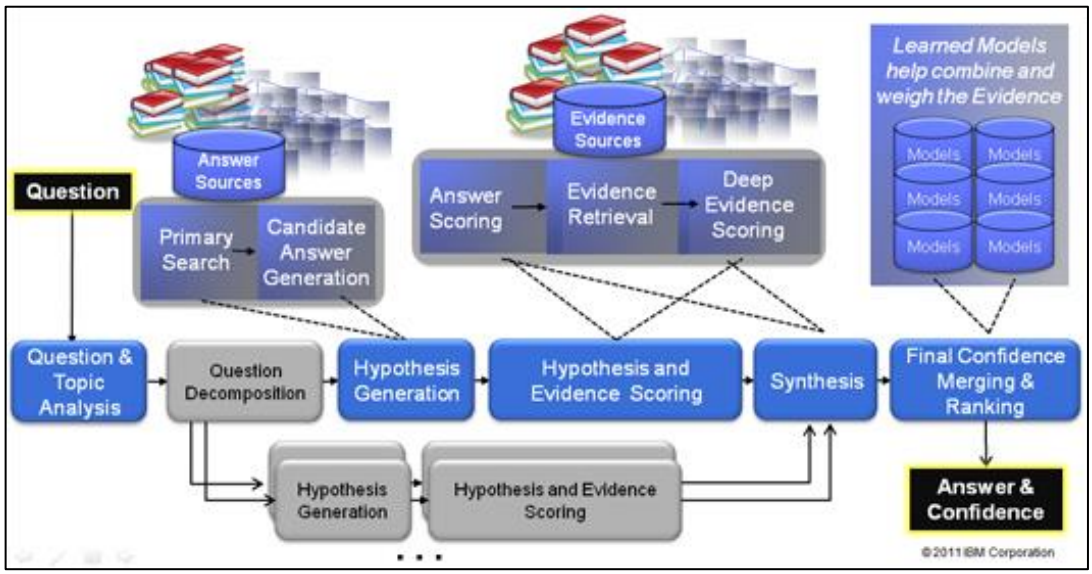


Рисунок 2.1 — Архітектура Watson: DeepQA

Створення системи, здатної здійснити глибоку обробку природної мови, дозволило вирішити й іншу проблему — аналіз величезної кількості інформації, яка генерується щодня. Це неструктурована інформація, начебто твітів, повідомлень соціальних мереж, звітів, статей і т.п. IBM Watson навчився розуміти неструктуровану інформацію, знаходити зв'язки між нею і використовувати дані знання в різних цілях.

2.1.2 Методи вилучення інформації

Методи добування інформації на основі правил. Системи вилучення інформації за допомогою правил засновані на застосуванні заданого набору правил. Такі системи використовують експертні знання для вирішення різних завдань, які зазвичай вимагають людського інтелекту. Експертне знання часто представляється у вигляді правил або як дані в комп'ютері, які можуть повторно застосовуватися для вирішення завдання. Переважно правила задаються у вигляді ЯКЩО X => ТО Y, де в якості паттерна X можуть виступати регулярні вирази, словники, частини мови або інші правила. Будемо говорити, що фрагмент тексту анотується відповідною функцією Y, якщо він задовольняє одному з правил X.

Існує два головних принципи алгоритму вивчення правил: висхідний метод (bottom-up) і спадний метод (top-down). У разі висхідного методу правила поширюються від винятків до загальних випадків, а при низхідному методі навпаки, від загальних випадків до винятків. Даний підхід використовується в алгоритмах Whisk, LP2 і Rapier. Важливою перевагою даних алгоритмів є те, що ступінь достовірності витягнутої інформації завжди є дуже високим. Однак безліч правил визначаються для конкретної предметної області, що є істотним недоліком.

Методи вилучення на основі класифікації. Витяг інформації з текстових документів з використанням статистичних моделей засновано на поділі вихідного тексту на вектор слів (токенів) і анотування кожного з цих слів класом із заданого вектору класів. При вирішенні задачі класифікації слова (об'єкту) в корпусі текстів визначається набір ознак, на підставі яких об'єкти будуть зіставлятися. Ознака може приймати як булеві значення, так і числові значення.

Таким чином, завдання вилучення інформації з тексту зводиться до класифікації об'єктів, для вирішення якого існують такі підходи:

– раціоналістичний підхід. Ідентифікація об'єктів відбувається на основі продукційних правил, які задаються вручну;

– машинне навчання. Завдання пошуку правил формулюється як рішення задачі класифікації з використанням статистичних моделей;
– гібридний підхід. Є об'єднанням двох попередніх підходів.

У разі машинного навчання модель класифікації поділяє на два етапи: навчання і прогнозування. Під час процесу навчання знаходиться модель з анотованих даних, яка може поділити навчальну вибірку, а при процесі прогнозування модель, яку знайшли під час навчання, використовується для визначення того, чи повинен непомічений екземпляр бути класифікований.

Найбільш відомими методами моделі класифікації є наступні: метод опорних векторів (Support Vector Machines, SVM), метод k-найближчих сусідів (k-Nearest Neighbors, kNN) і метод Naive Bayes.

2.1.3 Інструментальні засоби підтримки інтелектуального аналізу даних

На даний момент найбільш популярні інструменти аналітичного ПЗ надають широкий спектр механізмів класифікації даних, методів статистичного аналізу, засобів кластеризації і сегментації, інструментів візуалізації, а також пакети для аналізу текстів (Text Mining) і пошуку інформації (Information Retrieval). До числа універсальних сучасних інструментів інтелектуальної обробки даних можна віднести наступний інструментарій:

Apache OpenNLP [19]. Інтегрований пакет інструментів обробки тексту, що працюють на основі машинного навчання. Пакет працює на платформі Java і містить рішення більшості основних завдань обробки природної мови, зокрема засоби токенізаціі тексту, розбиття на пропозиції, морфологічної розмітки, вилучення іменованих сутностей, синтаксичного розбору пропозиції та ін. Як правило, ці завдання активно застосовуються при побудові складних систем обробки тексту. До складу OpenNLP включені інструменти машинного навчання на основі як методів максимальної ентропії, так і на основі персептрону.

SAP HANA [20]. Надає єдину платформу для вилучення та аналізу великого обсягу структурованих і неструктурованих даних в реальному часі з різних джерел: соціальні мережі, блоги, онлайн-огляди, повідомлення електронної пошти та обговорення на форумах. Текстова аналітика в SAP HANA — це набір таких лінгвістичних та статистичних інструментів, як морфологічна розмітка, витяг іменованих сутностей, витяг семантичних відносин, аналіз тональності, оцінка точності і повноти і інші.

IBM Watson [19]. Інструмент пропонує широкий спектр можливостей для Data Mining і Text Mining: розпізнавання природної мови, динамічне навчання системи, побудова та оцінка гіпотез.

Polyanalyst. Багатофункціональний набір, що підтримує широкий спектр алгоритмів Data Mining. Останні версії включають до свого складу аналіз текстів, ліс рішень, аналіз зв'язків. Є підтримка технологій OLE DB for Data Mining і DCOM.

Вільно розповсюджуваний пакет програм Stanford CoreNLP. Являє собою набір алгоритмів машинного навчання для вирішення завдань інтелектуального аналізу даних. Stanford CoreNLP реалізований на Java і запускається практично з усіх платформ. Переважно розробляється для роботи з англійським, але так само підтримує арабську, китайську, французьку та німецьку. Разом з бібліотекою окремим пакетом доступний набір моделей мов.

2.1.4 Метод виявлення причинно-наслідкових зв'язків

На основі аналізу методів вилучення причинно-наслідкових зв'язків і аналізу когнітивних сервісів IBM Watson і бібліотеки StanfordCoreNLP розроблений підхід поділяє процес на дві основні процедури подібно методу Соргенте, Веттіглі і Мале:

– Витяг зв'язків на основі правил. Для вилучення ключових слів з тексту використовується сервіс Natural Language Understanding і StanfordParser.

– Класифікація витягнутих зв'язків на причинно-наслідкові і не причинно-наслідкові. Для виконання даної процедури використовується сервіс Natural Language Classifier і Stanford Classifier.

Процес вилучення даних пропонується розділити на етапи, проілюстровані на рис. 2.2.

Рисунок 2.2 — Алгоритм виявлення причинно-наслідкових зв'язків

Першим етапом вирішення задачі автоматичної вилучення даних з текстів є перетворення тексту, що надходить. Мета попередньої обробки даних полягає в перетворенні до структурованого формату неструктурованого документу для конкретної предметної області.

Існує велика кількість лінгвістичних та математичних методів, що дозволяють витягати сутності, але, як правило, поділяють ці методи на два підходи. Перші підходи ґрунтувалися на складених вручну правилах, що вимагало великих знань в граматиці мови і робило таку систему орієнтованої на обмежену кількість мов, або складанні списків розглянутих слів у довідниках, основним недоліком якого була необхідність в їх постійної підтримки та оновленні.

Загальний принцип, застосований для вирішення завдання вилучення сутностей, можна розбити на наступні кроки:

– визначення ознак;
– навчання сегментатора і класифікатора сутностей на основі розмічених даних;
– підрахунок значень ознак для всіх токенів в досліджуваному тексті;
– виділення набору кандидатів за допомогою навченого сегментатора;
– підрахунок значень ознак для кандидатів до сутності, отримані з попереднього пункту;
– класифікація кандидатів до сутності.

У канонічному вигляді на вхід задачі вилучення сутностей подається пропозиція з попереднього кроку, а результатом даного етапу будемо вважати виділення множини сутностей з тексту. У разі NLU ключові слова витягуються автоматично сервісом, а при використанні StanfordParser застосовується правило.

Під отриманням зв'язку з тексту на природній мові мається на увазі, що пара витягнутих сутностей знаходяться в безпосередній близькості або ж є частиною одного і того ж пропозиції. Сформулюємо задачу даного етапу наступним чином: на вході є речення і множина сутностей, витягнуті з розглянутого речення. Потрібно визначити, чи існує зв'язок між парою сутності і згенерувати триплет, якщо зв'язок витягнутий.

Класифікація використовується для віднесення кожного документа до певного класу із заздалегідь відомими ознаками, отриманими на етапі навчання. У сучасних системах класифікація застосовується, наприклад, в таких завданнях: угруповання документів за загальними ознаками, розміщення документів в певні папки, виборче поширення новин передплатникам. Відповідно до [23] існують такі види зв'язків:

– причина-наслідок (Cause-Effect);
– інструмент-агент (Instument-Agency);
– продукт-виробник (Product-Producer);
– сутність-джерело (Entity-Origin);

– повідомлення-тема (Message-Topic);

– частина-ціле (Part-Whole);

– контент-контейнер (Content-Container).

Розроблений метод орієнтований на реалізацію зв'язку Причина-Слідство і, отже, він дозволяє будувати бінарний класифікатор із заданими класами: causal-relation і not-causal-relation.

У нашому випадку завдання класифікації зв'язків слід поділити на два основних етапи: побудова моделі і класифікація триплету на основі результату попереднього кроку. На першому етапі будуємо модель або класифікувальну функцію за допомогою навчання на прикладах, яка могла б поділити навчальну вибірку. Для побудови даної класифікувальної функції використовуються NLC і Stanford Classifier. Далі, відбувається класифікація триплету, отриманого на вході даного компонента, до двох наперед визначених класів: causal-relation і not-causal-relation.

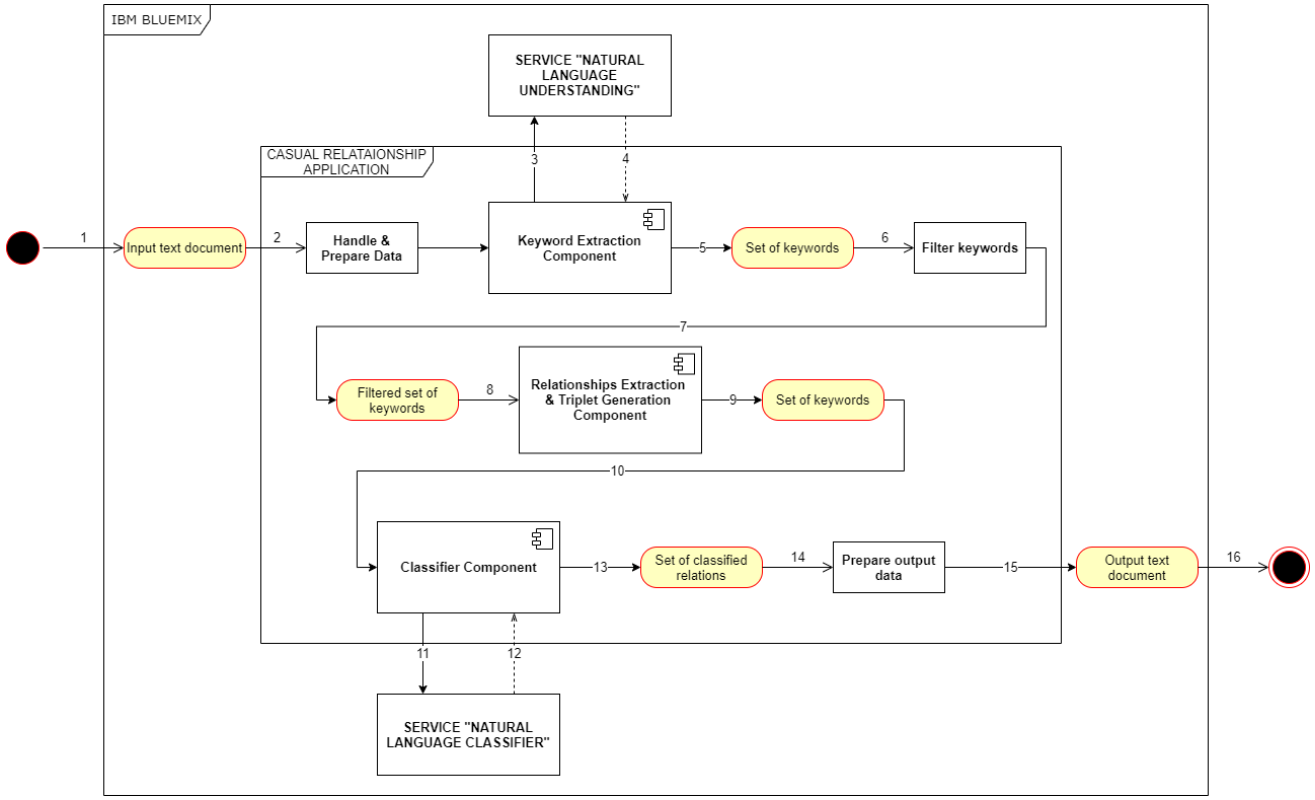


Рисунок 2.3 — Алгоритм виявлення причинно-наслідкових зв'язків.

Для даної задачі був обраний підхід навчання з учителем. В результаті класифікації отримуються ймовірні оцінки відповідності триплету до кожного класу. Отже, триплет буде співвідноситися того класу, у якого більша ймовірнісна оцінка.

Алгоритм виявлення причинно-наслідкових (рис. 2.3) зв'язків складається з наступних етапів:

а) виявлення пропозицій;

б) витяг ключових слів;

в) фільтрація ключових слів;

г) витяг відносин між парою ключових слів і генерація триплету;

ґ) класифікація триплету;

д) висновок результатів.

2.2 Опис методів та засобів реалізації тематичної класифікації тексту

Класифікація є основною формою аналізу тексту і широко використовується в різних областях. В основі класифікації лежить проста ідея: вивчити існуючі зв'язки між екземплярами, що складаються з незалежних змінних, і цільової категоріальної змінної. Оскільки мета відома заздалегідь, класифікацію називають машинним навчанням з учителем, і модель можна навчити для мінімізації помилки між передбаченими і фактичними категоріями навчальних даних. Після навчання модель класифікації зможе привласнювати категорії новим екземплярів, спираючись на шаблони, виявлені в процесі навчання [3].

Ця проста ідея має широкий спектр застосування за умови, що прикладну задачу можна сформулювати в термінах «так / ні» (двійкова класифікація) або групи дискретних категорій (багатокласова класифікація).

Найскладніша частина прикладного аналізу тексту — кураторство та збір предметно-орієнтованого корпусу для побудови моделей. Друга за складністю частина - вироблення аналітичного вирішення конкретної прикладної задачі

2.2.1 Алгоритми класифікації з вчителем

Алгоритми класифікації з вчителем сортують повнотекстові документи по заздалегідь відомим категоріям (класах). У ролі вчителя виступає вибірка документів, для яких заздалегідь відома приналежність тієї чи іншої категорії, що має назву навчальної множини [21]. Множину категорій $\mathcal{C} = \{c_j\}, j = \overline{1, |\mathcal{C}|}$ і навчальну множину документів $\Omega \subset \mathcal{D}$, де $\mathcal{D} = \{d_i\}, i = \overline{1, |\mathcal{D}|}$ — повна множина документів, формують експерти. Алгоритм класифікації з вчителем — алгоритм категоризації — використовує навчальну множину Ω, щоб побудувати класифікатор $\Phi: \mathcal{D} \times \mathcal{C} \to \{\text{істина}, \text{брехня}\}$, що забезпечує високу точність на всій множині документів $\mathcal{D}$, використовуючи припущення, що навчальні та нові дані схожі. Зазвичай множину документів Ω поділяють на дві частини: одна частина — дані для навчання алгоритму, друга — тестові дані для оцінки якості отриманого класифікатору.

2.2.2 Представлення даних у задачах класифікації текстів

Вхідними даними алгоритму класифікації є множина образів кожного документа $\vec{\mathcal{D}} = \{\vec{d}_i\}, i = \overline{1, |\mathcal{D}|}$, де $\vec{d}_i \in \vec{\mathcal{D}}$ — образ документа $d_i \in \mathcal{D}$ існує декілька підходів формування образів, застосовують той, що відповідає моделі, покладеної до основи конкретного алгоритму класифікації [22]. Образи документів у тих алгоритмах, які будуть далі розглядатися, представлені в наступному вигляді:

а) мультімножин термінів документів (наприклад, наївний байесовський класифікатор);

б) векторів у просторі термінів (алгоритми класифікації без вчителя, наприклад, алгоритм Роккі).

Під термінами документів будемо розуміти всі поодинокі слова, які зустріли в тексті хоча б одного документа, за винятком стоп-слів, тобто слів, які не характеризують документи за змістом (прийменники, спілки і тому подібне). До того ж, кожній зустрінутої формі слова, наприклад, у різних числах і відмінках, буде відповідати один і той же термін. В результаті отримуємо множину всіх термінів колекції $\mathcal{T} = \{t_k\}, k = \overline{1, |\mathcal{T}|}$.

Образом документа як вектору в просторі термінів є вектор дійсних чисел $\vec{d}_l = \left(d_{i1}, \dots, d_{i|T|}\right)^T$, де кожне дійсне число є координатою вектору, що відповідає даному терміну, і дорівнює вазі терміну в конкретному документі. Дуже часто використовують підхід до обчислення ваги терміну:

$$d_{ij} = \frac{w_{ij}}{\|\vec{w}_l\|}, w_{ij} = tf_{ij} \times \log\frac{|\mathcal{D}|}{df_j} \tag{2.1}$$

де $\|\vec{w}_l\|$ — евклідова норма $\vec{w}_l$;

tf_{ij} — частота терміну в документі;

df_j — документна частота (кількість документів, в яких зустрівся j-ий термін).

Такі ваги d_{ij} називають нормованими вагами за формулою «TF-IDF» («частота терміну — зворотна документна частота»), 0 <d_{ij} <1. Вони мають такі властивості [3]:

– високі значення, коли термін зустрічається в документах часто, тим самим посилюючи відміну документів одне від одного;

– низькі значення, коли термін зустрічається в якомусь документі зрідка, чи зустрічається у великій кількості документів, тим самим знижуючи їх відмінність.

Процес класифікації документів як векторів заснований на гіпотезі, що документи тематично близькі виявляться в просторі термінів геометрично близько розташованими. Тому в основі алгоритмів класифікації лежить поняття подібності або відстані між документами в просторі термінів.

Поняття подібності і відстані є взаємнозворотніми, і може називатися різницею. На результат класифікації впливає вибір способу обчислення відстані. Часто застосовують такі варіанти [3]:

$$dist(\vec{d}_i, \vec{d}_j) = \left(\sum_{k=1}^{|\mathcal{T}|} |d_{ik} - d_{jk}|\right)^{1/r} \quad (2.2)$$

де r — це параметр, заданий користувачем, $r \in \mathcal{R}, r > 0$.

Поширені приклади:

а) при r = 1: манхеттенська відстань, або відстань міських кварталів;

б) при r = 2: евклідова відстань;

в) при $r \to \infty$ отримаємо відстань Чебишева, яку можна обчислити як максимум модуля різниці компонент векторів:

$$dist(\vec{d}_i, \vec{d}_j) = max_{k=1\ldots|\mathcal{T}|} |d_{ik} - d_{jk}|$$

Інший часто використовуваний на практиці ступенем подібності є косинусною мірою:

$$sim(\vec{d}_i, \vec{d}_j) = \cos\left(\angle(\vec{d}_i, \vec{d}_j)\right) = \frac{\sum_{k=1}^{|\mathcal{T}|} d_{ik} d_{jk}}{\sqrt{\sum_{k=1}^{|\mathcal{T}|} d^2{}_{ik}} \times \sqrt{\sum_{k=1}^{|\mathcal{T}|} d^2{}_{jk}}} \quad (2.3)$$

Якщо вектори ваг документів нормовані як в (2.1), то косинусна міра виступає скалярним добутком векторів. Якщо вектори збігаються, то міра близькості дорівнює 1, якщо ортогональні — 1.

Коли вектори ваг термінів нормовані, значення евклидової відстані і косинусні заходи відповідають один одному.

2.2.3 Відбір термінів для класифікації

В завданні класифікації велика кількість ознак (термінів) документів призводить до певних проблем, серед яких [23]:

– високі обчислювальні витрати, пов'язані, наприклад, з отримання значень міри близькості між документами та ін.;
– низька якість класифікації, викликане наявністю великого числа шумових ознак (ознак зі слабкою класифікаційної здатністю).

Зокрема, (2.2) в класифікації з вчителем може призвести до перенавчання, ефекту, який виникає, якщо класифікатор налаштовувався на шумових (випадкових) характеристиках документів. У такій ситуації алгоритм значно гірше працює на нових, і добре, на яких він був навчений. Тому, прагнуть зменшити число термінів з множини T так, щоб скорочена множина термінів $\mathcal{T}'(|\mathcal{T}'| \ll |\mathcal{T}|)$ містила найбільш інформативні в деякому сенсі терміни.

Існує два способи застосування зменшення розмірності простору термінів: локально (скорочують кількість термінів для кожної категорії окремо) і глобально (працюють із загальним множиною терміном для всіх документів). Перший випадок застосуємо для класифікації з вчителем, другий — як для класифікації з вчителем, так і без нього.

Документна частота (DF) $\mathcal{T} = \{t_k \in \mathcal{T}: DF(t_k) > \tau\}$, де $\mathcal{D}F(t_k)$ — це кількість документів, в яких зустрічається термін t_k. Найпростіша і ефективна техніка оцінки термінів (важливості термінів для класифікації) полягає у спостереженні того, що значна кількість термінів зустрічаються в малому числі документів. На практиці часто використовують порогове значення τ, яке дорівнює 1-5 документам.

Наступні техніки: взаємна інформація, інформаційна вигода і критерій хі-квадрат. Будуть розглянуті їх локальні значення $f(t_k, c_j)$, щоб отримати значення глобально (незалежно від конкретної категорії), слід обчислити або просту суму $\sum_{j=1}^{|C|} f(t_k, c_j)$, або зважену суму $\sum_{j=1}^{|C|} P(c_j) f(t_k, c_j)$, або знайти максимум $max_{j=\overline{1,|C|}} f(t_k, c_j)$. Взаємна інформація (MI). Величина взаємної інформації терміну *t* і категорії *c*:

$$MI(t_k, c_j) = \log_2 \frac{P(t_k, c_j)}{P(t_k) \times P(c_j)} \quad (2.4)$$

Тоді вираз (2.4) можна записати в такий спосіб:

$$MI(t_k, c_j) = \log_2 \frac{A \times |\Omega|}{(A+C) \times (A+B)} \quad (2.5)$$

де A — кількість документів, яка належать категорії *c* і має термін *t*;

Ω — навчальна множина документів;

C — кількість документів, яка належать категорії *c* і не має термін *t*;

B — кількість документів, що не належать категорії *c* і має термін *t*.

$MI(t_k, c_j)$ приймає значення 0, якщо термін *t* і категорія *c* незалежні.

Недолік полягає в тому, що значення взаємної інформації сильно піддається впливу безумовної ймовірності термінів, оскільки $Ml(t_k, c_j) = log_2 P((t_k, c_j) - log_2 P(t_k))$ (це випливає з (2.4)). Якщо два терміни мають одну ймовірність, більш високе значення *Ml* буде у більш рідкісного. Отже, значення взаємної інформації непорівнянні для термінів з істотно розрізняється частотою зустрічальності в документах [5].

Інформаційна вигода (IG). Інформаційну вигоду часто називають очікуваної взаємної інформацією (EMI). Цей показник вимірює кількість інформації про приналежність категорії *c*, яке несе наявність / відсутність терміну *t*.

$$IG(t_k, c_j) = \sum_{c\in\{c_j,\bar{c}_j\}}\sum_{t\in\{t_k,\bar{t}_k\}} P(t,c) \times \log_2 \frac{P(t,c)}{P(t)\times P(c)} \quad (2.6)$$

де $\bar{c}_j$ — всі категорії, крім c_j;

$t_k, \bar{t}_k$ — ознаки наявності і відсутності терміну t_k відповідно.

На практиці формула (2.6) еквівалентна наступній:

$$IG(t_k, c_j) = \frac{A}{|\Omega|} \times \log_2 \frac{|\Omega|\times A}{(A+C)\times(A+B)} + \frac{C}{|\Omega|} \times \log_2 \frac{|\Omega|\times C}{(C+D)\times(A+C)} + \frac{B}{|\Omega|} \times \log_2 \frac{|\Omega|\times B}{(A+B)\times(B+D)} + \frac{D}{|\Omega|} \times \log_2 \frac{|\Omega|\times D}{(C+D)\times(B+D)} \quad (2.7)$$

де D — кількість документів, що не належать до категорії *c* і не має термін *t*.

Ступінь вигоди досягає максимуму, якщо термін є ідеальним індикатором категорії. Але коли розподіл терміну в категорії відповідає розподілу терміну в колекції, то інформаційна вигода дорівнює 0.

За заданої навчальної множині для кожного терміну обчислюють значення *IG* і видаляють з $\mathcal{T}$ такі терміни, значення інформаційної вигоди яких нижче деякого заздалегідь обраного порогового значення [3].

Критерій хі-квадрат (CHI). Критерій χ^2 використовується для перевірки незалежності двох випадкових подій: поява терміну X і поява класу Y. При X і Y незалежні, P(XY) = P(X)P(Y). Критерій χ^2 обчислюється за формулою:

$$CHI(t_k, c_j) = \sum_{c\in\{c_j,\bar{c}_j\}}\sum_{t\in\{t_k,\bar{t}_k\}} \frac{\left(P(t,c) - P_{exp}(t,c)\right)^2}{P_{exp}(t,c)} \quad (2.8)$$

де *P(t, c)* — спостерігається на навчальній множині;

P_{exp}*(t, c)* — очікувана при умові, що клас і термін незалежні.

На практиці формула (2.8) еквівалентна наступній:

$$CHI(t_k, c_j) = \frac{|\Omega| \times (A \times D - C \times B)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)} \quad (2.9)$$

На відміну від взаємної інформації критерій χ^2 нормалізований, це дозволяє порівнювати значення для різних термінів однієї категорії, винятком є лише рідкісні терміни.

Є необхідність синтезувати нові (штучні) ознаки документів, для підвищення якість класифікації, , шляхом дозволу неоднозначностей природної мови, наприклад, синонімії, омонімії, полісемії. Потім слід відобразити документи колекції в нове простору ознак, яка позбавлена старих проблем і краще, ніж вихідне, представляє зміст документів. Прикладами технік вилучення ознак документів є латентно-семантичне індексування і кластеризація термінів документів.

2.2.4 Алгоритм наївної байєсівської класифікації

Алгоритм наївної байєсівської класифікації використовує формулу Байеса для оцінки ймовірності приналежності документа класу на навчальній множині. Образ документа розглядається як мультімножина термінів.

Метою класифікації є пошук найкращого класу для документу, тобто має найбільшу апостеріорну ймовірність $P(c_j|d_i)$:

$$c^* = arg_{c_j \in \mathcal{C}} max P(c_j|d_i) \quad (2.10)$$

де $c_j \in \mathcal{C}$, $d_i \in \Omega$.

За формулою Байєса:

$$P(c_j|d_i) = \frac{P(c_j) \cdot P(d_i|c_j)}{P(d_i)} \quad (2.11)$$

де $P(c_j)$ — апріорна ймовірність, що документ належить c_j;

$P(d_i|c_j)$ — ймовірність зустріти документ типу d_i серед документів класу c_j.

Оскільки $P(d_i)$ не впливає на вибір класу, підсумкове ранжування класів по апріорної ймовірності можна провести без урахування знаменника у формулі (2.11).

Наївним даний алгоритм називають тому, що він використовує наївне припущення, що слова, що входять до тексту документа, не залежать одне від одного [11].

Отже, $P(c_j|d_i)$ можна обчислити як добуток ймовірностей зустріти термін t_k документах класу c_j:

$$P(c_j|d_i) = \prod_{k=1}^{|T_{d_i}|} P(t_k|c_j) \quad (2.12)$$

де T_{d_i} — множина термінів документу d_i;

$P(t_k|c_j)$ — оцінка терміна t_k вкладу до того, що $d_i \in c_j$.

Тоді вирішальне правило (2.10) приймає остаточний вигляд:

$$c^* = arg_{c_j \in C} max P(c_j) \prod_{k=1}^{|T_{d_i}|} P(t_k|c_j) \quad (2.13)$$

На практиці може спостерігатися втрата значущих розрядів при множенні $|T_{d_i}|$ умовних ймовірностей. Тоді у виразі (2.13) замість самих оцінок ймовірностей використовують логарифм цих ймовірностей. Оскільки логарифм — монотонно зростаюча функція, то клас c_j з найбільшим значенням логарифма ймовірності залишиться найбільш імовірним. Тоді [7]:

$$c^* = arg_{c_j \in \mathbb{C}} max \left[\log P(c_j) + \sum_{k=1}^{|T_{d_i}|} \log P(t_k|c_j) \right] \quad (2.14)$$

Оцінки ймовірностей на навчальній множині:

$$P(c_j) = \frac{\left|\mathcal{D}_{c_j}\right|}{|\mathcal{D}|}$$
$$P(t_k|c_j) = \frac{tf(t_k,c_j)}{\sum_{i=1}^{|\mathcal{T}|} tf(t_k,c_j)} \quad (2.15)$$

де $\mathcal{D}_{c_j}$ — множина документів в класі c_j;

$\mathcal{D}$ — кількість всіх документів ($\mathcal{D} = \Omega$);

$tf(t_k, c_j)$ — кількість входжень терміна t_k в документі класу c_j;

$\mathcal{T}$ — словник всієї колекції документів.

Оскільки навчальна множина не може бути достатньо великою, щоб утримувати всі терміни, які можуть зустрітися в нових документах, тоді якщо новий документ, містить новий (рідкісний) термін, то ймовірність приналежності своєму класу буде дорівнює нулю (випливає з (2.15) і (2.12)). Для вирішення цієї проблеми на практиці застосовують згладжування, наприклад, такого вигляду:

$$P(t_k|c_j) = \frac{tf(t_k,c_j)+1}{\sum_{i=1}^{|\mathcal{T}|}(tf(t_k,c_j)+1)} = \frac{tf(t_k,c_j)+1}{\sum_{i=1}^{|\mathcal{T}|} tf(t_k,c_j)+|\mathcal{T}|} \quad (2.16)$$

Додавання одиниці до кожної частоті терміна можна інтерпретувати як апріорне рівномірний розподіл (кожен термін зустрічається в кожному класі по одному разу), яке потім на навчальній множині уточняється.

Алгоритм в загальному вигляді.

Навчання.

Вхід: $\mathcal{C}$ і $\mathcal{D}$ = Q.

Крок 1. Скласти словник $\mathcal{T}$ з $\mathcal{D}$.

Крок 2. Для кожного $c_j \in \mathcal{C}$:

Крок 3. $prior[j] := \left|\mathcal{D}_{c_j}\right| / |\mathcal{D}|$;

Крок 4. *text [j]*: = <склеїти тексти всіх документів $d_i \in \mathcal{D}: d_i \in c_j$>;

Крок 5. Для кожного $t_k \in \mathcal{T}$:

Крок 6. *tf [k] [j]*: = <обчислити кількість входжень терміна t_k в *text [j]*>;

Крок 7. Для кожного $t_k \in \mathcal{T}$:

Крок 8. *cp [k] [j]*: = $(tf[k][j] + 1)/\left(\sum_{i=1}^{|\mathcal{T}|} tf[k][j] + |\mathcal{T}|\right)$;

Вихід: $\mathcal{T}$, *prior*, *cp*.

Тестування (застосування). Вхід: $\mathcal{C}$; $\mathcal{T}$; *prior; cp*; $d \in \mathcal{D}$.

Крок 1. terms: = <витягти терміни з d з урахуванням $\mathcal{T}$ і доповненнями *cp*>;

Крок 2. Для кожного $c_j \in \mathcal{C}$:

Крок 3. *score [j]: = log prior [j]*;

Крок 4. Для кожного $c_j \in \mathcal{C}$:

Крок 5. *score [j]: = score [j] + log cp [k] [j];*

Крок 6. *c *: = arg$_j$ max score [j];*

Вихід: *c **.

2.2.5 Етапи автоматичної класифікації текстової інформації

Процес побудови класифікатору текстової інформації складається з таких етапів (рис. 2.4):

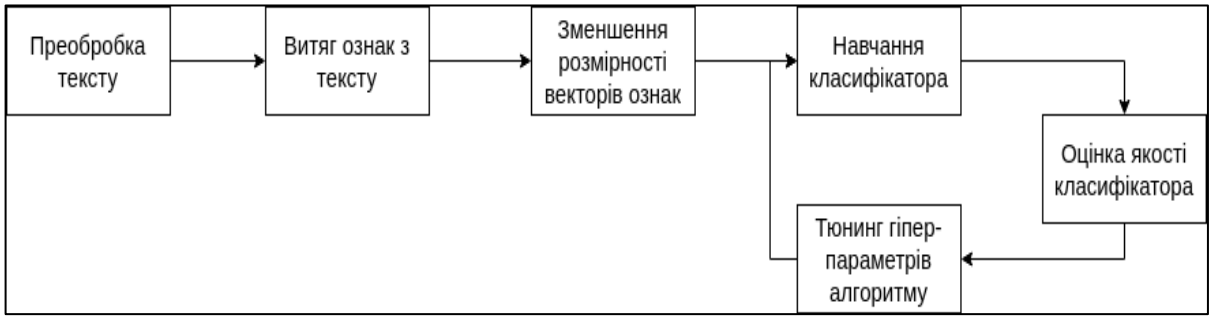


Рисунок 2.4 — Процес побудови класифікатора

Процес передобробки (рис. 2.5) тексту є необхідним та важливим етапом в задачах обробки текстів на природній мові. Він може використовуватися для виокремлення необхідної інформації з неструктурованих текстових даних [24].

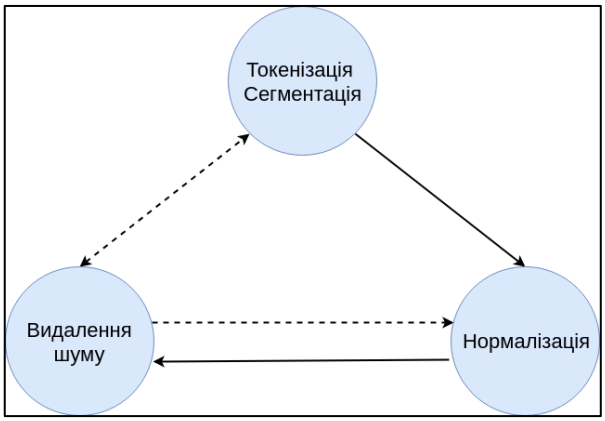


Рисунок 2.5 — Процес предобробки тексту

Токенізація — це етап, який розбиває рядки тексту на більш дрібні фрагменти або токени. Токенам може бути як речення, так і фрази, n-грами, слова, літери.

Текст потрібно нормалізувати перед подальшою обробкою. Нормалізація тексту — це процес перетворення тексту в єдину канонічну форму, яку він міг би не мати раніше. Існують такі популярними методи нормалізації як лемінг, стемінг та видалення стоп-слів [25].

Стемінг — це процес знаходження основи слова для заданого вихідного слова (рис. 2.6). Основа слова не обов'язково збігається з морфологічним коренем слова. Стемінг допомагає стандартизувати слова в їх основі й корені, незалежно від їх форм, що допомагає при класифікації чи кластеризації тексту, та навіть при пошуку інформації.

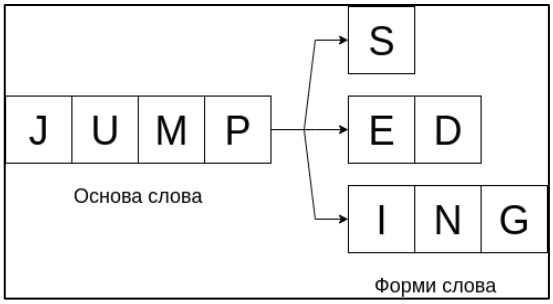


Рисунок 2.6 — Приклад результату стемінга

Лематизація — метод морфологічного аналізу, він зводиться до приведення словоформи до її первісної словникової формі (рис. 2.7).

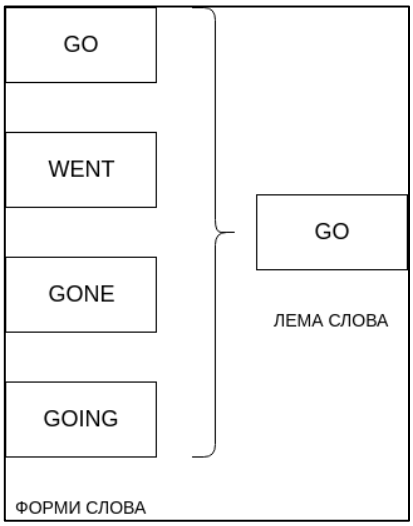


Рисунок 2.7 — Приклад результату лематизації

Відомо, що стоп-слова не роблять внесок в зміст або контекст текстових документів. Тому їх потрібно видалити. Процес видалення стоп-слів зменшує текстові дані і покращує продуктивність системи.

2.2.6 Опис методів побудови ознак

Усі слова, що наявні у корпусі документів, складають словник корпусу. Тому документ може бути представлений у вигляді бінарного вектору, де значення «1» означає, що певне слово є у цьому тексті, а «0» – що його не має. Проте на практиці підхід побудови словника з усіх слів зустрічається зрідка. Це пов’язано з тим, що його розмір може складати сотні тисяч чи навіть мільйони слів. Такі об’єми даних є надзвичайно великими для їх обробки та можуть негативно вплинути на точність алгоритму та сприяти його перенавчанню.

Bag-of-Words. Модель Bag-of-Words (або BoW) — це спосіб вилучення ознак з тексту для використання в моделюванні, наприклад, з алгоритмами машинного

навчання. Її названо так, тому що будь-яка інформація про структуру або порядок слів у документі ігнорується, тобто враховується тільки те зустрічаються в документі відомі слова чи ні, але не їх розташування [26].

Модель Bag-Of-Words досить проста для розуміння та реалізації, вона пропонує гнучкість, оскільки вона може бути налаштована під конкретну задачу. Вона використовується з великим успіхом у задачах обробки природної мови, таких як моделювання мови та класифікація документів.

Вона має певні суттєві недоліки:

– словник вимагає ретельного проектування, зокрема, його вузьким місцем є його обсяг, що впливає на обмеженість використання текстових даних;
– розрідженість представлення важче моделювати як по причинам просторової складності, так і за критерієм інформації, коли завдання полягає в тому, щоб моделі використовували малу кількість інформації в такому великому просторі представлення;
– відкидання порядку слів ігнорує контекст і значення слів в документі (семантику). Врахування контексту та сенсу можуть сильно покращити ефективність моделі.

TF-IDF метрики**.** Це показник важливості слова у певному контексті. Складається с двох складових, одна обраховує важливість слова у межах документа, а друга — інверсує частоту скільки зустрічається слово у корпусі.

Перша складова обраховується за формулою:

$$F(t,d) = \frac{n_t}{\sum_k n_k} \tag{4.17}$$

де $n_t -$ кількість входжень слова t в документ d;

$\sum_k n_k -$ загальна кількість слів у документі.

За допомогою другого показника зменшується кількість загальновживаних слів. Показник обраховується за формулою:

$$I(t, D) = \frac{|D|}{|\{d \in D : t \in d\}|} \quad (4.18)$$

де $|D|$ – кількість документів у корпусі;

$|\{d \in D : t \in d\}|$ – кількість документів з корпусу D, де зустрічається слово t.

Фінальна формула цього показника виглядає так:

$$T(t, d, D) = F(t, d) \times I(t, D) \quad (1.3)$$

де t — слово;

d — документ;

D — корпус текстів;

F(t, d) — показник частоти слова;

I(t, D) — показник оберненої частоти слова.

Найбільшу вагу мають слова з високою частотою в межах певного документа та з низькою частотою вживання в інших. Важливість слів визначається їх здатністю характеризувати відмінність між категоріями в корпусі документів. Метрика допомагає скласти словник вагомих слів та представити документи корпусу у вигляді числових векторів.

Word2Vec. Word2Vec — це модель, представлена компанією Google, яка навчена на великих корпусах текстових даних та представляє слова у вигляді вектора у 300-вимірному просторі. Її особливістю є те, що близькі за семантикою слова знаходяться поруч один з одним. Модель навчається за допомогою двох алгоритмів — моделі “Continuous Bag-of-Words”(CBOW) та моделі Skip-Gram. Google надає можливість використовувати натреновані Word2Vec у 100-вимірному та 300-вимірному просторі (рис. 2.8), також можна підлаштувати модель під конкретну задачу, дотренувавши її на власних корпусах текстів.

При вирішенні поставленої задачі модель word2vec у 300-вимірному просторі було дотреновано, використовуючи записи користувачів Facebook та Twitter на політичну тематику. Розмір корпусу — 3 Гб текстових даних. На рисунку

2.8 наведено сформований 300-вимірний векторний простір слів, близький за семантикою.

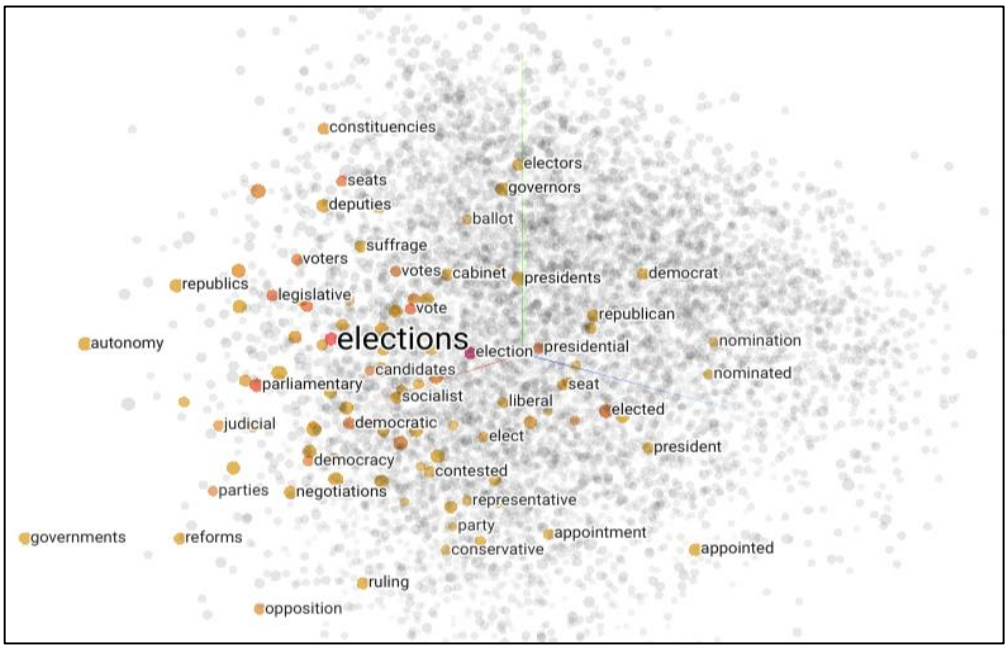


Рисунок 2.8 — Векторний простір слів [5]

Part-Of-Speech Tagging (PoS) — це алгоритм, який маркує слова як одну з декількох категорій, щоб ідентифікувати його функцію. В англійській мові слова потрапляють в одну з восьми або дев'яти частин мови. Категорії частини мови включають іменник, дієслово, артиклі, прикметник, прийменник, займенник, прислівник, сполучник і вигук.

Алгоритми PoS використовують алгоритми для маркування слів в текстових корпусах. Ці тегер роблять більш складні категорії, ніж ті, які визначені як базові PoS, з такими тегами, як «однина-множина» або навіть більш складними мітками.

Алгоритми PoS класифікують терміни в типах PoS по їх реляційної позиції у фразі, співвідношенні з близькими слова і за визначенням слова. Алгоритми PoS поділяються на ті, які базуються на стохастичних методах, на ймовірностних моделях та на правилах.

Алгоритми PoS класифікують терміни в типах PoS по їх реляційної позиції у фразі, співвідношенні з близькими слова і за визначенням слова. Алгоритми PoS

поділяються на ті, які базуються на стохастичних методах, на ймовірностних моделях та на правилах.

2.3 Огляд використовуваних засобів програмування

2.3.1 Огляд мови програмування Python

Python — це мультіпарадигмальна високорівнева мова програмування загального призначення. Орієнтований на підвищення швидкості написання і читання коду.

Відмінні риси Python:

– лаконічність;
– високий рівень абстракції;
– може застосовуватися практично в будь-якій предметній області;
– велика кількість модулів розширення Python;
– динамічна типізація.

Як зазначалося вище, Python — це мова загального призначення. Проте в кількох сферах Python застосовується частіше і успішніше всього.

Python — один з основних мов програмування, які застосовують в області машинного навчання і штучного інтелекту (Machine Learning і Artificial Intelligence). Наприклад, бібліотека з відкритим вихідним кодом TensorFlow, створена дослідницької командою Google Brain, написана з використанням Python. Google використовує цю бібліотеку для програмування і навчання нейронних мереж, які використовуються для вивчення штучного інтелекту.

Ще одна відома бібліотека — scikit-learn. Вона написана на Python з включеннями Cython — статично типізований компилируемого підмножини Python. Бібліотека scikit-learn застосовується в дослідженнях штучного інтелекту, для навчання інженерів machine learning, для управління промисловими системами.

В Python є кілька потужних і популярних бібліотек, які призначені для роботи з великими даними: аналізу, візуалізації, прогнозування тенденцій. Наприклад, бібліотека з відкритим вихідним кодом SciPy включає модулі для математичних, інженерних і наукових обчислень. Matplotlib — одна з найпопулярніших бібліотек для візуалізації даних. Бібліотека PANDAS застосовується для аналізу інформації.

Один із способів оцінки популярності мови програмування — індекс TIOBE. Він розраховується на основі кількості пошукових запитів в Google та інших пошукових системах. Враховуються запити, що включають назву мов програмування.

2.3.2 Огляд програмного середовища Anaconda

В якості програмного середовища буде використовуватися Anaconda, що є вільно та відкрито розповсюджуваний дистрибутивом різних програмних продуктів, зокрема, мов програмування Python та R. Платформа спеціалізується на "наукових обчисленнях": наука про дані, застосуванні методів машинного навчання, широкомасштабна обробка даних, передбачувальна аналітика тощо. Використання платформи має на меті спрощення управління пакетами та їх розгортання.

Пакети з відкритим кодом можуть бути встановлені індивідуально зі сховища Anaconda , так і з Anaconda Cloud чи з вашого власного сховища. Anaconda Inc компілює та створює всі пакунки у самому сховищі Anaconda та надає бінарні файли для Windows 32/64 біт, Linux 64 біт та MacOS 64-біт.

Anaconda Navigator — це графічний інтерфейс користувача настільних ПК (GUI), що входить у дистрибутив Anaconda, який дозволяє користувачам запускати пов'язані програми та керувати пакетами, середовищами та каналами Conda без використання часто менш зручного командного рядка.

За замовчуванням у Навігаторі доступні такі програми: Orange, Glueviz, Rstudio, Spyder, QtConsole, Jupyter Notebook, JupyterLab, Visual Studio Code.

Jupyter Notebook (раніше IPython Notebooks) — це веб-інтерактивне обчислювальне середовище для створення документів для ноутбуків Jupyter.

Ядро Jupyter — це програма, яка відповідає за обробку різних типів запитів (виконання коду, заповнення коду, перевірку) та надання відповіді. Ядра спілкуються з іншими компонентами Jupyter за допомогою ZeroMQ , і тому можуть бути на тих самих або віддалених машинах. На відміну від багатьох інших інтерфейсів, подібних до ноутбуків, в Jupyter ядра не знають, що вони прикріплені до певного документа, і їх можна підключати відразу до багатьох клієнтів. Зазвичай ядра дозволяють виконувати лише одну мову, але є кілька винятків.

2.3.3 Існуючі бібліотеки для обробки природної мови

Stanford CoreNLP. Даний безкоштовний програмний продукт був створений спільними зусиллями студентів і науковців університету Stanford. Основний завданням, яка була поставлена на початку його розробки, було створення набору сучасних інструментів, що дозволяють обробляти неструктурований текст. До цього дня цей продукт є одним з кращих у своїй ніші. З його допомогою можна провести повний аналіз частин мови в тексті, структури тексту, провести розпізнавання іменованих об'єктів, визначити, де в тексті різні іменники позначають один і той же об'єкт, провести аналіз тональності тексту і багато іншого.

Natural Language Toolkit, NLTK. Дана безкоштовна бібліотека для мови програмування Python є однією з кращих для створення різних програмних продуктів цією мовою. Вона надає великий набір інструментів, корпусів тексту, має передбачені обгортки для використання інших бібліотек всередині себе.

Наприклад, для аналізу тональності тексту і розмітки пропозицій, є можливість підключення вищеописаного продукту Stanford CoreNLP. Також для різних класифікацій в NLTK був передбачений інтерфейс для підключень класифікаторів з іншої бібліотеки — Scikit learn, про яку піде мова далі. А більше інформації про використання, пристрої даної можна знайти на офіційному сайті.

Scikit learn. Хоча ця бібліотека не має ніяких специфічних інструментів для обробки природної мови, в ній є величезна кількість класифікаторів, заснованих на різних алгоритмах; моделей нейронних мереж та інших спільних інструментів для машинного навчання. використовуючи її разом з іншими, вузькоспрямованими інструментами, можна створювати дуже складні і якісні системи для обробки, аналізу, класифікації і навіть генерації природного тексту.

3. ОПИС ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

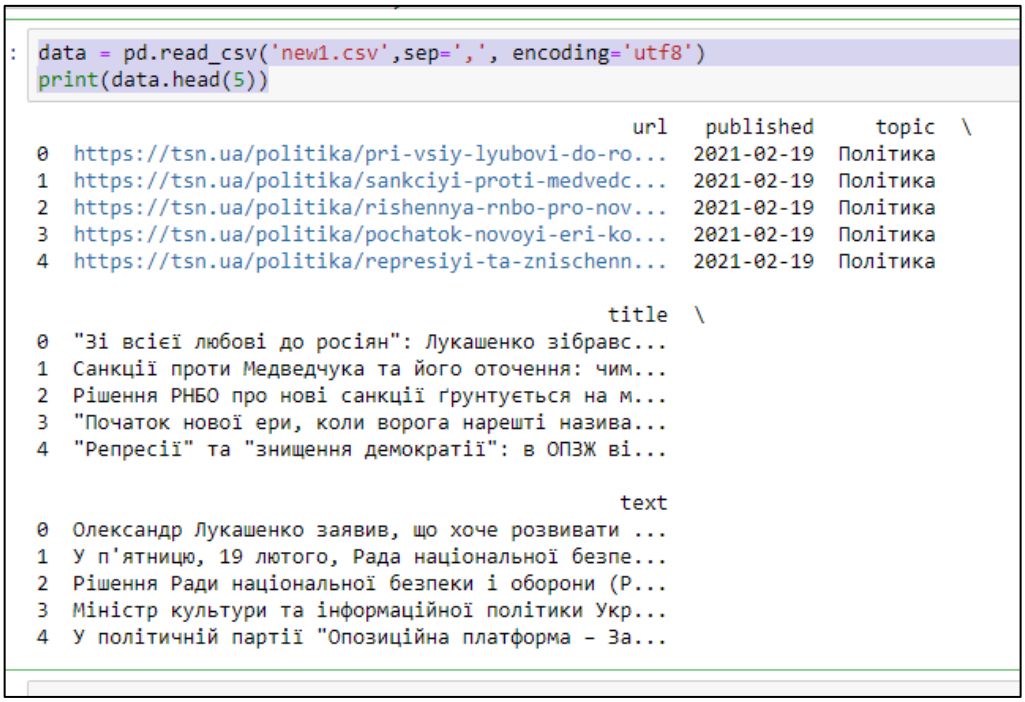
В даній програмі створеній на мові Python виконується навчання тематичного класифікатору, що працює за наївним байєсівським алгоритмом. Задача класифікатору відносити тексти новин до певної теми. Робота здійснюється виключно з україномовними текстами.

Для початку додаються бібліотеки, які будуть використовуватися:

```
import pandas as pd
import numpy as np
from tqdm.auto import tqdm, trange
import csv
```

Наступним кроком відбувається зчитування даних з підготовленого датасету, де зібрані новини телеканалу 1+1 з наглядним показом перших 5 даних (рис. 3.1).

```
data = pd.read_csv('new1.csv',sep=',', encoding='utf8')
print(data.head(5))
```



```
: data = pd.read_csv('new1.csv',sep=',', encoding='utf8')
  print(data.head(5))

                                                 url   published     topic  \
0  https://tsn.ua/politika/pri-vsiy-lyubovi-do-ro...  2021-02-19  Політика
1  https://tsn.ua/politika/sankciyi-proti-medvedc...  2021-02-19  Політика
2  https://tsn.ua/politika/rishennya-rnbo-pro-nov...  2021-02-19  Політика
3  https://tsn.ua/politika/pochatok-novoyi-eri-ko...  2021-02-19  Політика
4  https://tsn.ua/politika/represiyi-ta-znischenn...  2021-02-19  Політика

                                               title  \
0  "Зі всієї любові до росіян": Лукашенко зібравс...
1  Санкції проти Медведчука та його оточення: чим...
2  Рішення РНБО про нові санкції ґрунтується на м...
3  "Початок нової ери, коли ворога нарешті назива...
4  "Репресії" та "знищення демократії": в ОПЗЖ ві...

                                                text
0  Олександр Лукашенко заявив, що хоче розвивати ...
1  У п'ятницю, 19 лютого, Рада національної безпе...
2  Рішення Ради національної безпеки і оборони (Р...
3  Міністр культури та інформаційної політики Укр...
4  У політичній партії "Опозиційна платформа – За...
```

Рисунок 3.1 — Скріншот читання всіх даних файлу new1.csv з наглядним показом перших 5 даних

Встановлюється загальна кількість записів за допомогою команди:

```
data.shape
```

Таких записів всього 4671 розбитих за 5 категоріями: url, published, topic, title, text. (4671, 5)

Для подальшого аналізу обираємо 2000 записів з наглядним показом перших 5 даних (рис. 3.2)::

```
data = data[:2000]
data.head(10)
```

	url	published	topic	title	text
0	https://tsn.ua/politika/pri-vsiy-lyubovi-do-ro...	2021-02-19	Політика	"Зі всієї любові до росіян": Лукашенко зібравс...	Олександр Лукашенко заявив, що хоче розвивати ...
1	https://tsn.ua/politika/sankciyi-proti-medvedc...	2021-02-19	Політика	Санкції проти Медведчука та його оточення: чим...	У п'ятницю, 19 лютого, Рада національної безпе...
2	https://tsn.ua/politika/rishennya-rnbo-pro-nov...	2021-02-19	Політика	Рішення РНБО про нові санкції ґрунтується на м...	Рішення Ради національної безпеки і оборони (Р...
3	https://tsn.ua/politika/pochatok-novoyi-eri-ko...	2021-02-19	Політика	"Початок нової ери, коли ворога нарешті назива...	Міністр культури та інформаційної політики Укр...
4	https://tsn.ua/politika/represiyi-ta-znischenn...	2021-02-19	Політика	"Репресії" та "знищення демократії": в ОПЗЖ ві...	У політичній партії "Опозиційна платформа – За...
5	https://tsn.ua/politika/za-sankciyi-proti-medv...	2021-02-19	Політика	За санкції проти Медведчука та його оточення о...	Під час голосування за санкції проти нардепа в...
6	https://tsn.ua/politika/u-nas-zgidno-konstituc...	2021-02-19	Політика	"У нас згідно з Конституцією всі рівні": Даніл...	Секретар РНБО Олексій Даніловподякувавукраїнсь...
7	https://tsn.ua/politika/sankciyi-proti-medvedc...	2021-02-19	Політика	Санкції проти Медведчука: Кравчук підтримав рі...	19 лютого на засіданні Ради національної безпе...
8	https://tsn.ua/politika/rnbo-doruchiv-povernut...	2021-02-19	Політика	РНБО доручила повернути у держвласність частин...	Рада національної безпеки і оборони доручила К...
9	https://tsn.ua/politika/putin-zve-lukashenka-n...	2021-02-18	Політика	Путін кличе Лукашенка на килим у Сочі: чим Укр...	Хоч Лукашенко й пообіцяв не балотуватися на на...

Рисунок 3.2 — Скріншот читання даних 2000 записів з наглядним показом перших 10 даних

Тепер треба визначити, за якими темами (topic) поділяються новини та скільки цих тем:

```
print(data['topic'].unique(), len(data['topic'].unique()))
```

Усього таких тем виявилося 7:

– 'політика';
– 'АТО';
– 'економіка';
– 'гламур';
– 'туризм';
– 'наука та ІТ';
– 'книжки'

Починається етап вибірки даних. Обираються 5 категорій по 400 новин з кожної. Усе це записується до датафрейму під назвою df_res (рис. 3.3):.

```
topics = ['Політика' 'АТО' 'Економіка' 'Туризм' 'Наука та IT']
news_in_cat_count = 400
df_res = pd.DataFrame()
for topic in tqdm(topics):
    df_topic = data[data['topic'] == topic][:news_in_cat_count]
    df_res = df_res.append(df_topic, ignore_index=True)
```

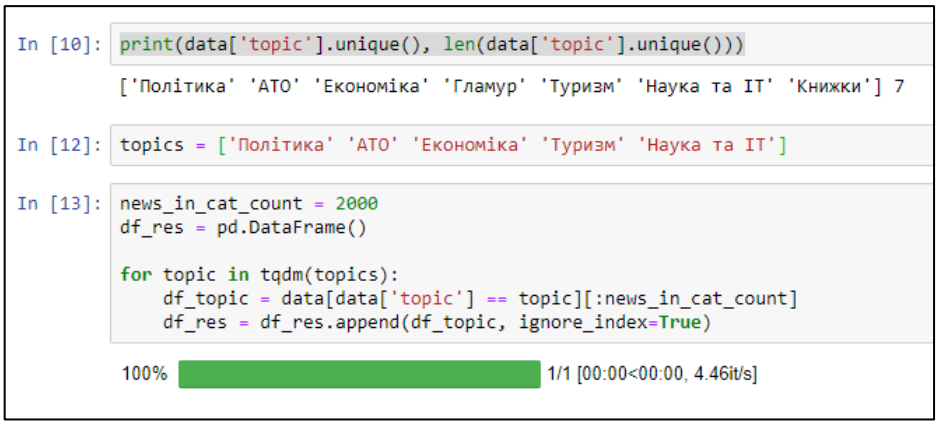


Рисунок 3.3 — Скріншот процесу вибірки даних за 5 темами з записом в датафреймі

```
df_res.shape (0, 5)
```

Тепер починається передобробка або токенізація тексту:

```
import string
```

Функція remove_punctuation здійснює заміну знаків пунктуації на пусті пробіли:

```
def remove_punctuation(text):
    return "".join([ch if ch not in string.punctuation else ' ' for ch in text])
```

Функція remove_numbers здійснює заміну чисел на пусті пробіли:

```
def remove_numbers(text):
    return ''.join([i if not i.isdigit() else ' ' for i in text])
import re
```

Функція remove_multiple_spaces здійснює заміну численних пробілів на пусті пробіли:

```
def remove_multiple_spaces(text):
```

```
        return re.sub(r'\s+', ' ', text, flags=re.I)
```

Для виділення з текстів так званих стоп-слів (паразитні слова) використовується бібліотека Mystem та будується функція lemmatize_text для здійснення лематизації текстів.

```
from nltk.stem import *
from nltk.corpus import stopwords
from pymystem3 import Mystem
from string import punctuation
mystem = Mystem()
ukranian_stopwords = stopwords.words("ukranian ")
ukranian _stopwords.extend(['…', '«', '»', '...'])
def lemmatize_text(text):
    tokens = mystem.lemmatize(text.lower())
    tokens = [token for token in tokens if token not in
ukranian_stopwords and token != " "]
    text = " ".join(tokens)
    return text
```

Наступним кроком здійснюється передобробка тексту (рис. 3.4):.

'Названі самі дратівливі фотографії які виїхали у відпустку люди викладають в соціальні мережі повідомляє the daily mail відповідний антирейтинг був складений на основі опитування відвідувачів онлайн сервісу для туристів top cot інтернет користувачі назвали найбільш дратівливим видом знімків з відпочинку скріншоти прогнозу погоди в місці перебування - відсоток опитаних не бажають бачити такі фото у своїй стрічці новин в соцмережі особливе невдоволення викликають Селфі вони дратують відсотка респондентів замикає трійку лідерів негативу такий вид фото як hot dog legs багато хто любить фотографувати свої засмаглі кінцівки на тлі моря басейну і т д «ноги сосиски» в соцмережах псують настрій відсоткам опитаних самі ненависні для користувачів типи знімків друзів розташувалися в рейтингу top cot так скріншоти з iphone про прогноз погоди на курорті відпускні Селфі знімки ніг на тлі моря басейну і т д так звані hot dog legs фотографії в стрибку оптичні ілюзії наприклад ейфелева вежа н а долоні фото написів на піску фото з тантамарески знімки заходів фото неба і пальм фото напоїв і їжі представник top cot алекс баттл alex buttle зазначив що укладачі рейтингу не збиралися ставити «банальні знімки» в докір інтернет користувачам за словами баттла в компанії усвідомлюють що причиною по якої багато користувачів проголосували за ті чи інші види фото може бути не роздратування а заздрість'

Рисунок 3.4 — Токенізований текст отриманий у результаті передобробки

```
preproccessing = lambda text:
(remove_multiple_spaces(remove_numbers(remove_punctuation(text))))
data['preproccessed'] = list(map(preproccessing, df_res['text']))
prep_text =
[remove_multiple_spaces(remove_numbers(remove_punctuation(text.lower())))
for text in tqdm(df_res['text'])]
len(prep_text)
prep_text[0]
```

Оброблений текст зберігається під назвою prep_text:

```
df_res['text_prep'] = prep_text
df_res.head(1)
```

Далі здійснюється стемінг, що полягає в видаленні так званих стоп-слів. Цей процес здійснюється з залученням бібліотеки SnowballStemmer.

```
from nltk.stem.snowball import SnowballStemmer
stemmer = SnowballStemmer("ukranian ")
ukranian_stopwords = stopwords.words("ukranian ")
ukranian _stopwords.extend(['…', '«', '»', '...', 'т.д.', 'т', 'д'])
text = df_res['text_prep'][0]
word_tokenize(text)
from nltk import word_tokenize
stemmed_texts_list = []
for text in tqdm(df_res['text_prep']):
    tokens = word_tokenize(text)
    stemmed_tokens = [stemmer.stem(token) for token in tokens if
token not in russian_stopwords]
    text = " ".join(stemmed_tokens)
    stemmed_texts_list.append(text)
df_res['text_stem'] = stemmed_texts_list
import nltk
nltk.download('punkt')
def remove_stop_words(text):
    tokens = word_tokenize(text)
    tokens = [token for token in tokens if token not in
russian_stopwords and token != ' ']
    return " ".join(tokens)
from nltk import word_tokenize
sw_texts_list = []
for text in tqdm(df_res['text_prep']):
```

```
            tokens = word_tokenize(text)
            tokens = [token for token in tokens if token not in
russian_stopwords and token != ' ']
            text = " ".join(tokens)
            sw_texts_list.append(text)
        df_res['text_sw'] = sw_texts_list
        df_res['text_sw'][0]
```

Список вилучених стоп-слів можна знайти в окремо створеному списку sw_texts_list

```
        df_res.to_csv('lenta_stemmed.csv')
        df_res['text_stem'][0]
```

Результат стемингу виводиться під списком даних df_res['text_sw'] (рис. 3.4):

'Названі самі дратівливі фотографії які виїхали відпустку люди викладають соціальні мережі повідомляє the daily mail відповідний антирейтинг складений основі опитування відвідувачів онлайн сервісу туристів top com інтернет користувачі назвали найбільш дратівливим видом знімків відпочинку скріншоти прогнозу погоди місці перебування - відсоток опитаних бажають бачити такі фото своєї новинній стрічці соцмережі особливе невдоволення викликають Селфі дратують відсотка респондентів замикає трійку лідерів негативу вид фото hot dog legs багато хто любить фотографувати свої засмаглі кінцівки тлі моря басейну ноги сосиски соцмережах псують настрій відсоткам опитаних самі ненависні користувачів типи знімків друзів розташувалися рейтингу top com скріншоти iphone прогнозі погоди курорті відпускні Селфі знімки ніг тлі моря басейну звані hot dog legs фотографії стрибку оптичні ілюзії наприклад ейфелева вежа долоні фото написів піску фото тантамарески знімки заходів фото неба пальм фото напи тково їжі представник top com алекс баттл alex buttle зазначив укладачі рейтингу збиралися ставити банальні знімки докір інтернет користувачам словами баттла компанії усвідомлюють причиною якої багато користувачів проголосували ті інші види фото роздратування заздрість '

Рисунок 3.4 —Текст отриманий у результаті стемінгу

Подальша лемматизація здійснюється з використанням бібліотеці mystem.lemmatize та з створенням функції lemmatize_text:

```
        lemm_texts_list = []
        for text in tqdm(df_res['text_sw']):
            #print(text)
```

```
        try:
            text_lem = mystem.lemmatize(text)
            tokens = [token for token in text_lem if token != ' ' and
token not in russian_stopwords]
            text = " ".join(tokens)
            lemm_texts_list.append(text)
        except Exception as e:
            print(e)
    df_res['text_lemm'] = lemm_texts_list
    def lemmatize_text(text):
        text_lem = mystem.lemmatize(text)
        tokens = [token for token in text_lem if token != ' ']
        return " ".join(tokens)
    df_res.to_csv('lemm.csv')
    df_res = pd.read_csv('lemm.csv', encoding = 'utf-8')
    df_res.head()
    df_res['text_lemm'][0]
```

Результат лемматизації (рис. 3.5) виводиться під набором даних df_res['text_lemm']:

'Називати самий дратівливий фотографія який їхати відпустку людина викладати соціальний мережу повідомляти the daily mail відповідний антирейтинг складати основа опитування відвідувач онлайн сервіс турист top сот інтернет користувач називати самий дратівливий вид знімок відпочинок скріншот прогноз погода місце перебування - відсоток опитувати бажати бачити фото свій новинний стрічка соцмережа особливий невдоволення викликати Селфі дратувати відсоток респондент замикати трійка лідер негатив вид фото hot dog legs багато любити фотографувати свій засмаглий кінцівку фон море басейн нога сосиска соцмережа псувати настрій відсоток опитувати самий ненависний користувач тип знімок один розташовуватися рейтинг top сот скріншот iphone прогноз погода курорт відпускної Селфі знімок нога фон море басейн називати hot dog legs фотографія стрибок оптичний ілюзія наприклад Ейфель вежа долоню фото напис пісок фото тантамаресок знімок захід фото небо пальма фото напій їжа представник top сот алекс баттл alex buttle відзначати укладач рейтинг збиратися ставити банальний знімок докір інтернет користувач слово баттла компанія усвідомлювати причина який многий користувач проголосувати інший вид фото роздратування заздрість '

Рисунок 3.5 —Текст отриманий у результаті лемматизації

Розбиття набору даних на залежні та незалежні змінні. Тепер потрібно розділити набір даних на значення X та Y. Y - це стовпець «target», тоді як X - решта незалежних змінних у наборі даних. Отже, X = df_res['text_sw'] – вхідні дані, а y = df_res['topic'] шуканий результат (target):

```
X = df_res['text_sw']
y = df_res['topic']
type(df_res['text_lemm'][0])
```

Для здійснення навчання тематичного класифікатору досліджуваний датасет розбивається на навчальну виборку X_train (70 % даних) та тренувальну виборку X_test (30 % даних):

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state = 42)
my_topics = df_res['topic'].unique()
my_tags
array(['Політика' 'АТО' 'Економіка' 'Туризм' 'Наука та ІТ'],
      dtype=object)
type(X_train[1])
df['Review'].values.astype('U')
```

Далі відбувається навчання тематичного класифікатору на основі використання наївного баєсовського алгоритму (Naive Bayes Classifier).

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer
nb = Pipeline([('vect', CountVectorizer()),
               ('tfidf', TfidfTransformer()),
               ('clf', MultinomialNB()),
              ])
nb.fit(X_train, y_train)
Pipeline(steps=[('vect', CountVectorizer()), ('tfidf',
TfidfTransformer()),
                ('clf', MultinomialNB())])
```

Точність та повнота регулюються при тренуванні класифікатора певним пороговим параметром (nb.predict).

```
from sklearn.metrics import classification_report
```

```
y_pred = nb.predict(X_test)
y_pred[0]
'Гламур'
print(X_test[0], y_test[0], y_pred[0])
```

Отримуємо тренувальний текст за категорією «гламур» (рис. 3.6):

названі самі дратівливі фотографії які виїхали відпустку люди викладають соціальні мережі повідомляє the daily mail відповідний антирейтинг складений основі опитування відвідувачів онлайн сервісу туристів top сот інтернет користувачі назвали найбільш дратівливим видом знімків відпочинку скріншоти прогнозу погоди місці перебування — відсоток опитаних бажають бачити такі фото своєї новинній стрічці соцмережі особливе невдоволення викликають Селфі дратують відсотка респондентів замикає трійку лідерів негативу вид фото hot dog legs багато хто любить фотографувати свої засмаглі кінцівки тлі моря басейну ноги сосиски соцмережах псують настрій відсоткам опитаних самі ненависні користувачів типи знімків друзів розташувалися рейтингу top сот скріншоти iphone прогнозі погоди курорті відпускні Селфі знімки ніг тлі моря басейну звані hot dog legs фотографії стрибку оптичні ілюзії наприклад ейфелева вежа долоні фото написів піску фото тантамарески знімки заходів фото неба пальм фото напит ков їжі представник top сот алекс баттл alex buttle зазначив укладачі рейтингу збиралися ставити банальні знімки докір інтернет користувачам словами баттла компанії усвідомлюють причиною якої багато користувачів проголосували ті інші види фото роздратування заздрість Подорожі

Рисунок 3.6 — Текст отриманий у результаті роботи баєсовського алгоритму

Під кінець, визначаються показники метрики навченої моделі тематичного класифікатору на основі використання наївного баєсовського алгоритму:

```
from sklearn.metrics import accuracy_score
print('accuracy %s' % accuracy_score(y_pred, y_test))
print(classification_report(y_test, y_pred, target_names=my_tags))
```

Точність системи (Precision) - це відношення кількості чітко визначених позитивних об'єктів до загальної кількості позитивних об'єктів, які класифікатор відніс до цього класу, тобто точність визначення позитивних відповідей.

Чим ближче до одиниці precision, тим менше неправильних визначень класів, які пораховані правильно.

$$Precision = \frac{TP}{TP+FP} \quad (3.1)$$

Повнота системи (Recall) - це відношення знайдених класифікатором об'єктів, що належать класу, до загальної кількості позитивних об'єктів в тестовій вибірці.

$$Recall = \frac{TP}{TP+FN} \quad (3.2)$$

Важливість точності і повноти залежить від конкретного завдання. Відповідно є функція, яка робить переваги повноті або точності, так звана F-міра:

$$F = (b^2 + 1)\frac{Precision \cdot Recall}{b^2 \cdot Precision + Recall} \quad (3.3)$$

У тих випадках, коли невідомо, яка з характеристик найбільш важлива, беремо *b=1(F₁-міра)*. В цьому випадку F_1 прагне до *0*, якщо точність або повнота прагнуть до *0*.

Точність та повнота регулюються при тренуванні класифікатора певним параметром (predict_prob), що має назву порогу. За допомогою зміни порога можна збільшити Precision до заданого нами значення, внаслідок цього Recall сильно впаде. Отримуємо оцінки точності моделі 0,922 (рис. 3.7):

accuracy 0.922

	precision	recall	f1-score	support
Політика	0.92	0.81	0.86	610
АТО	0.90	0.91	0.91	584
Економіка	0.92	0.98	0.95	626
Туризм	0.96	0.96	0.96	591
Наука та IT	0.91	0.96	0.93	589
accuracy			0.92	3000
macro avg	0.92	0.92	0.92	3000
weighted avg	0.92	0.92	0.92	3000

Рисунок 3.7 — Оцінка точності моделі

На фінішному етапі здійснюється тестова перевірка на прикладах новин з сайту 1+1 завантаженим окремо документом під назвою econ_text, що наводить опис окремої новини (рис. 3.8).

Долар США вважається символом надійності, безпеки та економічного процвітання. Він займає незаперечне домінуюче становище в міжнародній фінансовій системі з середини XX століття і справляє враження непереможного титану. Проте ера панування долара як основної світової резервної валюти повільно підходить до кінця. Найбільші банки пророкують йому різкий спад вже в наступному році, а відомий економіст. Стівен Роуч впевнений, що американська валюта може знецінитися на третину.

Причинами обвалу стануть скорочення заощаджень населення, зростання державного боргу США і посилення Китаю.

Рисунок 3.8 — Прикладах новин з сайту

Текст проходить стадії токенізації, стемингу та лематизації (рис. 3.9).

```
econ_text =
remove_multiple_spaces(remove_numbers(remove_punctuation(econ_text.lower())
))
econ_text = remove_stop_words(econ_text)
econ_text = lemmatize_text(econ_text)
```

'Долар сша вважатися символ надійність безпеку економічний процвітання займати незаперечний домінувати положення міжнародний фінансовий система середина xx століття справляти враження непереможний титан проте ера панування долар основною світовою резервний валюта повільно підходити кінець великий банк передрікати різкий спад наступний рік відомий економіст Стівен Роуч впевнений американський валюта знецінюватися третину причина обвал ставати скорочення заощадження населення зростання державний борг сша посилення китай захід доларовий диктат - матеріал стрічка ру непомірний привілей успіх американський економіка xx століття багато обумовлювати домінуючий роль долар чергу досягнення роль ставати результат політичний військовий перевага який сша купувати перший світовий війна цей час позиція долар світ фінанси представляти головний основа процвітання сша '

Рисунок 3.9 — Отриманий текст після виконання роботі алгоритма

```
ect_pred = logreg.predict([econ_text])
ect_pred
```

Внаслідок здійсненого тестового аналізу навченим тематичним класифікатором даний матеріал відноситься до теми «Політика».

```
array([' Політика '], dtype=object).
```

ВИСНОВОК

Під час обробки великих колекцій документів є актуальним завдання класифікації текстів. Класифікація означає віднесення кожного документа до певного класу із заздалегідь відомими параметрами. Для вирішення цих завдань застосовуються методи машинного навчання.

В роботі детально розглянуті основи комп'ютерної лінгвістики, наведені приклади текстомайнінгових програмних продуктів та існуючі сучасні підходи до подальшої розробки нових програмних продуктів у цьому напряму.

В якості основного алгоритму розглянутий такий алгоритм класифікації та обробки текстових даних, як наївний баєсівський класифікатор. Серед методів обробки природної мови розглянуті для подальшого використання токенізація, стемінг та лемматизація.

Наведена формальна постановка задачі класифікації текстових даних, створений комбінований алгоритм, що включає до себе визначення тематичної класифікації текстів з попереднім навчанням на основі заздалегідь створеного спеціалізованого набору текстів, що описують новини.

Проаналізована ефективність розробленого класифікатора на підготовленому датасету даних, створених на основі матеріалів сайту новин телеканалу 1+1. Отримана точність класифікації теми новин за 6 тематичними категоріями складає 92%. Здійснена практична перевірка навченого тематичного класифікатора на основі тексту новин, що є присвячений політиці. Результат перевірки позитивний.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Відношення в комп'ютерній лінгвістиці. URL: http://irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe/Nznuoaf_2013_36_15.pdf (дата звернення: 18.04.2021).

2. Chomsky, N. Syntactic Structures. The Hague: Mouton, 1957. - 342 P.

3. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика : учеб. пособие / Большакова Е.И., Клышинский Э.С., Ландэ Д.В., Носков А.А., Пескова О.В., Ягунова Е.В. - М.: МИЭМ, 2011. - 272 с.

4. Dempster A. P., Laird N. M., Rubin D. B. Maximum likelihood from incomplete data via the EM algorithm // J. of the Royal Statistical Society, Series B. - no. 34. - Pp. 1–38.

5. Eisenstein J., Ahmed A., Xing E. P. Sparse additive generative models of text // ICML’11. - 2011. - Pp. 1041–1048.

6. Павлов А. С., Добров Б. В. Метод обнаружения массово порожденных неестественных текстов на основе анализа тематической структуры // Вычислительные методы и программирование: новые вычислительные технологии. - 2011. - Т. 12. - С. 58–72.

7. Apishev M., Koltcov S., Koltsova O., Nikolenko S., Vorontsov K. Mining ethnic content online with additively regularized topic models // Computacion y Sistemas. - 2016. - Vol. 20, no. 3. - P. 387–403.

8. Автоматическая обработка текстов на естественном языке и анализ данных : учеб. пособие / Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. - М.: Изд-во НИУ ВШЭ, 2017. - 269 с.

9. Барсегян А.А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP - СПб.: БХВ-Петербург, 2008.

10. Blei D. M., Jordan M. I. Modeling annotated data // Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval. - New York, NY, USA: ACM, 2003. - Pp. 127–134.

11. Bodrunova S., Koltsov S., Koltsova O., Nikolenko S. I., Shimorina A. Interval semisupervised LDA: Classifying needles in a haystack // MICAI (1) / Ed. by F. C. Espinoza, A. F. Gelbukh, M. Gonzalez-Mendoza. - Vol. 8265 of Lecture Notes in Computer Science. - Springer, 2013. - Pp. 265–274.

12. Виноград Т. Программа, понимающая естественный язык - М.: Мир, 2011. – 245 с.

13. Извлечение информации из неструктурированных текстов. URL: https://compress.ru/article.aspx?id=19605 (дата звернення: 18.04.2021).

14. International youth conference «Informational systems and technologies» URL: http://www.analyst.ru (дата звернення: 18.04.2021).

15. SAPPHIRE NOW: Custom-made for you. URL: http://www.businessobjects.com/product/catalog/text_analysis/features.asp (дата звернення: 18.04.2021).

16. Lockheed Martin Signs CMD Solutions As Authorized Distributor Of AeroText(TM) Information Extraction Software URL: https://news.lockheedmartin.com/2005-02-24-Lockheed-Martin-Signs-CMD-Solutions-as-Authorized-Distributor-of-AeroText-TM-Information-Extraction-Software (дата звернення: 18.04.2021).

17. Attensity Semantic Annotation Natural Language Processing brings Real time Discovery URL: https://www.predictiveanalyticstoday.com/attensity-semantic-annotation-natural-language-processing-brings-real-time-discovery/ (дата звернення: 18.04.2021).

18. Васильев В. Г., Кривенко М. П. Методы автоматизированной обработки текстов. - М.: ИПИ РАН, 2008.

19. Умаров Т.С., Баженова И. Ю. Современные подходы к механизмам извлечения причинно-следственных связей из неструктурированных текстов на естественном языке// International Journal of Open Information Technologies - vol. 7, no.7, - 2019 – P. 81-89.

20. Медиалогия — разработчик автоматической системы мониторинга и анализа СМИ и соцмедиа в режиме реального времени. URL: http://www.mlg.ru/about (дата звернення: 18.04.2021).

21 FUZZY PROBABILISTIC NEURAL NETWORK IN DOCUMENT CLASSIFICATION TASKS. / A. Yerokhin, O. Zolotukhin. // Interbranch collection of scientific papers «Information Extraction and Processing». National Academy of Sciences of Ukraine. - 2019. http://vidbir.ipm.lviv.ua/

22 Overview and Analysis of Existing Decisions of Determining the Meaning of Text Documents. / Gruzdo I., Tereshchenko G. // International Scientific-Practical Conference Problems of Infocommunications. Science and Technology PIC S&T`2018 978-1-5386-6611-1/18/$31.00 ©2018 IEEE October 9-12, 2018 Kharkiv, Ukraine p.645-653

23. Bassiou N., Kotropoulos C. Online PLSA: Batch updating techniques including outof-vocabulary words // Neural Networks and Learning Systems, IEEE Transactions on. - Nov 2014. - Vol. 25, no. 11. - Pp. 1953–1966.

24 Effectiveness of Preprocessing Algorithms for Natural Language Processing Applications / Smelyakov K., Chupryna A., Karachevtsev D., Kulemza D., Samoilenko Y., Patlan O. // 2020 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), 6-9 Oct. 2020, Kharkiv, Ukraine. – P. 1-5.

25 Analysis of the problem of homonyms in the hyperchains construction for lexical units of natural language “Computer science and information technologies” / Puzik O., Tyshchenko O., Ghetverikov G. // Proceedings of the XIII-th International Scientific and Technical Conference CSIT 2018, 11-14 Sept. 2018, Lviv, Ukraine, pp. 356-359.

26 Applıcatıon of paragraphs vectors model for semantıc text analysıs / Kyrychenko, I., Tereshchenko, G., Gruzdo, I., Cherednichenko, O // 4th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2020), Lviv, Ukraine, April 23-24, 2020. – Volume I, PP. 283-293. SCOPUS