Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)
(рівень вищої освіти)

Дослідження методів тематичного моделювання корпусів текстів на прикладі контенту сайту.
(тема)

Виконав:
студент 2 курсу, групи СШІм-18-3
Водяницький Д.В.
(прізвище, ініціали)
Спеціальність 122 – Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо -наукова)

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва освітньої програми)
Керівник доц. Чала Л.Е.
(посада, прізвище, ініціали)

Допускається до захисту

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

2020 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти другий (магістерський)

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

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

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

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

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Водяницькому Дмитру Валерійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів тематичного моделювання корпусів текстів на прикладі контенту сайту»
затверджена наказом по університету від 30.03.2020 р. № 480Ст
2. Термін подання студентом роботи до екзаменаційної комісії 19 травня 2020 р.
3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та відомих наукових проектів щодо розробки та дослідження тематичного моделювання корпусів текстів, Python documentation,
4. Перелік питань, що потрібно опрацювати в роботі Аналіз науково-технічної літератури з питань обробки природномовних текстів, вивчення метрик міжнародних рейтингів, аналіз контенту сайту університету, проведення експериментального моделювання та навчання моделі, розробка тематичної моделі сайту університету

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _____
_____
_____
_____
_____
_____
_____
_____
_____
_____

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
Аналіз предметної галузі	доц. Чала Л.Е.		12.04.2020
Методи і алгоритми обробки текстів	доц. Чала Л.Е.		20.04.2020
Експериментальне моделювання	доц. Чала Л.Е.		30.04.2020
Розробка тематичної моделі сайту	доц. Чала Л.Е.		08.05.2020

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі	11.04.2020	виконано
2	Етапи контент-аналізу	13.04.2020	виконано
3	Аналіз метрик міжнародних рейтингів	14.04.2020	виконано
4	Аналіз методів обробки текстових документів	20.04.2020	виконано
5	Експериментальне моделювання та навчання моделі	30.04.2020	виконано
6	Розробка тематичної моделі сайту університету	08.05.2020	виконано
7	Написання пояснювальної записки	14.05.2020	виконано
8	Попередній захист	15.05.2020	виконано
9	Захист перед ЕК	19.05.2020	

Дата видачі завдання ____30 березня_2020 р.

Студент _____
(підпис)

Керівник роботи _____ _____доц. Чала Л.Е._____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Записка пояснювальна: 80 с., 41 рис., 1 табл., 1 дод., 44 джерела.

ТЕМАТИЧНЕ МОДЕЛЮВАННЯ, ОБРОБКА ПРИРОДНОМОВНИХ ТЕКСТІВ, КОНТЕНТ-АНАЛІЗ, ВЕБСАЙТ, ЛАТЕНТНО-СЕМАНТИЧНИЙ АНАЛІЗ, ЛАТЕНТНЕ РОЗМІЩЕННЯ ДІРІХЛЕ, РЕДУКЦІЯ

Об'єкт дослідження – процеси автоматизованого статистичного аналізу електронних текстових документів.

Предмет дослідження – методи тематичного моделювання корпусів природномовних текстів.

Мета роботи – дослідження методів тематичного моделювання корпусів текстів та створення тематичної моделі на основі контенту вебсайту університету.

Методи дослідження – методи статистичного аналізу природномовних текстів, латентно-семантичний аналіз, латентне розміщення Діріхле, методи редукції.

РЕФЕРАТ

Пояснительная записка: 80 с., 41 рис., 1 табл., 1 прил., 44 источника.

ТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ, ОБРАБОТКА ЕСТЕСТВЕННОЯЗЫКОВЫХ ТЕКСТОВ, КОНТЕНТ-АНАЛИЗ, ВЕБСАЙТ, ЛАТЕНТНО-СЕМАНТИЧЕСКИЙ АНАЛИЗ, ЛАТЕНТНОЕ РАЗМЕЩЕНИЕ ДИРИХЛЕ, РЕДУКЦИЯ

Объект исследования – процессы автоматизированного статистического анализа электронных текстовых документов.

Предмет исследования – методы тематического моделирования корпусов естественно-текстов.

Цель работы – исследование методов тематического моделирования корпусов текстов и создания тематической модели на основе контента вебсайта университета.

Методы исследования – методы статистического анализа естественно-текстов, латентно-семантический анализ, латентное размещение Дирихле, методы редукции.

ABSTRACT

Explanatory note: 80 p., 41 fig., 1 tabl., 1 ann., 44 sources.

TOPIC MODELING, NATURAL LANGUAGE PROCESSING, CONTENT ANALYSIS, WEBSITE, LATENT SEMANTIC ANALYSIS, LATENT DIRICHLE ALLOCATION

The object of research – the processes of automated statistical analysis of electronic text documents.

The subject of research – methods of topic modeling of natural language texts corpora.

The purpose of the work is to study the topic modeling methods of text corpora and to create a topic model based on the university website content.

Research methods – methods of statistical analysis of natural language texts, latent semantic analysis, latent Dirichlet allocation, reduction methods.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

КЛ – комп’ютерна лінгвістика;

ЛСА – латентний семантичний аналіз;

ЛДА – латентне розміщення Діріхле;

ПМ – природня мова;

ЦСР – цілі сталого розвитку;

ARTM – адитивна регуляризація тематичних моделей;

CL – Computational Linguistics;

LDA – Latent Dirichlet Allocation;

LSA – Latent Semantic Analysis;

NLP – Natural Language Processing;

THE – Times Higher Education.

ВСТУП

Поява мережі Інтернет та бурхливе зростання доступної текстової інформації значно прискорило розвиток наукової галузі, яка існує вже багато десятків років і відомої як автоматична обробка текстів (Natural Language Processing, NLP) і комп’ютерна лінгвістика (Computational Linguistics, CL). В рамках цієї області запропоновано багато перспективних ідей з автоматичної обробки текстів природньою мовою (ПМ), які були втілені в багатьох прикладних системах, в тому числі комерційних. Сфера додатків комп'ютерної лінгвістики постійно розширюється, з’являються все нові завдання, які успішно вирішуються, в тому числі із залученням результатів суміжних наукових областей.

Комп'ютерна лінгвістика (КЛ) – міждисциплінарна область, яка виникла на стику таких наук, як лінгвістика, математика, інформатика (computer science), штучний інтелект (artificial intelligence), в своєму розвитку вона до сих пір вбирає і застосовує (при необхідності адаптуючи) розроблені в цих науках методи і інструменти.

Витоки КЛ сходять до досліджень відомого американського лінгвіста Н. Хомського по формалізації структури природної мови [1], до перших експериментів з машинного перекладу, виконаним програмістами і математиками, а також до розробленим в області штучного інтелекту першим програмами розуміння природної мови (наприклад, [2]).

Напрямів комп’ютерної лінгвістики досить багато, зокрема машинний переклад, інформаційний пошук, реферування та анотування текстів, класифікація та кластеризація текстів, в тому числі тематичне моделювання, створення чат-ботів, аналіз тональності текстів, видобування знань з текстів, автоматична генерація текстів тощо.

Тематичне моделювання – один з сучасних напрямів прикладної лінгвістики, а саме статистичного аналіза текстів. Тематичне моделювання – один з методів кластеризації текстових документів, різниця полягає в тому,

що під час кластеризації документ має бути віднесеним тільки до одного кластеру, в той же час тематична модель дозволяє здійснити «м'яку кластеризацію» (soft clustering), яка відносить документ до декількох кластерів-тем з деякими ймовірностями [3]. Ця властивість дозволяє частково вирішити проблеми синонімії та полісемії слів. Синоніми, які взаємозамінні у подібних контекстах, групуються в тих самих темах. Багатозначні слова та омоніми, навпаки, розподіляють свої ймовірності за декількими не пов'язаними темами. Наприклад, слово «ядро» може бути розпізнаним з того, яка тема домінує в контексті даного слова – математика, фізика, біологія чи військова історія.

Тематичні моделі застосовуються для виявлення трендів в новинних потоках, патентних базах, архівах наукових публікацій [4, 5], для багатомовного інформаційного пошуку [6, 7], пошуку тематичних співтовариств в соціальних мережах [8, 9, 10, 11], класифікації та категоризації документів [12, 13], тематичної сегментації текстів [14, 15], аналізу зображень та відеопотоків [16], тегування веб-сторінок [17], виявлення текстового спаму [18], в рекомендаційних системах [19, 20]. Багато інших різновидів та додатків тематичних моделей згадуються в оглядах [21, 22].

У даній атестаційній роботі як корпус текстів розглядається контент веб сайтів університету та окремих кафедр з метою виявлення певних тем.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Контент-аналіз сайту

Контент-аналіз сайту потрібен для оцінки ефективності розміщених статей, оглядів, описів та інших видів контенту. Аналіз дозволяє зрозуміти, як на контент реагуватимуть користувачі і пошукові системи, що можна поліпшити для підвищення позицій у видачі, від яких прийомів краще відмовитися. Контент-аналіз сайту зазвичай використовують після запуску веб-ресурсу, коли з’являється перший трафік, в процесі розвитку порталу або після зміни подачі матеріалів, тематики сайту, його оформлення.

Методи контент-аналізу.

Є багато методів аналізу інформації, розміщеної на сайті. Їх умовно можна поділити на дві великі групи:

– підхід з точки зору користувача (якісний). У цьому випадку оцінюється структура статей, їх візуальне оформлення та користь, яку вони приносять читачеві. Часто такий підхід до оцінки вимагає ручної перевірки та вичитки;

– підхід з точки зору машин (кількісний). До них відносяться пошукові системи і сервіси веб-аналітики. У цьому випадку оцінюється оптимізація тексту, мета-теги, число посилань на сторінку та інші технічні параметри. Їх можна перевірити за допомогою різних сервісів.

При контент-аналізі сайту можна оцінювати його цілком, але тоді доведеться витратити час на аналіз кожної сторінки, або оцінити вибірково 2, 5, 20 або більше сторінок. В цьому випадку показники будуть не ідеально точними, але це потребує менше часу.

Метрики і показники для аналізу контенту:

а) для оцінки з точки зору користувачів:

– читабельність. Її краще оцінювати вручну. Ознака гарної читабельності тексту – відсутність незрозумілих абревіатур і довгих

неузгоджених речень з великою кількістю дієприкметникових і прикметникових оборотів. Якщо контент перевантажений формулами, термінами, довгими словосполученнями, краще зробити його більш простим і зрозумілим;

– стиль, пунктуація, помилки. Для оцінки стилю і пунктуації, а також пошуку допущених помилок теж краще вичитувати текст вручну. Якщо в контенті є стилістичні, орфографічні та пунктуаційні помилки, користувач може піти з сайту. А надто велика кількість помилок може підірвати репутацію сайту – наприклад, вони взагалі неприпустимі в експертному контенті.

– оформлення. Чим воно краще – тим швидше користувач знайде потрібну інформацію на сторінці. Оформлення можна оцінити візуально буквально за кілька секунд. Вдала верстка – та, в якій є зрозуміла структура: заголовки і підзаголовки, підзаголовки в підбір, марковані та нумеровані списки, виділені елементи з важливою інформацією. Якщо на сайті «полотно», швидше за все, користувач втомиться шукати відповідь на свій запит і покине сторінку.

Окремо треба зробити увагу на схемах та структурі організації контенту сайту – якщо схема та структура зрозуміла та «прозора», користувач без перешкод зможе знайти потрібну інформацію [23]. Пі час пошуку інформації користувач стикається з такими проблемами, як неоднозначність інформації, її гетерогенність, наявність різних точок зору у користувачів, внутрішні політики тощо. Уникнути цих проблем можливо, створивши правильні та зрозумілі схему та структуру інформації на сайті.

На рис. 1.1 наведені можливі схеми організації інформації на сайті.

Рисунок 1.1 – Схеми організації інформації

Правильна побудова схеми та структури організації інформації дозволить підвищити якість інформаційного пошуку та позитивно впливатиме на релевантність результатів.

Також треба передбачити зони пошуку, які являють собою підмножини вмісту веб сайту, які індексуються окремо від решти вмісту сайту. Коли користувач виконує пошук в деякій зоні, це означає, що в результаті взаємодії з сайтом він вже позначив себе як зацікавленого в цій конкретній інформації. В ідеалі зони пошуку на сайті повинні відповідати його конкретним потребам, що підвищить ефективність видобування інформації. Шляхом виключення вмісту, що не має відношення до його потреб, користувач має

отримати менше результатів, але ці результати будуть більш релевантними.

Тематичне моделювання допоможе розділити вміст сайту на конкретні тематичні зони.

1.2 Використання тематичного моделювання для вирішення проблем інформаційного пошуку

1.2.1 Розвідувальний інформаційний пошук

Важливим впровадженням методів тематичного моделювання є інформаційний пошук.

Сучасні пошукові системи призначені, головним чином, для пошуку конкретних відповідей на короткі текстові запити. Інші пошукові потреби виникають у користувачів, яким необхідно розібратися в новій предметної області або поповнити свій багаж знань. Користувач може не володіти термінологією, слабо розуміти структуру предметної області, не мати точних формулювань запиту і не мати на меті єдину правильну відповідь. У таких випадках потрібен пошук не по ключовим словами, а за змістом. Запитом може бути довгий фрагмент тексту, документ або добірка документів. результатом пошуку має бути зручно систематизована інформація, дорожня карта предметної області.

Для цих випадків підходить парадигма розвідувального інформаційного пошуку (exploratory searh) [24, 25]. Його метою є отримання відповідей на складні питання: які теми представлені в тексті запиту, що читати, в першу чергу, за цими темами, що знаходиться на стику цих тем із суміжними областями, яка тематична структура даної предметної області, як вона розвивалася в часі, які останні досягнення, де знаходяться основні центри компетентності, хто є експертом з даної теми тощо. Користувач звичайної пошукової системи змушений ітеративно переформулювати свої короткі запити, розширюючи зону пошуку під час засвоєння термінології предметної

області, періодично переглядаючи і систематизуючи результати пошуку. Це вимагає витрат часу і високої кваліфікації. При відсутності інструменту для отримання загальної картини залишається сумнів, що якісь важливі аспекти досліджуваної проблеми так і не були знайдені. Можна образно представити ітеративний пошук як блукання по лабіринту знань, тоді розвідувальний пошук – це засіб автоматичної побудови карти для будь-якої частини цього лабіринту.

Розвідувальний пошук використовується у випадку, коли користувач не знає ключових термінів, або його може зацікавити суміжні галузі [24].

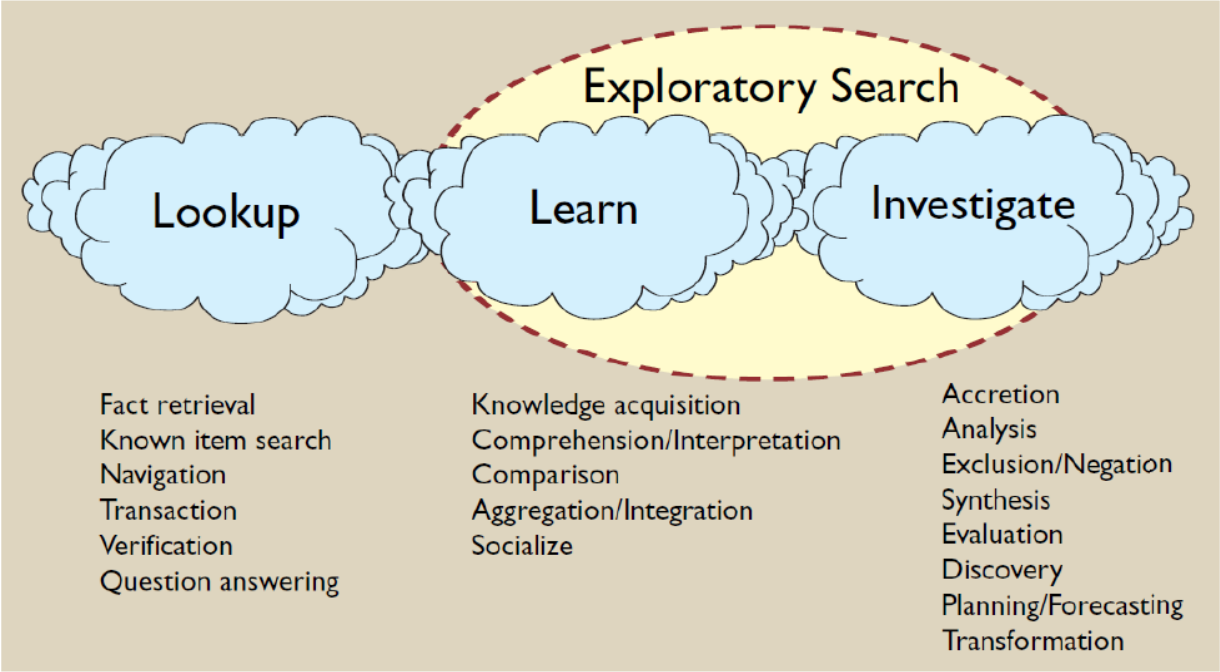

Рисунок 1.2 – Розвідувальний інформаційний пошук

1.2.2 Тематичний пошук

Повнотекстові пошукові системи засновані на інвертованих індексах, в яких для кожного слова зберігається список документів, які його містять [26]. Пошукова система шукає документи, що містять всі слова запиту, тому по довгому запиту, швидше за все, нічого не буде знайдено. Система тематичного розвідувального пошуку спочатку будує тематичну модель запиту і визначає короткий список тем запиту. Потім для пошуку документів схожої тематики застосовуються ті ж механізми індексування і пошуку, тільки в ролі слів виступають теми. Оскільки число тем на кілька порядків менше обсягу словника, тематичний пошук потребує набагато менше пам’яті порівняно з повнотекстовим пошуком і може бути реалізований за допомогою дуже скромної техніки. Технології інформаційного пошуку на основі тематичного моделювання в даний час активно розвиваються [27, 28].

1.3 Тематична організація інформації

В даній атестаційній роботі акцент зроблено на тематичну схему організації, яка дозволить структурувати контент сайту з метою найкращого задоволення потреб користувачів.

За основу взято контент веб сайту ХНУРЕ, який містить більше 4000 сторінок, з яких більше 2000 моніторяться різними міжнародними рейтингами. Для найкращого представництва університету у міжнародних рейтингах необхідно організувати інформацію на сайті таким чином, щоб пошукові системи рейтингових агентств легко змогли знайти всю інформацію, яка їх цікавить. Треба зауважити, що рейтингують університети за різними показниками та критеріями, тому метою адміністраторів є створення максимально зручної структури веб сайту.

Наприклад, глобальне дослідження і супроводжуючий його рейтинг кращих університетів світового значення за версією британського видання

Times Higher Education включає так званий рейтинг впливу університетів University Impact Rankings. ХНУРЕ рейтингується у THE University Impact Rankings – один з десяти українських ЗВО [29].

Рейтинг впливу від Times Higher Education вимірює успіхи університетів світу в досягненні цілей ООН у галузі сталого розвитку. В ньому використано відкалібровані індикатори для всебічного та збалансованого порівняння за такими напрямками:

- дослідження (research);

- аутріч (outreach) – робота по охопленню соціальними послугами цільової групи (як правило, соціально незахищеної);

- управління (stewardship) – відповідальне планування і управління ресурсами: навколишнім середовищем, природою, економікою, здоров'ям, власністю, інформацією тощо.

З 17 Цілей сталого розвитку (ЦСР) експертами з THE у першому виданні рейтингу оцінювалася успішність університетів по 11 з них:

а) 3 – Міцне здоров'я і благополуччя;

б) 4 – Якісна освіта;

в) 5 – Гендерна рівність;

г) 8 – Гідна праця та економічне зростання;

д) 9 – Промисловість, інновації та інфраструктура;

е) 10 – Скорочення нерівності;

ж) 12 – Відповідальне споживання і виробництво;

з) 13 – Пом'якшення наслідків зміни клімату;

и) 16 – Мир, справедливість та сильні інститути;

і) 17 – Партнерство в інтересах сталого розвитку.

Університети могли надавати дані по максимально можливій кількості ЦСР. У рейтинг включався будь-який університет, що надав дані про ЦСР 17 і, щонайменше, по трьох інших ЦСР. Підсумковий бал університету в загальному рейтингу розраховується шляхом додавання його балів за ЦСР 17 до трьох кращих цілей з решти десяти. На ЦСР 17 припадає 22% загального

бала, на інші цілі по 26%. Це означає, що різні університети оцінюються на основі різного набору ЦСР у залежності від їх спрямованості. Оцінка по кожній цілі масштабована таким чином, що найвища оцінка становить 100. Індикатори наукових досліджень отримані з даних, наданих Elsevier. Це документи зі Scopus, що відносяться до даної ЦСР. Як і в Світовому рейтингу використано п’ятирічне вікно даних: 2013 – 2017 рр.

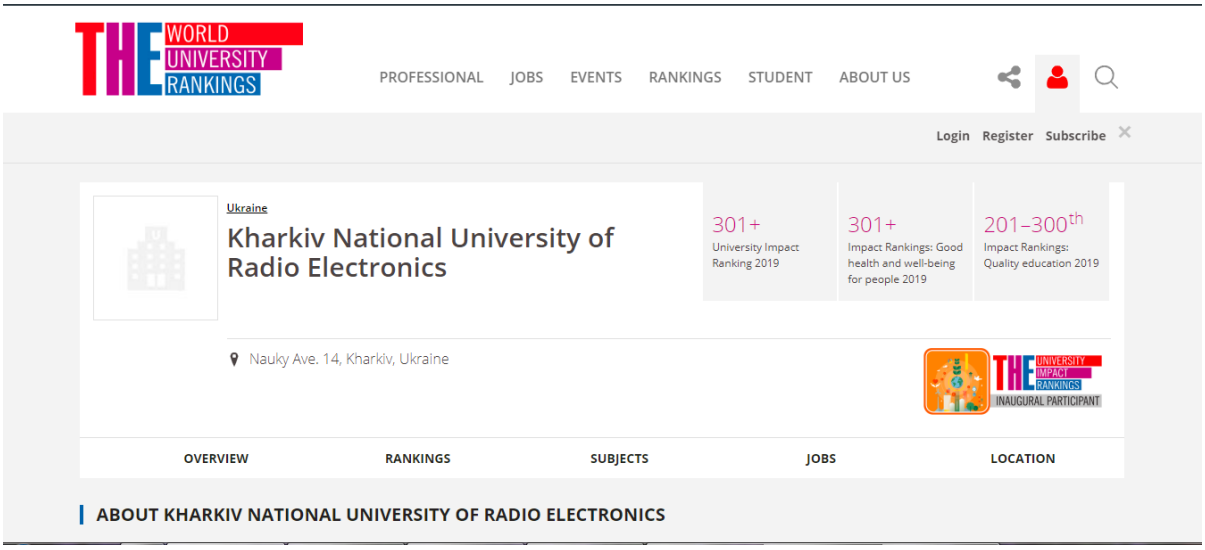

Рисунок 1.3 – Профіль ХНУРЕ у THE University Impact Rankings

Рисунок 1.4 – Сторінка сайту ХНУРЕ «Цілі сталого розвитку»

Для участі у цьому рейтингу університет повинен мати активності з виконання задач для досягнення цілей сталого розвитку. THE University Impact Rankings потребує доказів проведення активностей, які мають бути розміщені на сайті університету. Доцільно було створити окремий розділ на сайті, щоб всі досягнення університету у даному напрямку можна було легко знайти. Такий розділ було створено, та періодично він наповнюється записами про події та заходи відповідної направленості. Але на сайті університету та на окремих сайтах кафедр щоденно з'являється купа новин, які стосуються цілей сталого розвитку, і стає неможливо їх відстежити. Тому постала задача автоматичного аналізу контенту сайту університету та сайтів кафедр з метою розділити новини, записи и вміст сторінок за конкретними темами.

Ще один з найвпливовіших світових рейтингів університетів – QS, та один з його розділів Graduate Employability Rankings (Рейтинг працевлаштування випускників QS) – це спромога глобального порівняння результатів трудової діяльності випускників навчальних закладів [30]. Цей рейтинг, новітній в портфелі рейтингів QS, і, мабуть, самий інноваційний, представляє ряд проблем, найочевидніша з яких полягає в тому, що можливість працевлаштування є складною концепцією для оцінки в глобальному масштабі. Цей рейтинг призначений для того, щоб забезпечити студентів всього світу унікальним інструментом, за допомогою якого можна досягти успішності в університеті з точки зору результатів і перспектив працевлаштування випускників.

Цей рейтинг потребує доказів на сайті університету щодо партнерства з роботодавцями, заходів, спрямованих на сприяння працевлаштуванню студентів та випускників.

Кафедри регулярно проводять зустрічі зі своїми партнерами-роботодавцями з метою подальшого працевлаштування випускників. Новини про ці заходи розміщуються на сайтах кафедр, на офіційному сайті фіксуються лише загальноуніверситетські події.

Рисунок 1.5 – Сторінка QS World University Rankings на сайті ХНУРЕ

Тому, задля гідного представлення університету у рейтингу QS також важливо відстежувати записи на сайтах кафедр і у блоці новин університету з метою їх концентрації у одному розділі сайту, що і було зроблено. На рис. 1.6 наведено сторінку цього розділу.

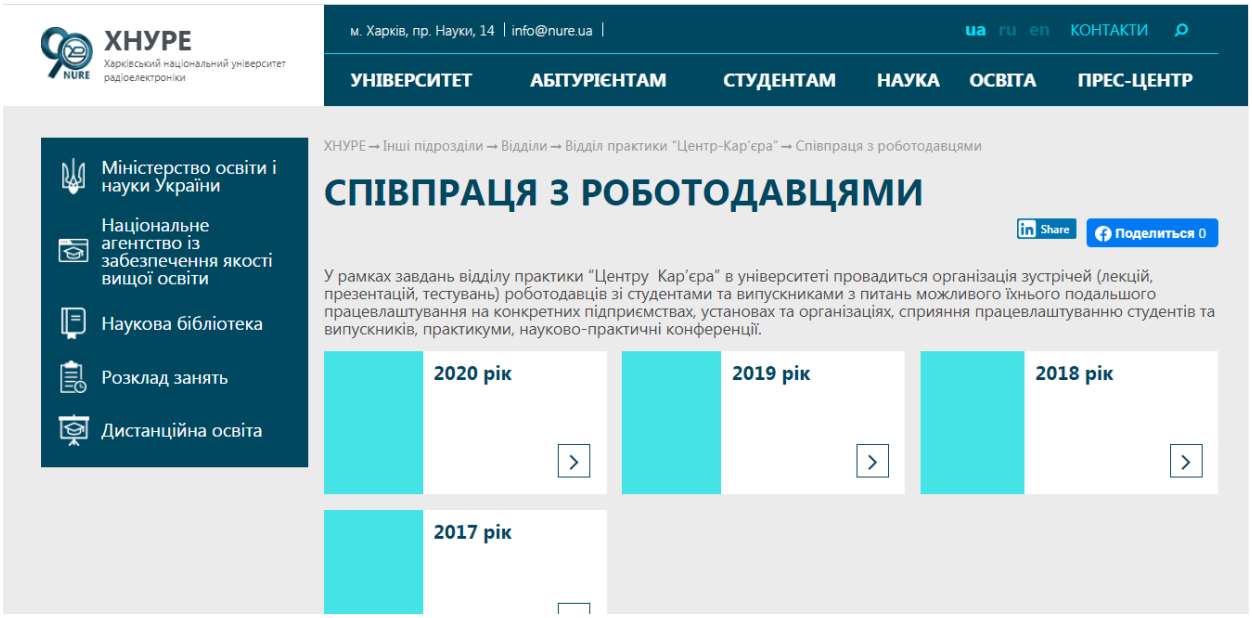

Рисунок 1.6 – Сторінка заходів з роботодавцями на сайті ХНУРЕ

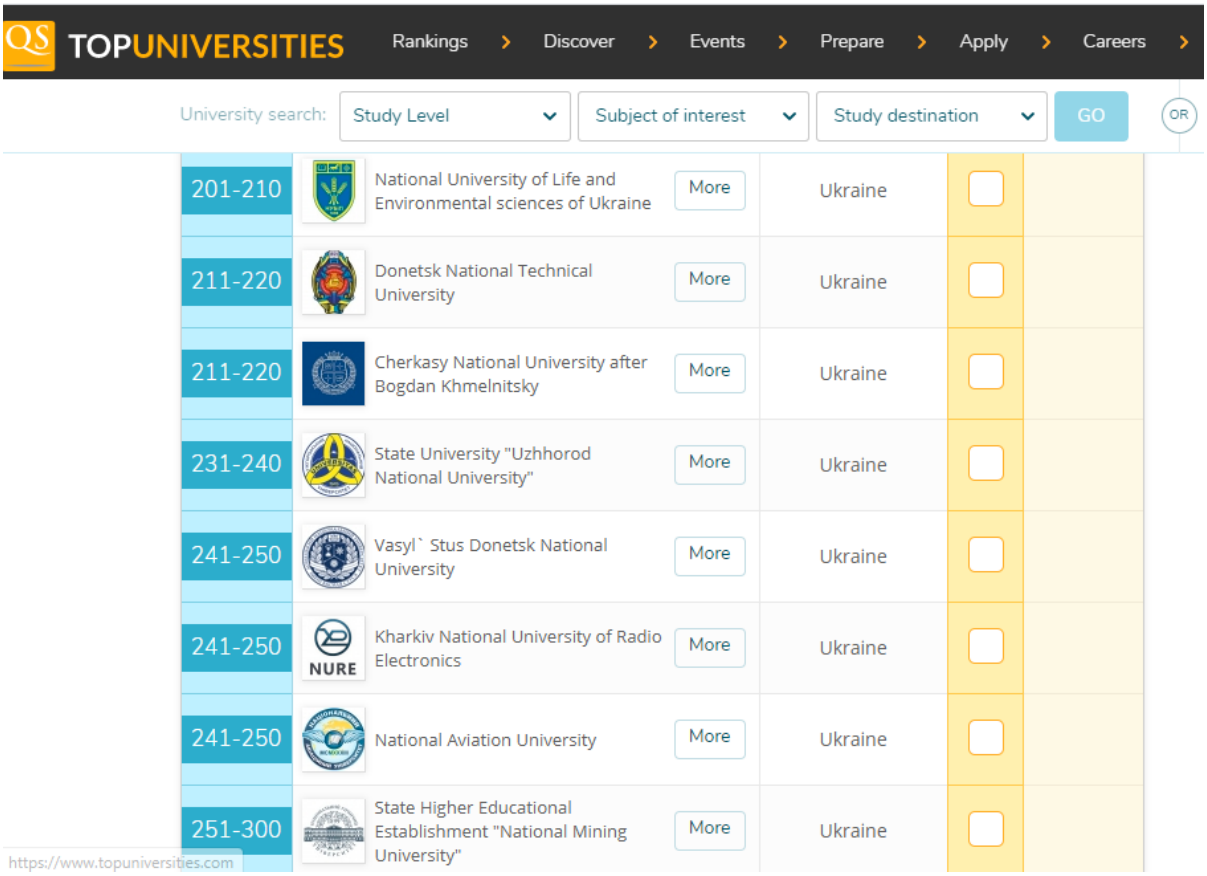

Рисунок 1.7 – Фрагмент рейтингового списку українських університетів у QS EECA University Rankings 2020

З огляду на вищезазначене актуальною задачею вважається створення тематичної моделі контенту сайтів університету і кафедр з метою виділити глобальні теми згідно з цілями сталого розвитку, а саме за цілями 3 – Міцне здоров'я та благополуччя, 4 – Якісна освіта, 5 – Гендерна рівність та 10 – Подолання нерівності.

1.4 Постановка задач дослідження

Метою атестаційної роботи є дослідження методів тематичного моделювання корпусів текстів та створення тематичної моделі на основі контенту сайтів університету та окремих кафедрі.

Згідно з метою роботи були поставлені наступні задачі:

- аналіз існуючих наукових джерел за темою дослідження;
- аналіз контенту сайтів університету та кафедр;
- аналіз вимог та критеріїв міжнародних рейтингів щодо наповнення сайтів університетів;
- дослідження методів та етапів тематичного моделювання;
- розробка тематичної моделі контенту сайтів університету та кафедр згідно з критеріями рейтингів;
- навчання та апробація тематичної моделі.

2 МЕТОДИ ТА АЛГОРИТМИ ОБРОБКИ ТЕКСТОВИХ ДОКУМЕНТІВ

2.1 Представлення даних в задачах класифікації текстів

Образи повнотекстових документів. Вхідними даними алгоритму класифікації є не сама колекція документів $D=\{d_i\}, i=\overline{1,|D|}$, а множина образів кожного документа $\vec{D}=\{\vec{d_i}\}, i=\overline{1,|D|}$, де $\vec{d_i}\in\vec{D}$ – образ документа $d_i\in D$. Існує кілька підходів до формування образів, застосовують той, який відповідає моделі, покладеної в основу конкретного алгоритму класифікації. Образи документів в тих алгоритмах, які розглядатимуться, представлені в наступному вигляді [31]:

а) мультимножин термінів документів (наприклад, наївний байєсівський класифікатор);

б) векторів в просторі термінів (наприклад, алгоритм Роккі, алгоритми класифікації без вчителя). В атестаційній роботі використаний інструмент Word2Vec.

Під термінами документів будемо розуміти все поодинокі слова, яких зустрілись в тексті хоча б одного документа колекції, за винятком стоп-слів, тобто поширених слів, які не характеризують документи за змістом, наприклад, прийменників, сполучників тощо. До того ж, кожній формі слова, що зустрічається, наприклад, в різних відмінках і числах, буде відповідати один і той же термін, наприклад, дане слово в початковій формі (лематизація). В результаті отримуємо множину всіх термінів колекції $\mathrm{T}=\{t_k\},\ k=\overline{1,|T|}$.

Образом документа як вектора в просторі термінів є вектор дійсних чисел $\vec{d_i}=(d_{i1},...,d_{i|T|})^T$, де кожне дійсне число є координатою вектора, що відповідає конкретному терміну, і дорівнює вазі терміна в даному документі. Найбільш часто використовують наступний підхід до обчислення ваги терміна:

$$d_{ij} = \frac{w_{ij}}{\left\|\overrightarrow{w_i}\right\|}, w_{ij} = tf_{ij} \times \log\frac{|D|}{df_j}, \quad (2.1)$$

де tf_{ij} – частота терміна в документі, тобто кількість разів, яке j-ий термін зустрівся в і-му документі; df_j – документна частота, тобто кількість документів, в яких зустрівся j-ий термін; $\left\|\overrightarrow{w_i}\right\|$ – евклідова норма $\overrightarrow{w_i}$. Такі ваги називають нормованими вагами за формулою «TF-IDF» («частота терміна – зворотна документна частота»), $0 \le d_{ij} \le 1$. Вони мають такі властивості:

а) мають високі значення, якщо термін часто зустрічається в невеликому числі документів, тим самим посилюючи відміну цих документів від інших;

б) мають низькі значення, якщо термін зрідка зустрічається в якомусь документі або зустрічається у багатьох документів, тим самим знижуючи відмінність між документами.

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

Міри подібності та відмінностей між образами документів.

В даному випадку поняття відстані і подібності є взаємооберненими, відстань можна було б називати різницею. Вибір способу обчислення відстані впливає на результат класифікації. Часто застосовують такі варіанти:

$$dist(\overrightarrow{d_i}, \overrightarrow{d_j}) = \left(\sum_{k=1}^{|T|} \left| d_{ik} - d_{jk} \right|^r \right)^{1/r}, \quad (2.2)$$

де r – це параметр, заданий користувачем, $r \in R, r > 0$. 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, яка обчислюється як

максимум модуля різниці компонент цих векторів $dist(\overrightarrow{d_i}, \overrightarrow{d_j}) = \max_{k=1,..,|T|} \left| d_{ik} - d_{jk} \right|$.

Іншою часто використовуваною мірою подібності є косинусна міра:

$$sim(\overrightarrow{d_i}, \overrightarrow{d_j}) = \cos(\angle(\overrightarrow{d_i}, \overrightarrow{d_j})) = \frac{\sum_{k=1}^{|T|} d_{ik} d_{jk}}{\sqrt{\sum_{k=1}^{|T|} d_{ik}^2} \times \sqrt{\sum_{k=1}^{|T|} d_{jk}^2}}. \quad (2.3)$$

Якщо вектори ваг документів нормовані як в (2.1), то косинусна міра є скалярний добуток векторів. Якщо вектори ортогональні, то міра близькості дорівнює 0, якщо співпадають, то 1. Зауважимо, що в разі, коли вектори ваг термінів нормовані, значення евклідової відстані і косинусної міри відповідають один одному.

2.2 Відбір термінів для класифікації

Велика кількість термінів (ознак) документів в завданні класифікації призводить до ряду проблем, серед яких:

а) високі обчислювальні витрати, пов’язані, наприклад, з отриманням значень міри близькості між документами тощо,

б) низька якість класифікації, викликана наявністю великої кількості ознак зі слабкою класифікаційної здатністю. Такі ознаки часто називають шумовими, при їх додаванні до представлення документа помилка класифікації на нових даних зростає.

Зокрема, низька якість класифікації з учителем може привести до перенавчання класифікатора, тобто ефекту, що виникає, коли класифікатор налаштовувався більшою мірою на випадкових (шумових) характеристиках документів, а не на суттєвих для їх тематик (категорій). У такій ситуації алгоритм добре працює на тих даних, на яких він був навчений, і значно гірше на нових.

Таким чином, треба прагнути скоротити число терміном з множини T

так, щоб нова (скорочена) множина термінів $T'(|T'| << |T|)$ містила найбільш інформативні в деякому сенсі терміни.

Техніки скорочення розмірності простору термінів (редукції) застосовують двома способами: локально (скорочують кількість термінів для кожної категорії окремо) і глобально (працюють із загальною множиною термінів для всіх документів). Перший випадок застосовується для класифікації з учителем, другий – як для класифікації з учителем, так і без нього. Інша істотна відмінність між техніками редукції полягає в природі підсумкових термінів. Одні техніки досягають скорочення числа термінів шляхом відсіву деякого числа вихідних термінів, керуючись заданим критерієм відсіву, – техніки відбору ознак. Тоді $T' \subset T$. Інші техніки формують нові терміни шляхом комбінації або перетворення вихідних термінів, іншими словами видобувають підсумкові ознаки з вихідних даних – техніки видобування ознак. Тоді елементи множини T' мають тип, відмінний від елементів множини T. Відбір і вилучення термінів реалізуються різними техніками. Розглянемо тільки техніки відбору ознак (термінів).

2.3 Відбір ознак документів

Необхідно відібрати такі терміни, які підвищать якість класифікатора. Для цього помістимо в T' всі терміни з T, які мають високе значення «важливості для розбиття по категоріям/класам». Для визначення «важливості» терміна і способу її обчислення використовують різні підходи.

2.3.1 Документна частота (DF)

Найпростіша і цілком ефективна техніка оцінки «важливості термінів для класифікації» заснована на спостереженні того, що значна кількість термінів колекції зустрічаються в малому числі документів, а найбільшу інформативність мають терміни з середньою або навіть високою частотою,

якщо попередньо були видалені стоп-слова. На практиці часто використовують порогове значення τ, рівне 1-5 документам. Таким чином, $T = \{t_k \in T : DF(t_k) > \tau\}$ $DF(t_k)$ – це кількість документів, в яких зустрічається термін t_k. Дана техніка може застосовуватися як єдина, так і передувати іншій техніці відбору ознак.

Наступні техніки беруть свій початок з теорії інформації: взаємна інформація, інформаційна вигода і критерій хі-квадрат. Розглянемо 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 $\max_{j=1,|C|} f(t_k, c_j)$.

2.3.2 Взаємна інформація (MI)

Величина взаємної інформації терміна t і категорії c:

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

Нехай A – кількість документів, що належать категорії c і містять термін t;

B – кількість документів, які не належать категорії c і містять термін t;

C – кількість документів, що належать категорії c і не містять термін t.

Тоді вираз (2.4) можна записати в такий спосіб:

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

де Ω – навчальна множина документів; $MI(t_k, c_j)$ приймає значення 0, якщо термін t і категорія c є незалежними.

Недолік взаємної інформації полягає в тому, що її значення сильно піддається впливу безумовної ймовірності термінів, так як $MI(t_k, c_j) = \log_2 P(t_k | c_j) - \log_2 P(t_k)$ (це випливає з (2.4)). Якщо два терміни мають однакову умовну ймовірність, більш високе значення MI буде у більш рідкісного. Отже, значення взаємної інформації не можна порівнювати для термінів з істотно різною частотою зустрічальності в документах.

2.3.3 Інформаційна вигода (IG)

Інформаційну вигоду часто називають очікуваною взаємною інформацією (EMI). Цей показник вимірює кількість інформації про належність до категорії c, на яке вказує наявність / відсутність терміна t.

$$IG(t_k, c_j) = \sum_{c \in \{c_j, \overline{c_j}\}} \sum_{t \in \{t_k, \overline{t_k}\}} P(t,c) \times \log_2 \frac{P(t,c)}{P(c) \times P(c)}, \qquad (2.6)$$

де $\overline{c_j}$ – всі категорії, крім c_j; $t_k, \overline{t_k}$ – ознаки наявності і відсутності терміну t_k відповідно.

На практиці формула (6) еквівалентна наступній:

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

де D – кількість документів, які не належать категорії с і не містять термін t.

Міра інформаційної вигоди досягає свого максимуму, коли термін є ідеальним індикатором категорії, тобто присутній в документі тоді і тільки тоді, коли документ належить класу. Якщо розподіл терміна в категорії відповідає розподілу терміна в колекції, то інформаційна вигода дорівнює 0.

При заданій навчальній множині для кожного терміна обчислюють значення IG і видаляють з T такі терміни, значення інформаційної вигоди яких нижче деякого заздалегідь обраного порогового значення.

2.3.4 Критерій хі-квадрат (CHI)

Критерій χ^2 використовується для перевірки незалежності двох випадкових подій: поява терміна X і поява класу Y. Якщо X і Y незалежні, то $P(XY) = P(X) \cdot P(Y)$. Критерій χ^2 обчислюється за формулою:

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

де $P(t,c)$ – спостережена на навчальній множині, $P_{\exp}(t,c)$ – очікувана за умови, що термін і клас є незалежними. Величина критерію χ^2 дозволяє судити про те, наскільки очікувана і спостережена ймовірності відхиляються одна від одної, і приймає значення 0, якщо термін і категорія незалежні. Критерій обчислюють локально (для кожної категорії), потім отримують його глобальне значення, за яким ранжують ознаки колекції документів.

На практиці формула (8) еквівалентна наступній:

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

На відміну від взаємної інформації критерій χ^2 нормалізований, що дозволяє порівнювати між собою його значення для різних термів однієї категорії, винятком є лише рідкісні терми.

2.4 Видобування ознак документів

Необхідно синтезувати нові (штучні) ознаки документів так, щоб підвищити якість класифікації, наприклад, шляхом розв’язання неоднозначностей природної мови (синонімії, омонімії, полісемії). Потім слід відобразити документи колекції в новий простір ознак, який позбавлений старих проблем і краще, ніж вихідний, представляє зміст документів. Прикладом техніки видобування ознак документів є тематичне моделювання, яке використовує латентно-семантичний аналіз, латентний аналіз Діріхле, алгоритми регулярізації.

2.4.1 Тематичне моделювання

Для початку розглянемо декілька гіпотез.

Гіпотеза про існування тем. Кожне входження терма w в документ d пов'язано з деякою темою t з заданої скінченої множини T. Колекція документів являє собою послідовність трійок $\Omega_n = \{(d_i, w_i, t_i) | \; i = 1, \dots, n\}$. Терми w_i і документи d_i є спостереженими змінними, теми t_i не відомі і є латентними (прихованими) змінними.

Гіпотеза «мішка слів». Порядок термів в документах не важливий для виявлення тематики, тобто тематику документа можна дізнатися навіть після довільної перестановки термів, хоча для людини такий текст втратить сенс. Це припущення називають гіпотезою «мішка слів» (bag of words). Порядок документів в колекції також не має значення – це припущення називають гіпотезою «мішка документів». Гіпотеза «мішка слів» дозволяє перейти до компактного подання документа як мультимножини – підмножини термів $d \subset W$, в якому кожен терм $w \in d$ повторений n_{dw} раз.

Гіпотеза про імовірнісне породження даних. Множина $\Omega = \mathrm{D} \times \mathrm{W} \times \mathrm{T}$ є кінцевим імовірнісним простором з невідомою функцією ймовірності $p(d, w, t)$. Колекція документів є вибіркою трійок (d_i, w_i, t_i), породжуваною

випадково і незалежно одна від одної з розподілу $p(d,w,t)$. Це припущення є імовірнісним уточненням гіпотези «мішка слів».

Завдяки припущенню про незалежність, реалізовану вибірку Ω_n елементів з Ω можна розглядати як новий імовірнісний простір з n рівноімовірними елементарними наслідками. У просторі Ω_n легко знаходити ймовірності різних подій, причому вони збігаються з частотними оцінками ймовірностей тих же подій в просторі Ω. Зокрема, в просторі Ω_n вираз

$$\hat{p}(d,w,t) = \frac{1}{n}\sum_{i=1}^{n}[d_i = d][w_i = w][t_i = t]$$

дорівнює ймовірності того, що терм w документа d пов'язаний з темою t, а в просторі Ω він дорівнює вибірковій частотній оцінці ймовірності тієї ж події.

Гіпотеза умовної незалежності. Поява термів в документі d по темі t залежить від теми, але не залежить від документа d, і описується загальним для всіх документів розподілом $p(w|t)$:

$$p(w|d,t) = p(w|t) \qquad 2.10$$

Імовірнісна тематична модель породження тексту. Відповідно до формули повної ймовірності та гіпотезі умовної незалежності, розподіл термів в документі $p(w|d)$ описується ймовірнісною сумішшю розподілів термів в темах $\varphi_{wt} = p(w|t)$ з вагами $\theta_{td} = p(t|d)$:

$$p(w|d) = \sum_{t\in T} p(w|t,d)p(t|d) = \sum_{t\in T} p(w|t)p(t|d) = \sum_{t\in T} \varphi_{wt}\theta_{td} \qquad (2.11)$$

Імовірнісна модель (2.11) описує процес породження колекції за відомим розподілом $p(w|t)$ і $p(t|d)$.

Задача тематичного моделювання – це зворотна задача: по заданій колекції D потрібно знайти параметри φ_{wt} і θ_{td}, при яких тематична модель

(2.11) добре наближає частотні оцінки умовних ймовірностей $p(w|d) = n_{dw}/n_d$.

Формальна постановка задачі. На вхід подається W – словник термінів (уніграм чи біграм), D – колекція документів $d \subset D$, n_{dw} – лічильник частоти виникнення слова w в документі d. Треба знайти модель $p(w|d) = \sum_{t \in T} \varphi_{wt} \theta_{td}$ з параметрами $\underset{W \times T}{\Phi}$ і $\underset{T \times D}{\Theta}$: $\varphi_{wt} = p(w|t)$ – ймовірності термінів w в кожній темі t; θ_{td} – ймовірності тем t в кожному документі d. Критерієм оберемо максимізацію логарифма правдоподібності:

$$\sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} \varphi_{wt} \theta_{td} \to \max_{\Phi, \Theta}; \tag{2.12}$$

$$\sum_{w \in W} \varphi_{wt} = 1;\ \varphi_{wt} \geq 0;\ \sum_{t \in T} \theta_{td} = 1;\ \theta_{td} \geq 0. \tag{2.13}$$

Розподіл виду $p(t|x)$ будемо називати тематикою об'єкта x. Можна говорити про тематику документа $p(t|d)$, терма $p(t|w)$, терма в документі $p(t|d,w)$.

Метою тематичного моделювання є визначення тематики документів і пов'язаних з ними об'єктів. Також потрібно знаходити розподіли $\varphi_{wt} = p(w|t)$, що описують семантику кожної теми t словами природної мови.

Низькорангове матричне розкладання. Рівність (2.11) можна переписати в матричному вигляді. У лівій частині рівності знаходиться відома матриця частот термів в документах $F = (\hat{p}(w|d))_{W \times D}$. Права частина являє добуток двох невідомих матриць – матриці термів тем $\Phi = (\varphi_{wt})_{W \times T}$ і матриці тем документів $\Theta = (\theta_{td})_{T \times D}$. Зазвичай число тем $|T|$ багато менше $|D|$ і $|W|$, тому завдання тематичного моделювання зводиться до пошуку наближеного матричного розкладання $F \approx \Phi\Theta$, ранг якого не перевищує $|T|$.

Всі три матриці F, Φ, Θ є стохастичними, тобто мають невід'ємні

нормовані стовпці f_d, φ_t, θ_d, що представляють дискретні розподіли. Добуток ΦΘ називається стохастичним матричним розкладанням.

Частотні оцінки умовних ймовірностей. У просторі Ω_n ймовірності, що виражаються через змінні d і w, збігаються з частотами відповідних спостережуваних подій:

$$p(d,w)=\frac{n_{dw}}{n}, \qquad p(d)=\frac{n_d}{n}, \qquad p(w)=\frac{n_w}{n}, \quad p(w|d)=\frac{n_{dw}}{n_d}; \qquad (2.14)$$

де n_{dw} – число входжень терма w в документ d;
$n_d=\sum_w n_{dw}$ – довжина документа d в термах;
$n_w=\sum_d n_{dw}$ – число входжень терма w у всі документи колекції;
$n=\sum_d \sum_w n_{dw}$ – довжина колекції в термах.

Ймовірності, пов'язані з прихованою змінною t, теж визначаються як частоти:

$$p(t)=\frac{n_t}{n}, \quad p(w|t)=\frac{n_{wt}}{n_t}, \quad p(t|d)=\frac{n_{td}}{n_d}, \quad p(t|d,w)=\frac{n_{tdw}}{n_{dw}}, \qquad (2.15)$$

де n_{tdw} – число трійок, у яких терм w документа d з'язаний з темою t;
$\mathrm{n_{td}}=\sum_{\mathrm{w}} \mathrm{n_{tdw}}$ – число трійок, у яких терм документа d связан с темой t.

На відміну від (2.14), ці частотні оцінки не можуть бути обчислені безпосередньо за вихідними даними, так як теми t_i невідомі.

Відповідно до закону великих чисел, при $n\to\infty$ частотні оцінки, які визначаються формулами (2.14) – (2.15), наближаються до відповідних ймовірностей в просторі Ω.

ЕМ-алгоритм. Зауважимо, що всі оцінки (2.15) виражаються через $n_{tdw}=n_{dw}p(t\,|\,d,\,w)$. Знаючи умовні розподіли $p(t\,|\,d,\,w)$, можна оцінити шукані параметри тематичної моделі $\varphi_{wt}=p(w|t)$ і $\theta_{td}=p(t\,|\,d)$. І, навпаки, знаючи параметри моделі можна виразити умовні ймовірності $p(t\,|\,d,\,w)$ за

формулою Байєса:

$$p(t|d,w) = \frac{p(t,w|d)}{p(w|d)} = \frac{p(w|t)p(t|d)}{p(w|d)} = \frac{\varphi_{wt}\theta_{td}}{\sum_s \varphi_{ws}\theta_{sd}}$$

Таким чином, отримуємо систему нелінійних рівнянь щодо параметрів моделі φ_{wt}, θ_{td} і допоміжних змінних p_{tdw}, n_{wt}, n_{td}:

$$p_{tdw} = \frac{\varphi_{wt}\theta_{td}}{\sum_s \varphi_{ws}\theta_{sd}}; \quad (2.16)$$

$$\varphi_{wt} = \frac{n_{wt}}{\sum_{w'} n_{w't}}; n_{wt} = \sum_{d\in D} n_{dw} p_{tdw}; \quad (2.17)$$

$$\theta_{td} = \frac{n_{td}}{\sum_{t'} n_{t'd}};\ n_{td} = \sum_{w\in d} n_{dw} p_{tdw.} \quad (2.18)$$

Для її вирішення зручно застосовувати метод простих ітерацій: спочатку обираються початкові наближення параметрів φ_{wt} і θ_{td}, по ним обчислюються допоміжні змінні p_{tdw}, які дозволяють знайти наступне наближення параметрів φ_{wt} і θ_{td}. Обчислення за формулами (2.16) – (2.18) продовжуються в циклі до збіжності.

Цей ітераційний процес є окремим випадком EM-алгоритму, призначеного для побудови імовірнісних моделей з прихованими змінними [32]. Обчислення умовних розподілів прихованих змінних (2.16) називається E-кроком (expectation), обчислення параметрів моделі (2.17) – (2.18) M-кроком (maximization).

2.4.2 Регулярізація

Задача називається коректно поставленою за Адамаром, якщо її рішення існує, є єдиним і стійким.

Задача стохастичного матричного розкладання є некоректно поставленою, так як множина її рішень в загальному випадку нескінчена. Якщо $\Phi\Theta$ – рішення, то $(\Phi S)(S^{-1}\Theta)$ так само є рішенням для всіх невироджених матриць S, за умови, що матриці ΦS і $S^{-1}\Theta$ – стохастичні.

Існує загальний підхід до вирішення некоректно поставлених обернених задач, званий регуляризацією [33]. Коли оптимізаційна задача недовизначеною, до основного критерію додають додатковий – критерій регуляризатора, що враховує специфіку розв'язуваної задачі і знання предметної області. У практичних задачах автоматичної обробки текстів додаткових критеріїв та обмежень на рішення може бути багато.

Аддитивна регуляризація тематичних моделей (ARTM) [34] заснована на максимізації лінійної комбінації логарифма правдоподібності і регуляризатора $R_i(\Phi,\Theta)$ з невід'ємними коефіцієнтами регуляризації $\tau_i, i = 1, \ldots, k$:

$$\sum_{d\in D}\sum_{w\in d} n_{dw} ln \sum_{t\in T}\varphi_{wt}\theta_{td} + R(\Phi,\Theta) \to \max_{\Phi,\Theta};\ R(\Phi,\Theta) = \sum_{i=1}^{k}\tau_i R_i(\Phi,\Theta); \qquad (2.19)$$

за обмежень невід'ємності та нормування:

$$\sum_{w\in W}\varphi_{wt} \in \{0,1\};\ \varphi_{wt} \geq 0;\ \sum_{t\in T}\theta_{td} \in \{0,1\}; \theta_{td} > 0. \qquad (2.20)$$

Умови нормування з альтернативною правою частиною (2.20) послаблюють умови (2.13), допускаючи появу нульових стовпців в матрицях рішення Φ і Θ.

Перетворення вектора критеріїв в один скалярний критерій – це один з базових прийомів багатокритеріальної оптимізації, званий скаляризацією.

Задача тематичного моделювання по суті є багатокритеріальною. Теми повинні задовольняти багатьом вимогам одночасно: інтерпретованості, розбіжності, розрідженості тощо. Крім того, тематична модель зазвичай

використовується не сама по собі, а як допоміжний інструмент для вирішення різних завдань текстової аналітики: інформаційного пошуку, візуалізації, категоризації, сегментації, суммарізаціі тощо. Кожне завдання висуває свої вимоги до моделі. У ARTM всі вимоги формалізуються у вигляді критеріїв регуляризації R_i і балансуються за допомогою коефіцієнтів τ_i. Коефіцієнти τ_i доводиться підбирати в кожному задачі експериментально щоб знайти компроміс між усіма критеріями. Більш того, для вимірювання якості моделі зазвичай використовуються не самі регуляризатори R_i, а якісь інші метрики якості. Регуляризатори мають бути гладкими функціями, зручними для обчислень на М-кроці. Метрики якості повинні мати зручні для інтерпретації числові значення. На жаль, ці вимоги часто суперечать одна одній. Наприклад, загальноприйняті метрики якості інформаційного пошуку майже ніколи не є гладкими функціями.

На практиці проблема вибору коефіцієнтів регуляризації переростає в більш загальну проблему управління якістю моделі, оскільки ніщо не забороняє змінювати коефіцієнти τ_i в ході ітерацій. Одні регуляризатори можуть робити підготовчу роботу для інших. Деякі регуляризатори рекомендується включати, коли ЕМ-алгоритм вже почав сходитися. Інші краще відключати після того, як вони виконали свою роботу. Деякі регуляризатори можуть нейтралізувати один одного, і тоді їх доводиться застосовувати по черзі. Систематизація і використання цих ефектів стає предметом дослідження в ARTM.

Стратегією регуляризації називається функція коефіцієнтів регуляризації τ_i від номера ітерації. Зокрема вона може використовувати поточні значення параметрів моделі і метрик якості.

2.4.3 Латентно-семантичний аналіз (ЛСА)

Вперше ЛСА був застосований для автоматичного індексування

текстів, виявлення семантичної структури тексту і отримання псевдодокументу [35]. Потім цей метод був досить успішно використаний для представлення баз знань [36] і побудови когнітивних моделей [37].

В останні роки метод ЛСА часто використовується для пошуку інформації (індексація документів), класифікації документів, у моделях розуміння та інших областях, де потрібно виявлення головних факторів з масиву інформаційних даних.

Алгоритм ЛСА (спочатку відомий як «Латентне семантичне індексування» («Latent Semantic Indexing», LSI)) розроблений для вирішення завдань пошуку і видобування інформації (information retrieval) і являє собою виділення з великої бази даних невеликої кількості документів, релевантних заданому запиту.

Опис роботи ЛСА [38].

На вхід ЛСА подається матриця терми-на-документи (терми – слова, словосполучення або n-грами; документи – тексти, класифіковані або за будь-яким критерієм, або розділені довільним чином – це залежить від розв’язуваної задачі), що описує набір даних, який використовується для навчання системи. Елементи цієї матриці містять, як правило, ваги, що враховують частоти використання кожного терма в кожному документі або імовірнісні міри (PLSA – ймовірнісний латентно-семантичний аналіз), засновані на незалежному мультимодальному розподілі.

Найбільш поширений варіант ЛСА заснований на використанні розкладання дійснозначної матриці по сингулярним значенням або SVD-розкладання (SVD – Singular Value Decomposition). За допомогою нього будь-яку матрицю можна розкласти в множину ортогональних матриць, лінійна комбінація яких є досить точним наближенням до початкової матриці.

Згідно з теоремою про сингулярне розкладання в найпростішому випадку матриця може бути розкладена на добуток трьох матриць:

$$A = USV^T$$

де матриці U і V – ортогональні, а S – діагональна матриця, значення на діагоналі якої називаються сингулярними значеннями матриці A.

Особливість такого розкладу в тому, що якщо в матриці S залишити тільки k найбільших сингулярних значень, а в матрицях U і V – тільки відповідні цим значенням стовпці, то лінійна комбінація одержаних матриць буде найкращим наближенням початкової матриці A до матриці $\hat{A}$ рангу k:

$$\hat{A} \approx A = USV^T$$

Основна ідея латентно-семантичного аналізу полягає в тому, що якщо в якості матриці A використовувалася матриця терми-на-документи, то матриця $\hat{A}$, яка містить тільки k перших лінійно незалежних компонент A, відображає основну структуру різних залежностей, присутніх у вихідній матриці. Структура залежностей визначається ваговими функціями термів.

Таким чином, кожен терм і документ представляються за допомогою векторів в загальному просторі розмірності k (так званому просторі гіпотез). Близькість між будь-якою комбінацією термів і/або документів легко обчислюється за допомогою скалярного добутку векторів.

Як правило, вибір k залежить від поставленого завдання і підбирається емпірично. Якщо вибране значення k занадто велике, то метод втрачає свою потужність і наближається за характеристиками до стандартних векторних методів. Занадто мале значення k не дозволяє вловлювати відмінності між схожими термами або документами.

Ілюстрація процесу виявлення тематик в матриці «документи-слова» наведена на рис. 2.1. Кожен стовпець матриці відповідає документу, кожен рядок – слову. Осередки матриці містять ваги слів в документах (наприклад, значення TF-IDF), більш темні відтінки відповідають більш високій вазі. Алгоритм LSA групує як документи, які використовують схожі слова, так і слова, які зустрічаються в схожому наборі документів. Отримані кластери в

матриці використовуються для виявлення латентних (прихованих) компонентів у вихідних даних, що відповідають певним тематикам [39].

Основна ідея латентно-семантичного аналізу полягає в наступному: після перемноження матриць отримана матриця $\hat{A}$, що містить тільки k перших лінійно незалежних компонент вихідної матриці A, відображає структуру залежностей (в даному випадку асоціативних), латентно присутніх у вихідній матриці. Структура залежностей визначається ваговими функціями термів для кожного документа.

Вибір k залежить від поставленого завдання і підбирається емпірично. Він залежить від кількості вихідних документів. Якщо документів не багато, наприклад сотня, то k можна брати 5-10% від загального числа діагональних значень; якщо документів сотні тисяч, то беруть 0,1-2%. Слід пам'ятати, що якщо вибране значення k занадто велике, то метод втрачає свою потужність і наближається за характеристиками до стандартних векторних методів. А надто маленьке значення k не дозволяє вловлювати відмінності між схожими термами або документами: залишиться тільки одна головна компонента, яка «перетягне на себе ковдру», тобто всі слабо і навіть незв'язані терми.

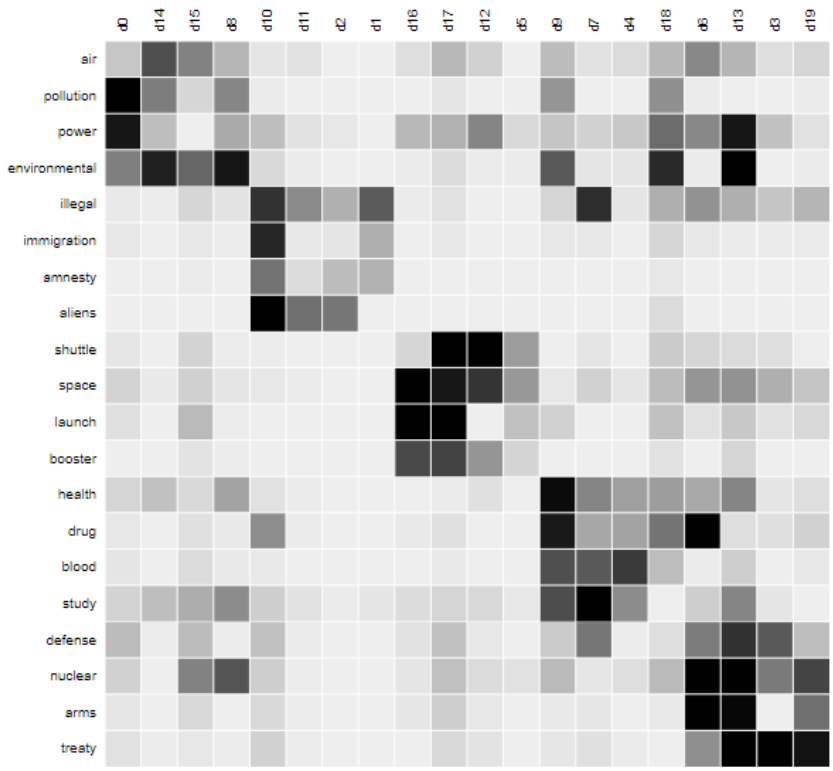

Рисунок 2.1 – Ілюстрація виявлення тематик в матриці «документи-слова»

Обсяг корпусу для побудови моделі має бути великим – бажано близько трьох-п'яти мільйонів слововживань. Але метод працює і на колекціях меншого обсягу, правда дещо гірше.

Довільне розбиття тексту на документи зазвичай виробляє від тисячі до кількох десятків тисяч частин приблизно однакового обсягу. Таким чином, матриця терми-на-документи виходить прямокутною і може бути сильно розрідженою. Наприклад, при обсязі 5 млн. словоформ виходить матриця 30-50 тисяч документів на 200-300 тисяч, а іноді і більше, термів. Насправді, низькочастотні терми можна опустити, тому що це помітно знизить розмірність матриці, що призведе до зниження обчислювальних ресурсів і часу.

Вибір скорочення сингулярних значень діагональної матриці (розмірності k) при зворотному перемножуванні матриць є довільним. При вищевказаній розмірності матриці залишають кілька сотень (100-300) головних компонент. При цьому, як показує практика, залежність кількості компонент і точність змінюються нелінійно: наприклад, якщо починати збільшувати їх число, то точність буде падати, але при деякому значенні, скажімо, 10000 – знову виросте до оптимального випадку.

Застосування ЛСА:

- порівняння двох термів між собою;
- порівняння двох документів між собою;
- порівняння терма і документа.

Також іноді цей метод використовують для знаходження «найближчого сусіда» – найбільш близьких по вазі термів, асоціативно пов'язаних з вихідним. Це властивість використовують для пошуку близьких за змістом термів.

Слід зауважити, що близькість за змістом – це контекстнозалежна величина, тому не всякий близький терм буде відповідати асоціації (це може бути і синонім, і антонім, і просто слово або словосполучення, яке часто

зустрічається разом з шуканим).

Переваги та недоліки ЛСА.

Перевагою методу можна вважати його здатність виявляти залежності між словами, коли звичайні статистичні методи безсилі. ЛСА також може бути застосований як з навчанням (з попередньої тематичною класифікацією документів), так і без навчання (довільне розбиття довільного тексту), що залежить від розв'язуваної задачі.

До недоліків можна віднести значне зниження швидкості обчислення при збільшенні обсягу вхідних даних.

Для запобігання «прокляття розмірності» та досягнення кращої обумовленності матрицю «слова-документи» доцільно представити у вигляді матричного розкладання (рис. 2.2).

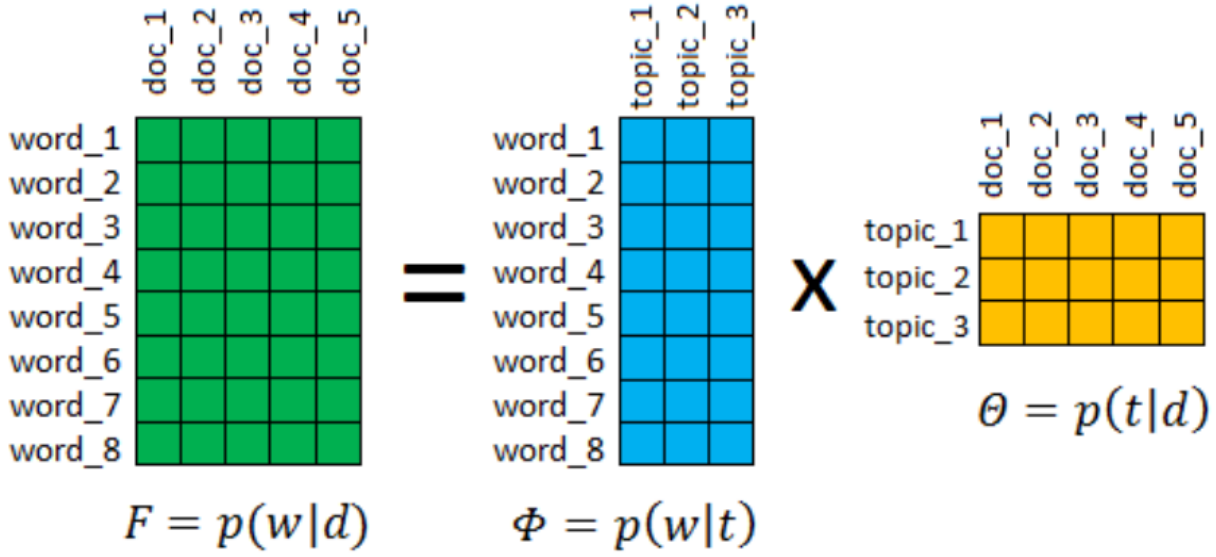

Рисунок 2.2 – Розкладання матриці «слова-документи» на дві матриці «слова-теми» та «теми-документи»

2.4.4 Модель латентного розміщення Діріхле (ЛДА)

Девід Блей, Ендрю Ин і Майкл Джордан запропонували модель LDA

(latent Dirichlet allocation) для вирішення проблеми перенавчання в PLSA, яка передбачала ймовірності термів $p(w|d)$ на нових документах помітно гірше, ніж на навчальній колекції [40]. Пізніше з'ясувалося, що на великих колекціях обидві моделі майже не перенавчаються, а їх правдоподібності відрізняються незначно [41]. Відмінності проявляються лише на низькочастотних термах, які не важливі для утворення тем. У робастних варіантах PLSA і LDA такі терми ігноруються, що різко знижує як перенавчання, так і відмінність в правдоподібності моделей [42]. Саме питання про перенавчання поставлено не цілком коректно. По-перше, тематичні моделі будуються не заради передбачення термів в документах, а для виявлення кластерної структури колекції. По-друге, величина перенавчання залежить не тільки від самої моделі, а й від того, як домовляються вимірювати її якість. Для вимірювання зазвичай використовується перплексія, яка сильно штрафує занижені ймовірності низькочастотних термів. Можливо, перевага LDA над PLSA не настільки істотна, як це прийнято вважати. Проте, LDA де-факто є найбільш використовуваною тематичної моделлю.

Модель LDA заснована на припущенні, що стовпці θ_d і φ_t є випадковими векторами, які породжуються розподілами Діріхле з параметрами $\alpha \in R^{|T|}$ і $\beta \in R^{|W|}$ відповідно:

$$\mathrm{Dir}(\theta_d;\alpha) = \frac{\Gamma(\alpha_0)}{\prod_w \Gamma(\alpha_t)} \prod_w \theta_{td}^{\alpha_t - 1}, \alpha_t > 0,\ \alpha_0 = \sum_t \alpha_t, \theta_{td} > 0, \sum_t \theta_{td} = 1;$$

$$\mathrm{Dir}(\varphi_t;\beta) = \frac{\Gamma(\beta_0)}{\prod_w \Gamma(\beta_w)} \prod_w \varphi_{wt}^{\beta_w - 1}, \beta_w > 0, \beta_0 = \sum_w \beta_w, \varphi_{wt} > 0, \sum_w \varphi_{wt} = 1;$$

де $\Gamma(z)$ – гамма-функція. Параметри розподілу Діріхле пов'язані з математичним очікуванням породжуваних випадкових векторів: $E\theta_{td} = \alpha_t / \alpha_0, E\varphi_{wt} = \beta_w / \beta_0$.

Розподіли Діріхле здатні породжувати як розріджені, так і щільні

вектори дискретних розподілів. Чим менше β_w, тим більше розріджена відповідна w компонента φ_{wt} в породжуваних векторах φ_t. Якщо вектор параметрів складається з рівних значень β_w, то розподіл Діріхле називається симетричним. При β_{w-1} симетричний розподіл Діріхле збігається з рівномірним розподілом на одиничному симплексі.

Тематична модель породження даних є дворівневою: спочатку з розподілу Діріхле породжуються вектор-стовпчики φ_t, які задають теми. Потім з отриманих розподілів $p(w|t)=\varphi_{wt}$ породжуються терми, що утворюють тематичні частини документів $p(w|t,d)$. Таким чином, дворівнева модель описує кластерні структури в текстових колекціях. Вектори розподілів $p(w|t)$ інтерпретуються як центри кластерів, а розподіли $p(w|t,d)$ є точками цих кластерів.

Більш переконливих лінгвістичних обґрунтувань розподіл Діріхле не має. Його широке поширення в тематичному моделюванні пояснюється швидше за його чисто математичною зручністю для байєсівського навчання. Розподіл Діріхле є зв'язаним з поліноміальним розподілом, що істотно спрощує байесовске виведення. Завдяки цій властивості воно опиняється «на особливому положенні» в байесовскому тематичному моделюванні, і більшість моделей будуються з використанням розподілів Діріхле.

Байєсівська регуляризація. До цього часу ми припускали, що дані породжуються ймовірнісною моделлю з параметрами (Φ, Θ), які не відомі і не випадкові. У байєсовському підході передбачається, що параметри також випадкові і підпорядковуються деякому апріорному розподілу $p(\Phi,\Theta;\gamma)$ з невипадковим гіперпараметром γ. В цьому випадку максимізація спільної правдоподібності даних і моделі призводить до принципу максимуму апостеріорної ймовірності (maximum posteriori probability, MAP):

$$p(D,\Phi,\Theta;\gamma) = p(D|\Phi,\Theta)p(\Phi,\Theta;\gamma) = p(\Phi,\Theta;\gamma)\prod_{i=1}^{n}\prod p(d_i w_i|\Phi,\Theta) \to \max_{\Phi,\Theta,\gamma}$$

Після логарифмування отримуємо модифікацію задачі (2.12), в якій логарифм апріорного розподілу є регуляризатором:

$$\sum_{d\in D}\sum_{w\in d} n_{dw} \ln \sum_{t\in T} \varphi_{wt}\theta_{td} + \underbrace{\ln p(\Phi,\Theta;\gamma)}_{R(\Phi,\Theta)} \to \max_{\Phi,\Theta;\gamma} \qquad (2.21)$$

У байєсівському підході застосовується також принцип максимізації неповної правдоподібності, в якому за випадковими параметрами (Φ,Θ) проводиться інтегрування і оптимізуються гіперпараметри γ. Вважається, що цей прийом знижує розмірність задачі та ризик перенавчання. Дійсно, розмірність вектора γ, як правило, набагато менше розмірів матриць Φ,Θ і не залежить від обсягу колекції. Однак для вирішення прикладних задач все одно потрібні саме ці матриці. Формули для них виводяться громіздкими наближеними методами, але в підсумку мало відрізняються від MAP-оцінок [43].

В байєсівському підході оцінюються не власне параметри Φ,Θ, а їх апостеріорний розподіл $p\left(\Phi, \Theta| D; \gamma\right)$. Для задач тематичного моделювання в цьому немає особливої необхідності. На практиці отриманий розподіл використовується виключно для того, щоб повернутися до точкових оцінок математичного очікування. Ні апостеріорні розподіли, ні інтервальні оцінки, ні навіть альтернативні точкові оцінки медіани або моди в додатках не використовуються.

Згідно (2.21), моделі LDA відповідає регуляризатор, з точністю до константи рівний логарифму апріорного розподілу Діріхле:

$$R(\Phi,\Theta) = \ln \prod_{t\in T} Dir(\varphi_t;\beta) \prod_{d\in D} Dir(\theta_d;\alpha) + const == \sum_{t\in T}\sum_{w\in W}(\beta_w - 1)\ln\varphi_{wt} + \sum_{d\in D}\sum_{t\in T}(\alpha_t - 1)\ln\theta_{td}. \qquad (2.22)$$

Застосування рівнянь пошуку локального екстремуму до цього регуляризатора дає формули М-кроку:

$$\varphi_{wt} = \underset{w \in W}{norm}(n_{wt} + \beta_w - 1); \qquad \theta_{td} = \underset{t \in T}{norm}(n_{td} + \alpha_t - 1).$$

При $\beta_w = 1$, $\alpha_t = 1$ апріорний розподіл Діріхле збігається з рівномірним розподілом на симплексі, формули М-кроку переходять в незміщені частотні оцінки умовних ймовірностей, а модель LDA переходить в PLSA [44].

При $\beta_w > 1$, $\alpha_t > 1$ регуляризатор має згладжувальний ефект, він робить великі ймовірності ще більше, при цьому малі ймовірності за рахунок нормування стають менше, проте ніколи не досягають нуля.

При $0 < \beta_w < 1$, $0 < \alpha_t < 1$ регуляризатор має розріджувальний ефект і здатний обнуляти малі ймовірності.

В атестаційній роботі стемінг проведено за допомогою стемера SnowballStemmer, формування векторів документів – за допомогою Word2Vec, для зменшення розмірності векторів був використаний алгоритм t-SNE, для тематичного моделювання використовується модель LDA як найбільш поширена для такого роду задач.

3 ЕКСПЕРИМЕНТАЛЬНЕ МОДЕЛЮВАННЯ ТА НАВЧАННЯ МОДЕЛІ

Для експериментального моделювання та навчання моделі було обрано текстову коллекцію EUR-Lex. EUR-Lex є корпусом текстів на 24 мовах Євросоюзу. Користувачі можуть отримати вільний доступ до 3,9 мільонів текстових документів. Також у корпусі зберігаються метадані документів. Документи включають в себе договори, міжнародні угоди, законодавчі пропозиції, прецедентне право, парламентські питання і ряд інших законодавчих документів. Обробка даних юридичних текстів у Європейскій комісії почалась ще у 1960-х роках, тоді ще використовували перфокарти. Система розроблялась для аналізу співвідношень між документами та подальшого видобутку та повторного використання метаданих, ще одна функція – спрощення пошуку за документами.

В атестаційній роботі використовувались два файли даних – перший містить два стовпчики: перший – ідентифікатор документа, другий – сам документ у текстовому форматі; другий файл містить три стовпчики: перший – ім’я мітки класу, другий – документ, до якого ця мітка належить (у кожного документа може бути декілька міток класів), третій – константа 1 (не приймається до уваги). Колекція містить близько 20 тисяч документів. Словник колекції становить $\approx 1.9 \times 10^5$ слів. Число класів (міток), за якими розбиті документи колекції, приблизно дорівнює 3950. При цьому кожен документ може відноситься до декількох класів одночасно. Щоб перевірити розподіл міток в наборі даних, був побудований графік (рис. 3.1), який показує кількість входжень кожної мітки (на графіку представлені 100 міток, які найбільш часто зустрічаються).

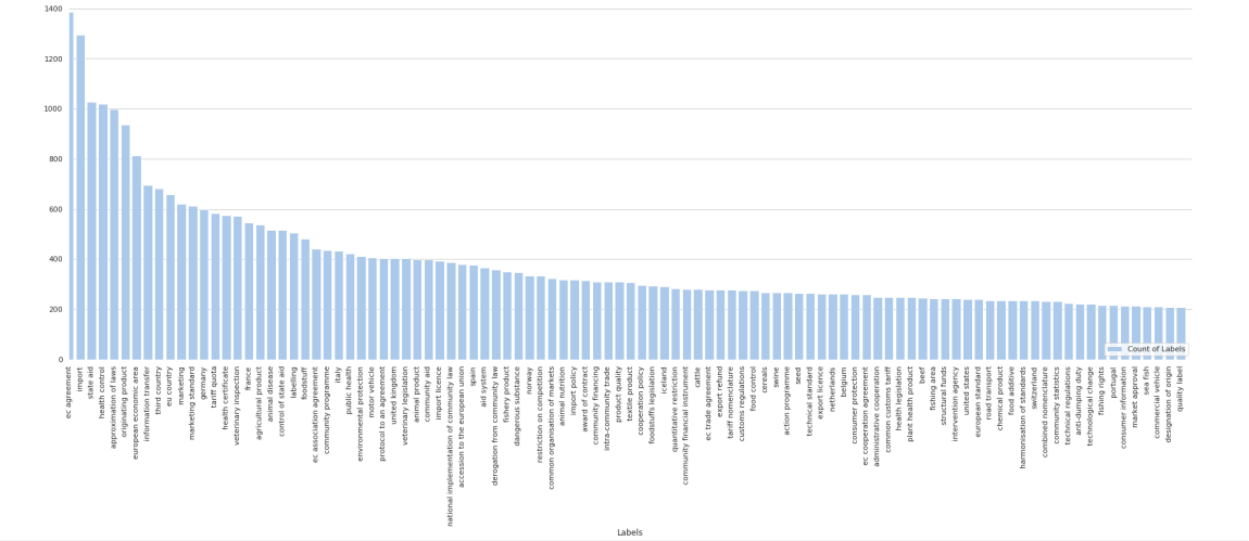

Рисунок 3.1 – Розподіл 100 міток, які найбільш часто зустрічаються в колекції EUR-Lex

3.1 Попередня обробка корпусу текстів

На рис. 3.2 наведена хмара слів, отримана до попередньої обробки.

Рисунок 3.2 – Хмара слів перед передобробкою

В першу чергу необхідно провести попередню обробку текстових документів, а саме: видалити пунктуаційні знаки, характерні стоп-слова, провести стемінг текстів. Для цього використовувалась бібліотека NLTK – вона містить алгоритми лематизації та стемінга, словники стоп-слів (рис. 3.3).

```
import nltk
from nltk.corpus import stopwords
```

Рисунок 3.3 – Імпорт бібліотеки NLTK та словника стоп-слів

Модіфікуємо структуру текстового документу з даними для зручності подальшої роботи, додаємо лейбли структурним стовпцям, а саме: label – для ідентифікатора документа, та document для текста документа. Підключаємо програмну бібліотеку pandas, завантажуємо необхідний файл та перевіряємо коректність зчитування (рис. 3.4, 3.5).

```
import pandas as pd
df = pd.read_csv('H:\\test.txt')
print(df.head())
```

Рисунок 3.4 – Зчитування даних з файлу

	label	document
0	14	convent intern commiss rhine bern convent le ...
1	17	exchang letter commiss intern bureau weight m...
2	38	council decis juli conclus protocol access un...
3	41	addit agreement agreement product clock watch...
4	42	exchang letter india excel mr swaminathan amb...

Рисунок 3.5 – Результат завантаження корпусу текстів

Найбільш важливим етапом етапом обробки даних є попередня обробка або препроцесінг. До нього входить велика кількість функцій. У роботі використано декілька з них.

Ядром препроцесінга є стемінг, реалізований у SnowballStemmer. Стемінг – це процес знаходження основи слова для заданого вихідного слова. Основа слова, отримана в процесі стемінгу, не обов'язково співпадає з морфологічним коренем слова.

Видалення пунктуації проведено з допомогою регулярного виразу regex.sub – це метод, який шукає шаблон у рядку та змінює його на вказаний підрядок. Якщо шаблон не знайдений, рядок залишається незмінним.

Також необхідно було перетворити кожен документ у список слів, видалим непотрібні слова (стоп-слова та слова менше трьох символів, як такі, що не несуть змістовного навантаження).

Використовуваний стемер не використовує коренів слів, а просто відкидає суфікси та закінчення за основними морфологічними правилами мови, основна ідея полягає в тому, що існує обмежена кількість словоутворюючих суфіксів, тому стемінг слова відбувається без використовування будь-яких баз основ. Завдяки такому спрощенню він працює швидко, але з деякою похибкою (рис. 3.6).

```
import re, string
from nltk.stem.snowball import SnowballStemmer
def preprocessing(text):
    text = regex.sub('', text) # удаляем пунктуацию
    text = [token for token in text.split() if len(token) > 3 and token not in fr
ench_stopwords] # Удаляем стоп слова
    text = [stemmer.stem(token) for token in text] # Выполняем стэмминг
    text = [token for token in text if token] # Удаляем пустые токены
    return ' '.join(text)
```

Рисунок 3.6 – Фрагмент коду препроцесінгу

Фрагмент результату роботи алгоритмів препроцесінгу наведений на рис.3.7.

```
df['document'] = df['document'].apply(lambda x: preprocessing(x))

  Preprocessing output:

    label                                           document
 0     14  convent intern commiss rhin bern convent gouve...
 1     17  exchang let commiss intern bureau weight measu...
 2     38  council dec jul conclus protocol access unit a...
 3     41  addit agre agre product clock watch industr eu...
 4     42  exchang let indi excel swaminathan ambassador ...
```

Рисунок 3.7 – Фрагмент результату роботи алгоритмів препроцесінгу

Після попередньої обробки хмара слів значно зменшилась до ≈ 20000 слів. Хмара слів після попередньої обробки наведена на рис. 3.8.

Рисунок 3.8 – Хмара слів після передобробки

3.2 Формування векторів документів за допомогою Word2Vec

Word2Vec – інструмент для аналізу семантики природних мов, заснований на дистрибутивній семантиці, машинному навчанні та

векторному поданні слів.

Робота програми здійснюється наступним чином: Word2Vec приймає великий текстовий корпус в якості вхідних даних і зіставляє кожному слову вектор, видаючи координати слів на виході. Спочатку він генерує словник корпусу, а потім обчислює векторне подання слів, «навчаючись» на вхідних текстах. Векторне подання ґрунтується на контекстної близькості: слова, що зустрічаються в тексті поруч з однаковими словами (а, отже, мають схожий зміст), будуть мати близькі (за косинусною відстанню) вектори. Отримані векторні представлення слів можуть бути використані для обробки природної мови та машинного навчання. CBoW – архітектура, яка передбачає поточне слово, виходячи з навколишнього його контексту. Отримувані на виході векторні представлення слів дозволяють обчислювати «семантичну відстань» між словами. Word2Vec виконує прогнозування на підставі контекстної близькості цих слів. Так як інструмент Word2Vec заснований на навчанні простої нейронної мережі, щоб домогтися його найбільш ефективної роботи, необхідно використовувати великі корпуси для його навчання. Це дозволяє підвищити якість прогнозів.

Для роботи з Word2Vec використано бібліотеку Gensim – це Python бібліотека з відкритим вихідним кодом для обробки природної мови, яка була розроблена і підтримується чеським дослідником обробки природної мови Радімом Шехржреком. Бібліотека Gensim дозволяє розробляти вбудовування слів, навчаючи власні моделі Word2Vec в розширеному корпусі або з використанням CBoW алгоритмів скіп-грам.

Навчання моделі gensim Word2Vec за допомогою нашого корпусу зроблено наступним чином (рис. 3.9).

```
my_model = Word2Vec(text_clean, size=100, window=5, workers=8)
```

Рисунок 3.9 – Навчання моделі gensim

Гіперпараметри цієї моделі:

- size: кількість вимірів вкладення, за замовчуванням складає 100;
- window: максимальна відстань між цільовим словом то словами біля цільового слова. Вікно за замовчуванням – 5;
- workers: кількість розділів під час навчання та робітників за замовчуванням дорівнює 8;
- sg: алгоритм навчання, або CBOW (0), або n-грамм (1). Алгоритм навчання за замовчуванням – CBOW.

Після навчання моделі Word2Vec можна отримати вектори слів безпосередньо з моделі навчання наступним чином (рис. 3.10)

```
print(my_model['nation'])
[ 0.08531986  0.18195157  0.06285619  0.00646402 -0.04072823  0.02371704
  0.22574295 -0.18614407  0.18122077 -0.05087738  0.03467123  0.10516423
 -0.06096142  0.02374216 -0.03932733 -0.02013918 -0.04808024 -0.06764343
 -0.04307321  0.02998317  0.14025466 -0.02837947  0.03217368 -0.00335588
  0.21767552 -0.03216136  0.14414977 -0.07705694  0.02489855  0.0297093
 -0.01127423  0.18226656 -0.00872993 -0.0912379   0.05229177 -0.01584101
  0.06042768  0.16986379  0.04138967  0.19783553  0.00660569 -0.07786646
 -0.01146282 -0.03635168  0.1032733  -0.07183895 -0.08549486  0.15287548
  0.28130612  0.09241089  0.06704859 -0.08978235 -0.09740423 -0.00575989
  0.04227378  0.02308913 -0.15896632  0.0554995  -0.14996408 -0.08178575
 -0.00640133 -0.04214219  0.06618742 -0.09167817  0.1543481  -0.01321609
  0.08548995 -0.10715462  0.06917164 -0.11577259 -0.07484116 -0.03806829
 -0.02785777 -0.04037475 -0.08792593 -0.1643432  -0.06325394  0.05034333
  0.0990499  -0.19239408  0.03057059 -0.14073257  0.01688818  0.04478226
 -0.1424461   0.10103362 -0.06240192  0.13775818]
```

Рисунок 3.10 – Фрагмент отриманих векторів слів

Використовуємо Word2Vec для обчислення вбудованої функції model.most_s Similar (), для отримання набору найбільш схожих моделей для даної моделі на основі евклідової метрики (рис. 3.11).

```
print('\n Most similar for council:')
print(my_model.most_similar('council'))
[('local', 0.5685728788375854), ('stat', 0.5399948358535767), ('relev',
0.5140730738639832), ('memb', 0.49664372205734253), ('mandator',
0.4923013746738434), ('hom', 0.4916461706161499), ('drouart',
0.47473931312561035), ('particip', 0.47267037630081177), ('δημόσια',
0.4641476273536682), ('exist', 0.46074172854423523)]
```

Рисунок 3.11 – Отримання набору схожих моделей на основі евклідової метрики

Для візуалізації векторів, отриманих за допомогою Word2Vec використано бібліотеку matplotlib для роботи з графічним поданням та алгоритм зменшення розмірності t-SNE.

3.3 Зменшення розмірності векторів

Для зменшення розмірності векторів був використаний алгоритм t-SNE – це нелінійний метод зменшення розмірності, який є найбільш відповідним для візуалізації багатовимірних наборів даних. Він широко застосовується в обробці зображень, NLP, геномних даних та обробки мови. Короткий огляд роботи t-SNE:

- алгоритм починає роботу з обчислення ймовірності подібності точок в багатовимірному просторі і обчислення ймовірності подібності точок у відповідному низькорозмірному просторі. Подібність точок розраховується як умовна ймовірність того, що точка A вибере точку B в якості свого сусіда; якщо сусіди будуть обрані пропорційно їх щільності ймовірності при нормальному розподілі з центром в точці A;
- потім мінімізується різниця між цими умовними ймовірностями (або подібностями) в багатовимірному і низькорозмірному просторі для ідеального представлення точок даних в низькорозмірному просторі;
- для вимірювання мінімізації суми різниці умовної ймовірності t-SNE мінімізує суму розбіжності Кульбака-Лейблера спільних точок даних з використанням методу градієнтного спуску.

На рис. 3.12 наведено ілюстрацію графічного представлення векторів слів, отриманих за допомогою Word2Vec.

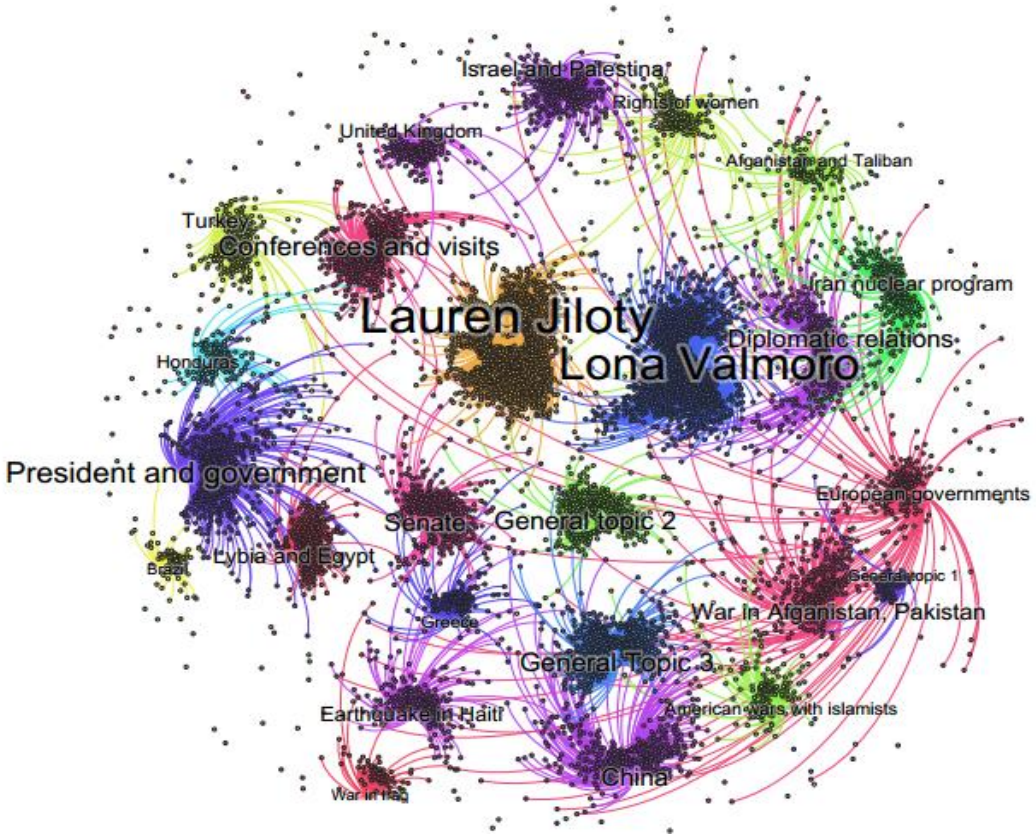

Рисунок 3.12 – Ілюстрація графичного представлення векторів Word2Vec

3.4 Тематичне моделювання

Для тематичного моделювання в роботі використана LDA модель. Це породжуюча модель, яка дозволяє пояснити результати спостережень за допомогою неявних груп, завдяки чому стає можливим виявлення причин схожості деяких частин даних. Наприклад, якщо спостереженнями є слова, зібрані в документи, стверджується, що кожен документ являє собою суміш невеликої кількості тем і що поява кожного слова пов'язана з однією з тем документа. У LDA кожен документ може розглядатися як набір різних тематик.

Двома основними вхідними даними для тематичної моделі LDA є

словник (id2word) і корпус. Приклад створення словника і корпусу наведено на рис. 3.13.

```
from gensim.corpora.dictionary import Dictionary
id2word = corpora.Dictionary(text_clean)
# Create Corpus
texts = text_clean
# Term Document Frequency
corpus = [id2word.doc2bow(text) for text in texts]
#Corpus view
print([[(id2word[id], freq) for id, freq in cp] for cp in corpus[:1]])

[[('aanbevelingen', 2), ('aangegeven', 2), ('aangenomen', 3), ('aangetoond', 1),
('aangevuld', 3), ('aangewezen', 5), ('aannem', 1), ('aantal', 2), ('aanvaard',
1), ('aanvaardbar', 1), ('aanvullend', 8), ('aanvullingen', 1), ('aanwezig', 1),
('aanwijst', 1), ('aanwijz', 2), ('aanzien', 2), ('aanzienlijk', 2), ('aard', 1),
('aardol', 2), ('absenc', 2), ('abstent', 2), ('accept', 5), ('accid', 2),
('accompliss', 1), ('accord', 33), ('account', 5), ('accroiss', 1), ('accroître',
1), ('accumulat', 1), ('achev', 1), ('achiev', 3), ('acht', 2), ('acquis', 2),
('acti', 3), ('action', 5), ('activ', 5), ('adapt', 2), ('addit', 9),
('additionnel', 7), ('administr', 1), ('administratiev', 1), ('admiss', 4),
('adopt', 8), ('adress', 2), ('advanc', 2), ('adver', 1), ('aesthet', 1),
('affect', 4), ('afférent', 1), ('afhangt', 1), ('afin', 1), ('afloop', 2),
('afschrift', 2), ('afvalwat', 2), ('afzonderlijk', 2), ('agiss', 1),
('agrarisch', 1), ('agre', 23), ('agricol', 1), ('agricultur', 1), ('ains', 5),
('aldu', 1), ('ali', 2), ('aliment', 1), ('allemagn', 5), ('allemand', 2),
('allen', 2), ('allow', 1), ('alsmed', 4), ('alsook', 2), ('alter', 1),
('alvoren', 1), ('amend', 4), ('ammoni', 1), ('ammoniak', 1), ('ammoniaqu', 1),
('amélior', 4), ('analy', 2), ('analys', 1), ('ancien', 1), ('ander', 10),
('anim', 1), ('animal', 1), ('annex', 77), ('annual', 2),…]]
```

Рисунок 3.13 – Приклад створення словника і корпусу текстів

Gensim створює унікальний ідентифікатор для кожного слова в документі. Створений корпус, показаний вище, є відображенням (word_id, word_frequency).

Це використовується в якості вхідних даних для моделі LDA.

Тепер є все необхідне для навчання моделі LDA. На додаток до корпусу і словника необхідно також вказати кількість тем.

Крім того, alpha і eta є гіперпараметрами, які впливають на розрідженість тем. Згідно з документацією Gensim, обидва значення за замовчуванням рівні 1.0; num_topics, chu – це кількість документів, які будуть

використовуватися в кожному навчальному чанку; update_every визначає, як часто параметри моделі повинні оновлюватися, а passes – це загальна кількість проходів навчання (рис.3.14).

```
# Build LDA model
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=4,
                                            random_state=100,
                                            update_every=1,
                                            chunksize=100,
                                            passes=10,
                                            alpha=0.2,
                                            per_word_topics=True)
```

Рисунок 3.14 – Приклад задання параметрів Gensim

Модель LDA побудовано на чотирьох різних темах, де кожна тема представлена комбінацією ключових слів, і кожне ключове слово вносить певну вагу в тему.

Ключові слова для кожної теми і вагу (важливість) кожного ключового слова можна переглянути, використовуючи функцію print_topics.

```
print('Topics: \n')
print( lda_model.print_topics())

Topics:
[(0, '0.042*"articl" + 0.030*"regul" + 0.023*"memb" + 0.022*"stat" + 0.020*"quot"
+ 0.017*"european" + 0.017*"commun" + 0.016*"commiss" + 0.014*"council" +
0.011*"annex"'), (1, '0.011*"appropr" + 0.011*"stat" + 0.010*"commiss" +
0.010*"financ" + 0.007*"comm" + 0.007*"payment" + 0.006*"articl" + 0.006*"inform"
+ 0.006*"quot" + 0.006*"measur"'), (2, '0.021*"quot" + 0.010*"cond" +
0.009*"direct" + 0.007*"extract" + 0.007*"acid" + 0.007*"control" + 0.007*"annex"
+ 0.006*"test" + 0.006*"mater" + 0.006*"sampl"'), (3, '0.032*"product" +
0.019*"regul" + 0.018*"import" + 0.015*"export" + 0.015*"market" + 0.015*"commun"
+ 0.014*"quot" + 0.012*"produc" + 0.012*"pric" + 0.008*"articl"')]
```

Рисунок 3.15 – Перегляд ключових слів

Після створення моделі LDA, наступним кроком є вивчення створених тем і пов'язаних з ними ключових слів. Для цього був використаний такий

інструмент, як інтерактивна діаграма пакета pyLDAvis – бібліотека Python для візуалізації інтерактивних тематичних моделей (рис. 3.16). Це порт пакета R від Карсона Сиверта і Кенні Ширлі. Пакет витягує інформацію з тематичної моделі LDA для інтерактивної візуалізації. Візуалізація призначена для використання в IPython, але також може бути збережена в окремий HTML-файл.

```
import pyLDAvis
import pyLDAvis.gensim
import matplotlib.pyplot as plt
vis = pyLDAvis.gensim.prepare(lda_model, corpus, id2word)
pyLDAvis.save_html(vis, 'H:\\LDA_Visualization.html')
```

Рисунок 3.16 – Створення візуалізації

На рис. 3.17 наведено візуалізацію для чотирьох обраних тем. Кожна «бульбашка» на лівому графіку представляє тему. Чим більше «бульбашка», тим більше поширена ця тема. Хороша тематична модель матиме досить великі непересічні «бульбашки», розкидані по всій діаграмі, а не згруповані в одному квадраті. Модель з дуже великою кількістю тем, як правило, має багато перекриттів, «бульбашки» невеликого розміру, згруповані в одній області діаграми.

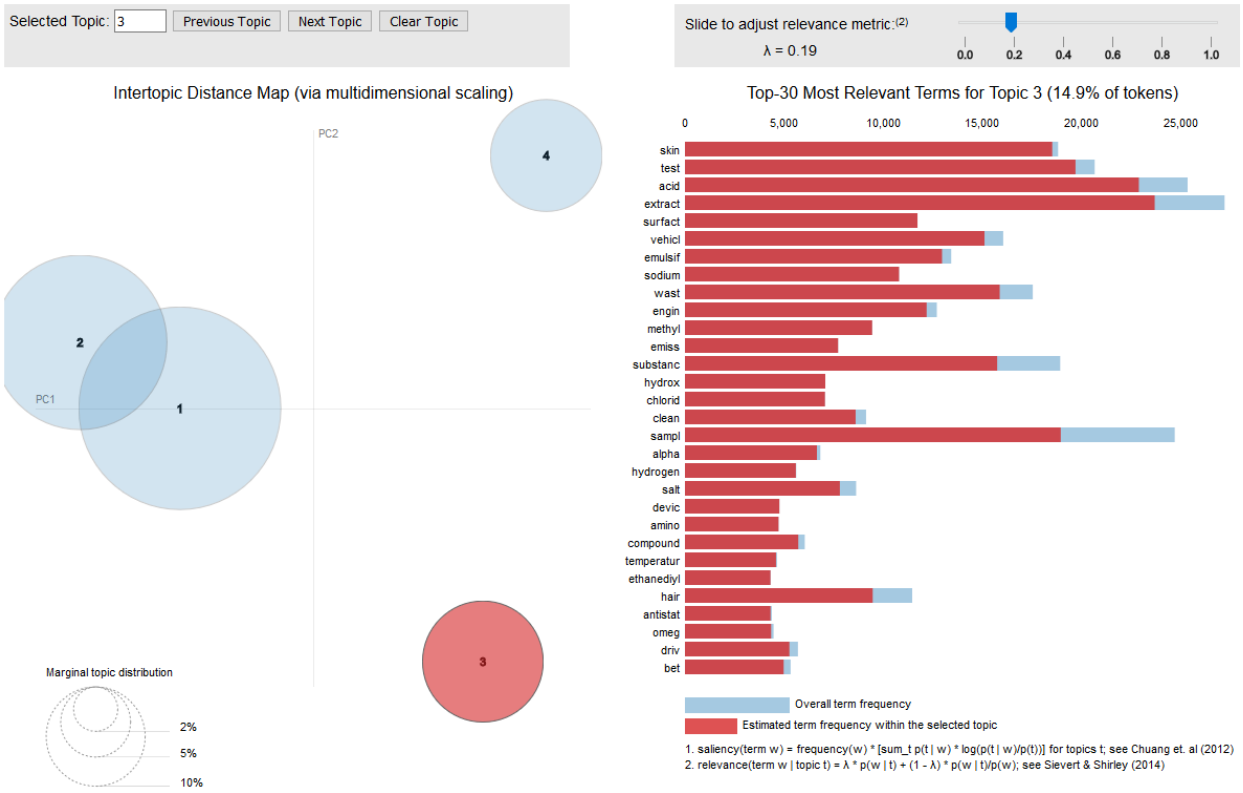

Рисунок 3.17 – Візуалізація pyLDAvis

Тепер, дивлячись на топ ключових слів для кожної теми, стає можливим підібрати назви для них.

Встановивши гіперпараметр λ = 0.2, було отримано топ ключових слів з найбільшою передбачуваною частотою зустрічей щодо кожного окремого топіка.

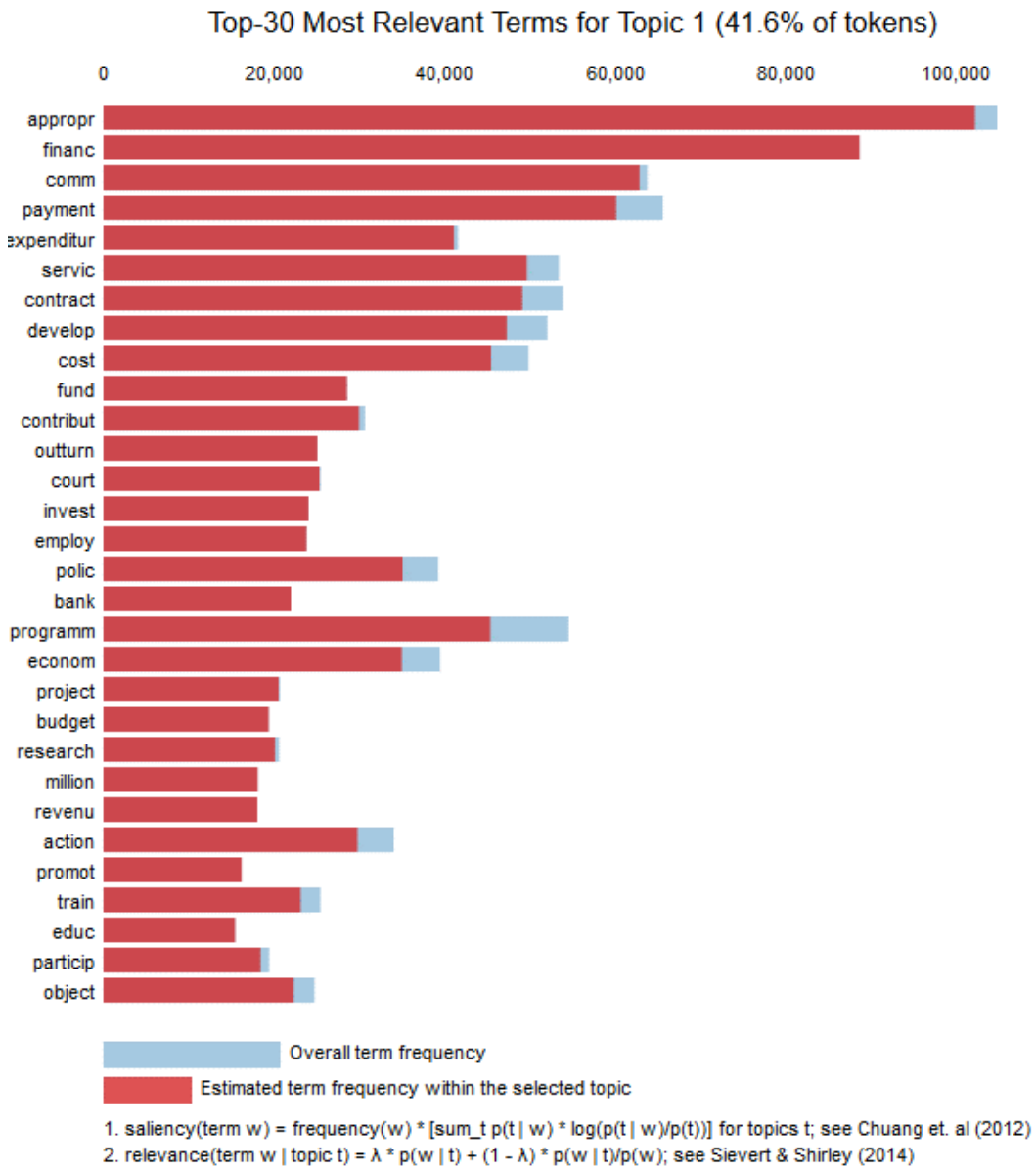

Рисунок 3.18 – Топ ключових слів для першої теми

Виходячи з представленої вище діаграми, можна зробити висновок, що тема стосується фінансових питань, значить назву теми можна визначити як – financial_cooperation.

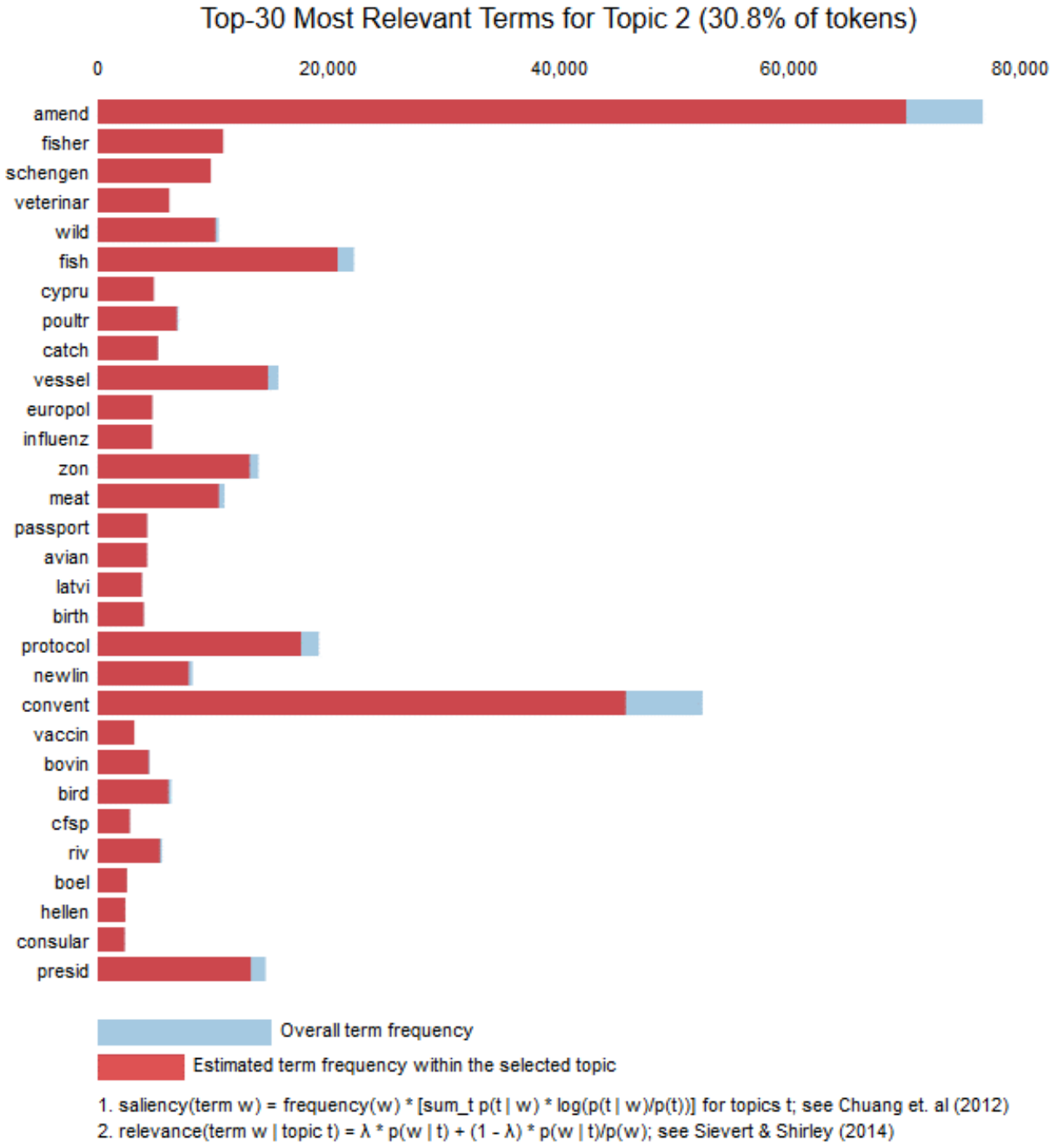

Рисунок 3.19 – Топ ключових слів для другої теми

Виходячи з наведеного вище, можна зробити висновок, що тема пов'язана з тваринами, тому назву теми можна визначити як – protection_of_animals.

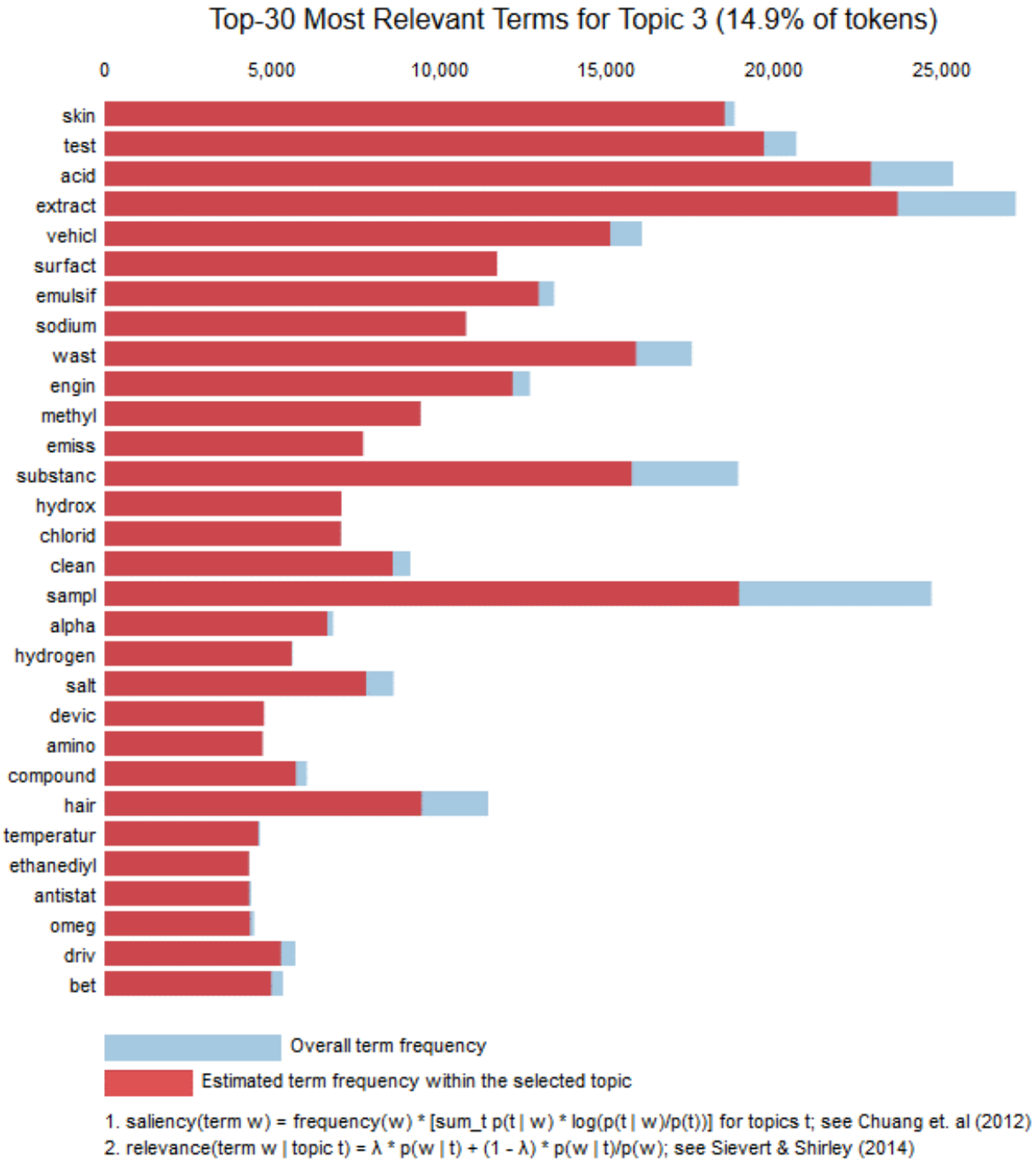

Рисунок 3.20 – Топ ключових слів для третьої теми

Виходячи з наведеної вище діаграми, можна зробити висновок, що тема стосується хімічних речовин і навколишнього середовища, значить назву теми можна визначити як – environmental_protection.

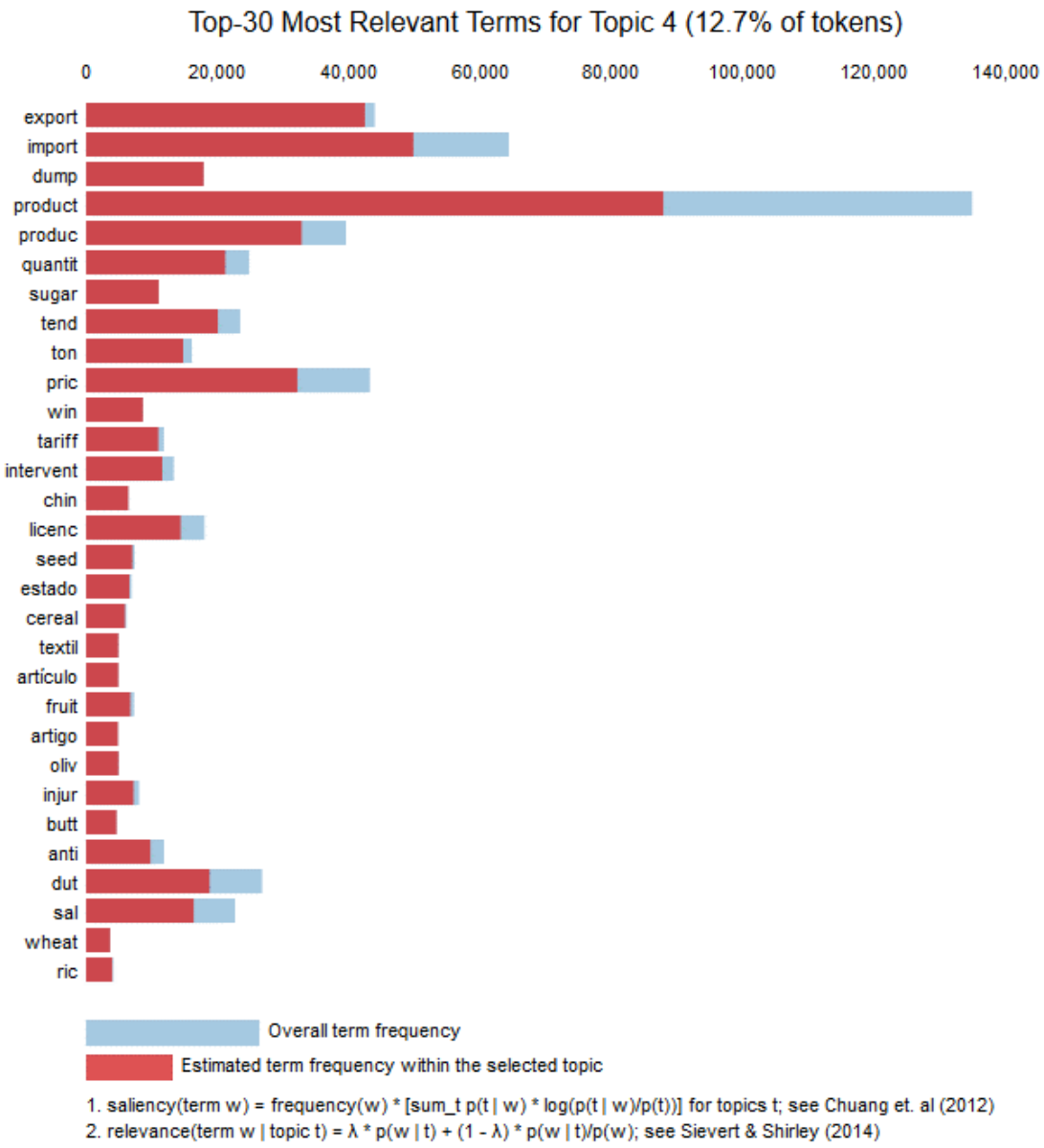

Рисунок 3.21 – Топ ключових слів для четвертої теми

Виходячи з наведеної вище діаграми, можна зробити висновок, що тема пов'язана з торгівлею, значить назву теми можна визначити як – trade_policy.

В результаті експериментального моделювання була проведена попередня обробка обраної текстової колекції засобами стемера SnowballStemmer, видалення пунктуації та стоп-слів, отримані вектори слів за допомогою Word2Vec, проведено тематичне моделювання за допомогою LDA та визначено назви для кожної з чотирьох тем. Це доводить працездатність обраних алгоритмів.

4 ТЕМАТИЧНЕ МОДЕЛЮВАННЯ КОНТЕНТУ САЙТІВ УНІВЕРСИТЕТУ ТА КАФЕДР

Як описано у розділі 1, укладачі міжнародних рейтингів багато уваги приділяють наповненню сайтів університету та всіх сайтів, які знаходяться в домені університету. Але найбільша увага приділяється саме офіційному сайту. Зараз багато уваги приділяється цілям сталого розвитку (ЦСР) та участі університетів у їх досягненні. На цю тему звертають безпосередню увагу такі рейтинги: THE University Impact Rankings, UI Greenmetric World University Rankings, приділяють також увагу QS Word University Rankings та U-Multirank. Більшість потрібної для рейтингування інформації організатори рейтингів беруть саме з офіційних сайтах університетів, а ту інформацію, яку університети надають самостійно, – ретельно перевіряють на тих же сайтах. Тому вкрай важливо розміщувати на сайтах актуальну інформацію, і не менш важливо – структурувати інформацію таким чином, щоб сторонні відвідувачі змогли швидко знайти відповіді на всі свої запити.

Тому в атестаційній роботі акцент зроблено на структурування контенту сайту університету для задоволення інформаційних потреб сторонніх відвідувачів, а саме – організаторів міжнародних рейтингів.

Для цього найпридатнішим інструментом можна вважати тематичне моделювання.

4.1 Підготовка корпусу текстів

Для випробування роботи алгоритмів тематичного моделювання було зібрано тексти сторінок (англійською мовою) офіційного сайту університету та сайтів кафедр як окремі документи. Усім документам були поставлені у відповідність свої мітки, і збережено у окремий текстовий файл. Таким чином отримали файл, подібний до першого файлу колекції EUR-Lex, але набагато меншого розміру.

Була проведена наступна попередня обробка:

- всі слова були конвертовані в нижній регістр;

- були вилучені слова з цифрами і буквено-цифровими символами, винятком стали слова з дефісом («-»), оскільки складові слова пишуться з дефісом;

- був проведений стемінг за допомогою стемера SnowballStemmer, який перетворює слова з однаковою семантикою в їх основу;

- були вилучені стоп-слова, для цього був використаний набір з бібліотеки NLTK. Крім того, були додані кілька стоп-слів самостійно (наприклад, «university», «department», тощо). Завдяки цьому методу видаляються найбільш часто використовувані загальні слова, які дають найменшу кількість інформації або взагалі не містять інформації про документ. Таким чином, це допомагає сфокусуватися на важливих словах.

У зв’язку з тим, що кількість документів, яка відповідає тематикам щодо цілей сталого розвитку доволі мала, довелось використовувати ключові слова, які рекомендовані рейтингом THE University Impact Rankings у якості ключових слів за кожною з ЦСР. Ключові слова доступні за посиланням https://www.scival.com/sdg. На рис. 4.1 – 4.4 наведені хмари слів для ЦСР 3, 4, 5, 10:

- ЦСР 3 – Міцне здоров’я та благополуччя: забезпечення здорового способу життя і сприяння добробуту для всіх в будь-якому віці;

- ЦСР 4 – Якісна освіта: забезпечення всеохоплюючої та справедливої якісної освіти та заохочування можливості навчання протягом усього життя для всіх;

- ЦСР 5 – Гендерна рівність: домагання гендерної рівності та розширення прав і можливостей всіх жінок і дівчаток;

- ЦСР 10 – Зменшення нерівності: зменшення нерівності всередині країн і між ними.

Ці ЦСР обрані для аналізу тому, що кількість документів за цими ЦСР дозволяє проводити процедуру тематичног моделювання. Інші ЦСР, на жаль,

недостатньо представлені на сайті ХНУРЕ.

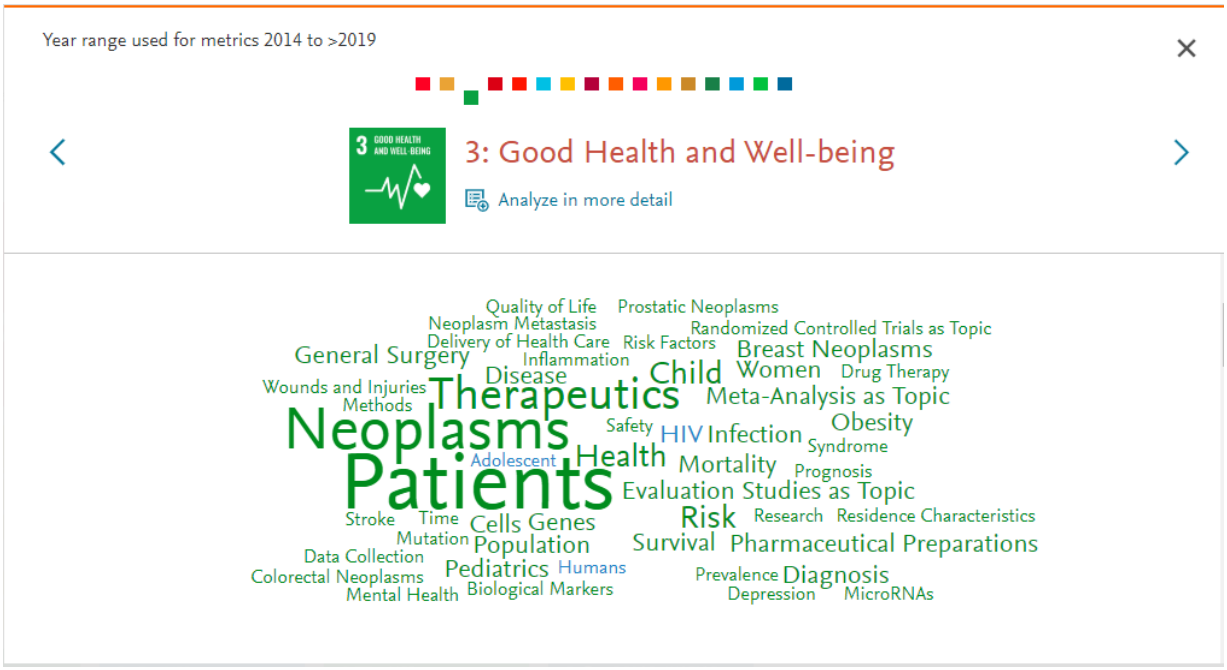

Рисунок 4.1 – Хмара слів з ЦСР 3 – Міцне здоров’я та благополуччя

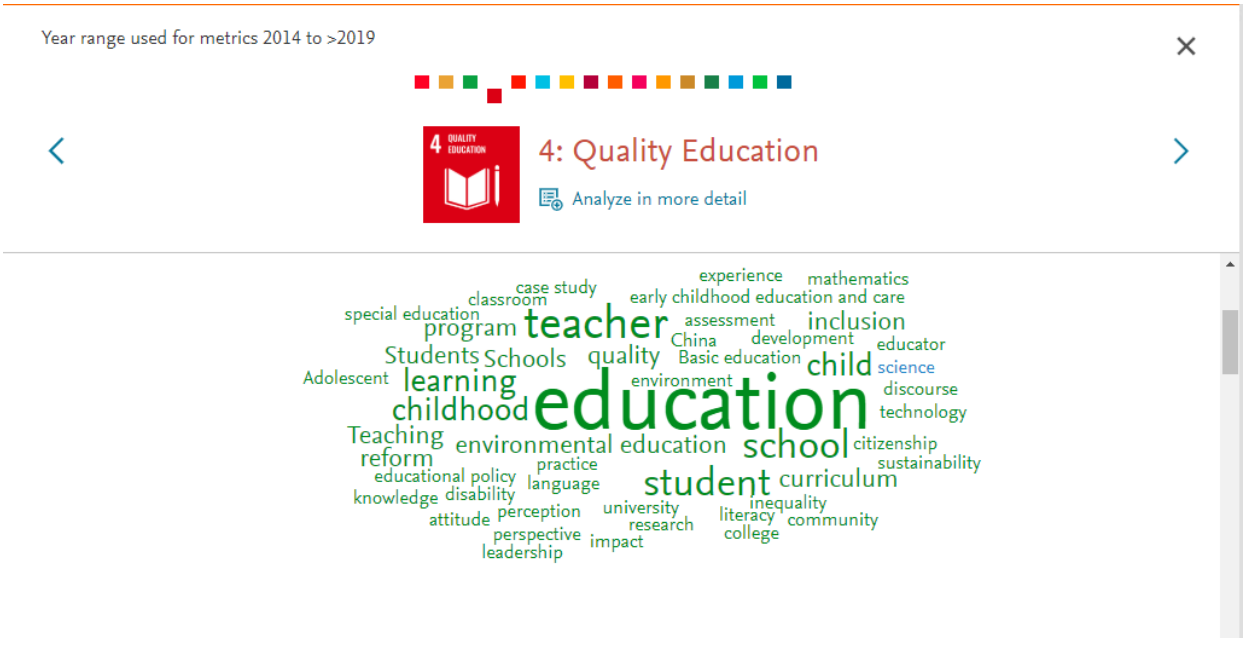

Рисунок 4.2 – Хмара слів з ЦСР 4 – Якісна освіта

Рисунок 4.3 – Хмара слів з ЦСР 5 – Гендерна рівність

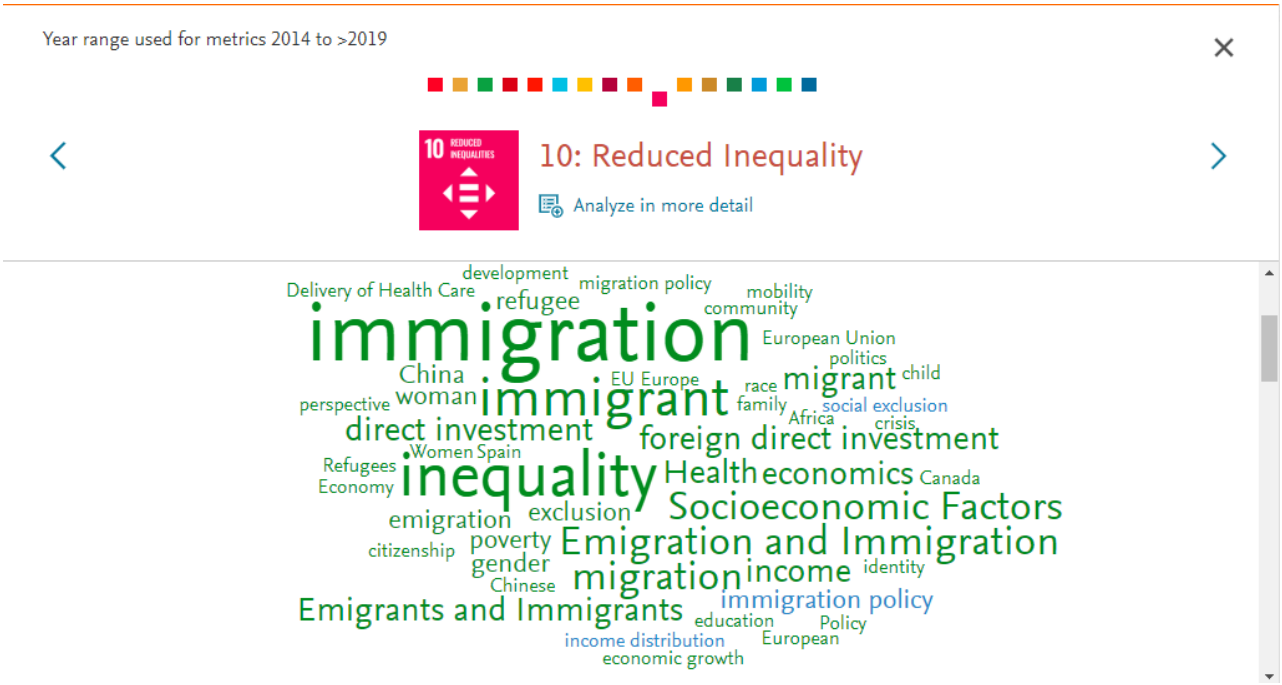

Рисунок 4.4 – Хмара слів з ЦСР 10 – Скорочення нерівності

Векторне подання.

На основі передоброблених даних були отримані вектора, які описують підготовлений набір даних.

CountVectorizer (бібліотека scikit-learn) перетворює вхідний текст в матрицю, значеннями якої є кількості входження даного ключа (слова) в текст.

```
vector = CountVectorizer(max_df=0.8, min_df=2)
vocabulary = vector.fit_transform(daf_text['1'].values.astype('U'))
```

Рисунок 4.5 – Реалізація метода CountVectorizer (бібліотека scikit-learn)

4.2 Використання моделі LDA

Далі переходимо до застосування LDA – яка, як було описано у другому розділі, породжує модель, що дозволяє пояснювати результати спостережень за допомогою неявних груп, завдяки чому можливе виявлення причин подібності деяких частин даних. Якщо спостереженнями є слова, зібрані в документи, стверджується, що кожен документ являє собою суміш невеликої кількості тем і що поява кожного слова пов'язане з однією з тем документа. LDA є одним з методів тематичного моделювання.

```
lda = LatentDirichletAllocation(n_components=4)
lda.fit(vocabulary)

topic_values = LDA.transform(vocabulary)
```

Рисунок 4.6 – Застосування LDA

Для класифікації необхідно підготувати дані, а саме розділити весь датасет на тестову (70%) і навчальну (30%) вибірку. У зв’язку з тим, що

кількість текстів на сайті університету є невеликою, до датасету були додані статті з сайту рейтингу THE Impact Ranking, які відповідають ЦСР 3, 4, 5 та 10.

Для проведення класифікації даних було використано два алгоритми: SGD та SVM.

SGD – стохастичний градієнтний спуск відноситься до оптимізаційних алгоритмів для налаштування параметрів моделі машинного навчання. При стандартному градієнтному спуску для коригування параметрів моделі використовується градієнт. Значення градієнта апроксимуються градієнтом функції вартості, обчисленому тільки на одному елементу навчання, а параметри моделі модифікуються після кожного елемента навчальної вибірки. Даний метод використовується в тому випадку, якщо необхідно застосувати класифікатор на великому наборі даних, так як він є досить швидким.

SVM – метод опорних векторів. Суть даного методу: дві паралельні гіперплощини будуються по обидва боки гіперплощини, яка в свою чергу максимізує відстань до двох паралельних гіперплощ, розділяючи класи. Чим більше відстань між цими паралельними гіперплощинами, тим менше середня помилка класифікатора.

Для оцінювання якості класифікації були обрані метрики ROC AUC і PR AUC. Вони можуть бути розраховані за допомогою бібліотеки scikit-learn в модулі metrics.

Крива ROC – це співвідношення між вірним позитивним значенням і помилковим позитивним значенням, заданим іншим граничним значенням. ROC AUC – це область під кривою ROC. Це метрика, яка використовується для вимірювання того, наскільки добре модель може розрізняти два класи.

Чим краще алгоритм класифікації, тим вище площа під кривою ROC. Оцінка варіюється від 0,5 до 1, і оцінка, яка дорівнює 1, є ідеальним випадком, коли TPR дорівнює 1, а FPR дорівнює 0, що означає, що правильно класифікуються всі позитивні і негативні сторони.

Precision-Recall AUC (PR AUC) аналогічний ROC AUC в тому сенсі, що він підсумовує криву з діапазоном порогових значень у вигляді єдиної оцінки. Precision можна інтерпретувати як частку об'єктів, названих класифікатором позитивними і при цьому дійсно є позитивними, а recall показує, яку частку об'єктів позитивного класу з усіх об'єктів позитивного класу знайшов алгоритм.

Результати класифікації наведені в таблиці 4.1.

Таблиця 4.1 – Результати класифікації (порівняння значень метрик ROC AUC та PR AUC)

	ROC AUC	PR AUC
LSVM_tfidf_vec_1	0.69	0.61
LSVM_tfidf_vec_2	0.56	0.50
LSVM_tfidf_vec_3	0.69	0.63
LSVM_tfidf_vec_4	0.53	0.50
SGD_tfidf_vec_1	0.75	0.61
SGD_tfidf_vec_2	0.58	0.50
SGD_tfidf_vec_3	0.73	0.62
SGD_tfidf_vec_4	0.55	0.51

Результатами виконаного завдання є натреновані моделі SVM і SGD на чотирьох різних векторах ознак для класифікації документів. З таблиці 4.1 видно, що класифікатор SVM справляється із завданням гірше, ніж SGD класифікатор – значення ROC AUC і PR AUC моделі SVM на всіх векторах менше, ніж відповідні значення моделі SGD. Виняток є SVM для 3-го вектора, який має більш велике значення PR AUC із запропонованих. Класифікатор SGD має високі показники ROC AUC і PR AUC для 1-го і 3-го

векторів зі значеннями ngram_range (1, 1) і ngram_range (2, 2), min_df = 0.01, max_df = 0.8.

Можна помітити, що зі збільшенням значення min_df – нижнього порога частоти документа для терміна, ROC AUC і PR AUC значно зменшуються для обох класифікаторів. Чим більше min_df, тим більше термінів відсіюється, що може привести до нестачі термінів, необхідних для досягнення кращих результатів класифікатора.

Так само, можна зробити висновок про те, що SGD класифікатор краще класифікує документи на уніграмах, а SVM – на біграмах.

4.2 Візуалізація результатів

Для візуалізації векторів використовувався пакет pyLDAvis. Пакет витягує інформацію з тематичної моделі LDA для інтерактивної візуалізації.

```
import pyLDAvis
import pyLDAvis.sklearn
pyLDAvis.enable_notebook()
pyLDAvis.sklearn.prepare(LDA, vocabulary, vectorizer=vector)
data = pyLDAvis.sklearn.prepare(LDA, vocabulary, vectorizer= vector)
pyLDAvis.display(data)
```

Рисунок 4.7 – Підключення пакета pyLDAvis для візуалізації

За допомогою інструменту The Classification GUI була проведена візуалізація матриці «слова-теми».

The Classification GUI являє собою графічний інтерфейс, який дозволяє інтуїтивно призначати класи / концепції з онтології для набору документів. Передбачуване використання полягає в тому, щоб легко і швидко створити анотований набір даних, який можна використовувати для експериментів по класифікації. властивість багаторівневих призначень, таким чином можна зіставити документи окремих класів, а не тільки одного.

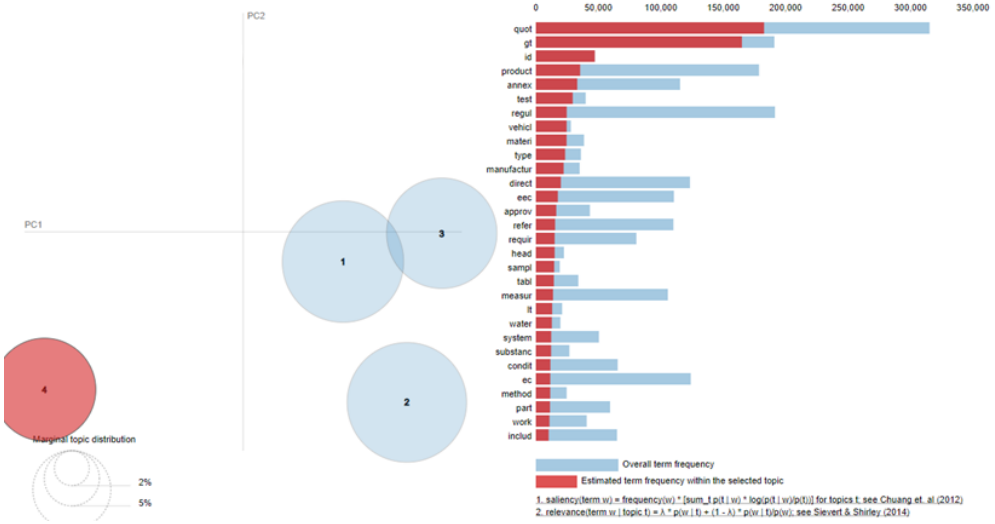

Рисунок 4.8 – Візуалізація вектора для ЦСР 3

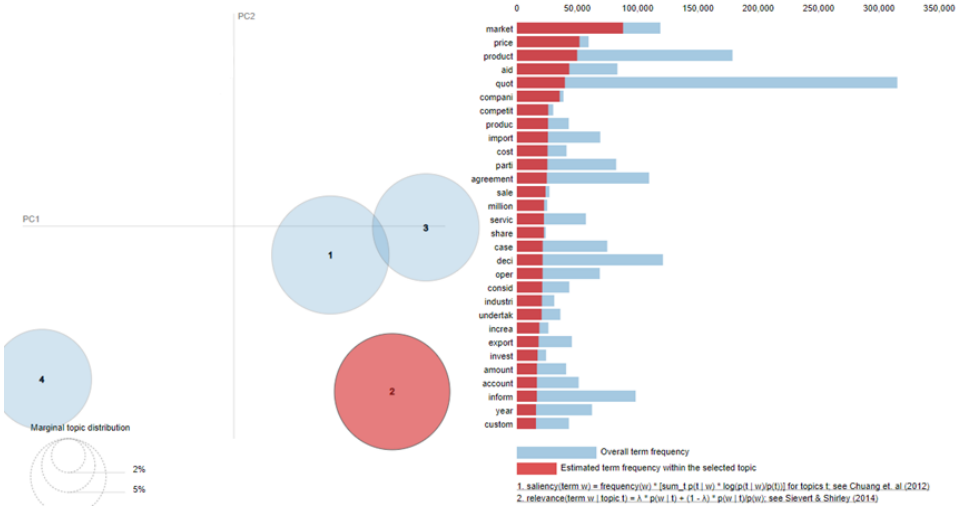

Рисунок 4.9 – Візуалізація вектора для ЦСР 4

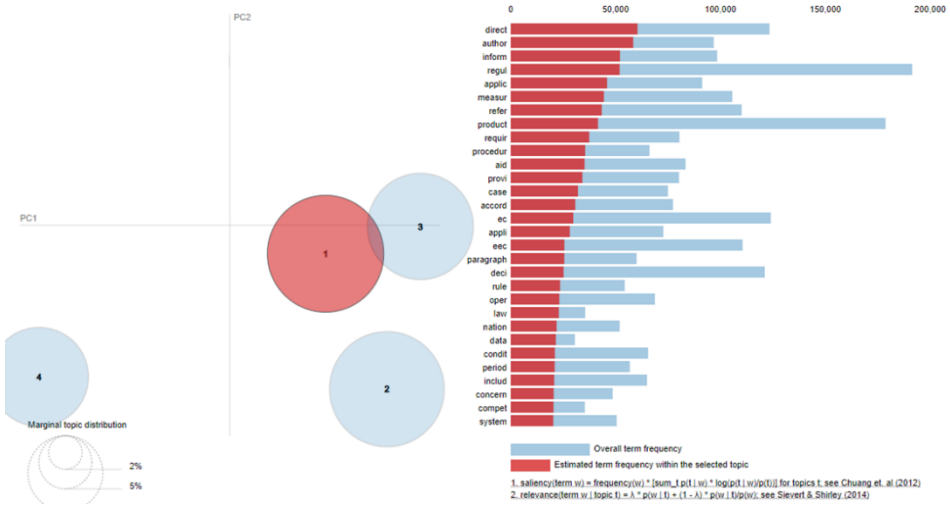

Рисунок 4.10 – Візуалізація вектора для ЦСР 5

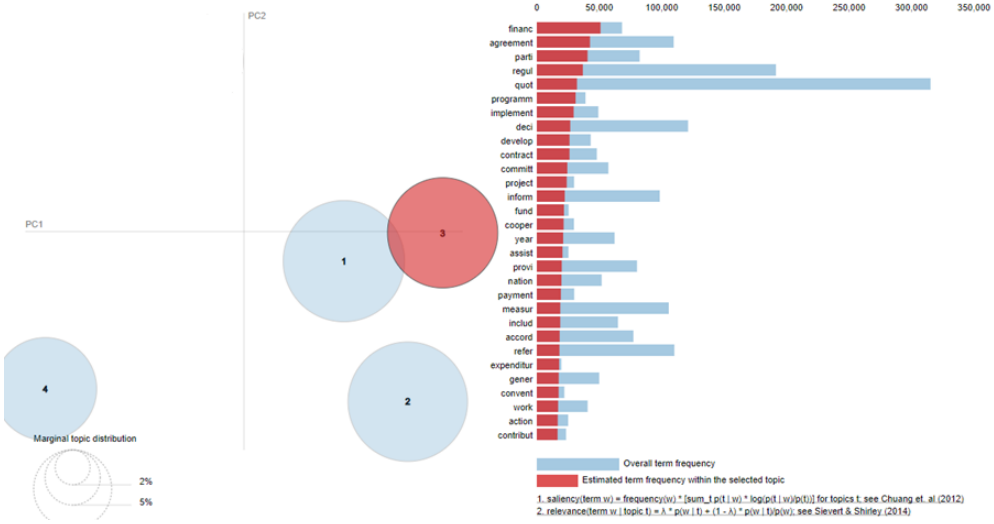

Рисунок 4.11 – Візуалізація вектора для ЦСР 10

Таким чином за допомогою імітаційного модулювання було перевірено роботу моделі LDA і виявлено її працездатність.

Треба зауважити, що модель набагато краще працює на великих корпусах текстів, і у якості термів бажано використовувати біграми.

ВИСНОВКИ

В атестаційній роботі вирішується актуальна задача контент-аналізу веб сайту університету з точки зору тематичної організації. Для вирішення цієї задачі були використані методи тематичного моделювання корпусу текстів.

Метою атестаційної роботи є дослідження методів тематичного моделювання корпусів текстів та створення тематичної моделі на основі контенту сайтів університету та окремих кафедрі.

Згідно з метою в атестаційній роботі були вирішені наступні задачі:

- проведений аналіз існуючих наукових джерел з обробки природномовних текстів;
- проведений аналіз контенту сайтів університету та кафедр з метою виявлення публікацій на тему «Цілі сталого розвитку»;
- зроблено аналіз вимог та критеріїв міжнародних рейтингів щодо наповнення сайтів університетів, зокрема рейтингу THE Impact Ranking;
- досліджені методи та етапи тематичного моделювання, а саме, латентно-семантичний аналіз та латентне розміщення Діріхле;
- розроблено тематичну модель корпусу текстів EurLex;
- розроблено тематичну модель контенту сайту університету згідно з критеріями міжнародних університетських рейтингів.

Під час виконання роботи були докладно проаналізовані наукові джерела з обраної тематики, а саме – математичні методи обробки природномовних текстів: латентно-семантичний аналіз, латентний розподіл Діріхле, методи редукції для уникнення низької обумовленості матриць. Під час моделювання застосовані різні алгоритми та бібліотеки мови Python, яка найкращим чином пристосована для обробки текстових даних.

Імітаційне моделювання довело, що кращі результати можна отримати, якщо застосовувати попередню обробку текста (видалення пунктуації, стоп-слів, стемінг); використовувати векторне подання документів замість моделі «мішок слів»; використовувати біграми замість унірамів.

Звичайно, кращі результати отримують при великому обсязі корпусу текстів. Тому в атестаційній роботі документи з сайту університету були доповнені релевантними статтями з різних джерел за схожою тематикою.

В роботі використовувались тексти англійською мовою задля спрощення роботи укладачам міжнародних університетських рейтингів – вони беруть інформацію з англомовних сторінок сайтів університетів.

Таким чином, результати атестаційної роботи можна використовувати для тематичного структурування контенту сайту університету, що дозволить підвищити рейтингові показники ХНУРЕ.

ПЕРЕЛІК ПОСИЛАНЬ

1. Chomsky N. Syntactic Structures. The Hague: Mouton, 1957.

2. Виноград Т. Программа, понимающая естественный язык. М.: Мир, 1976.

3. Воронцов К. В. Вероятностное тематическое моделирование: теория, модели и проект BigARTM. МФТИ, 2019. 95с.

4. Zhang J., Song Y., Zhang C., Liu S. Evolutionary hierarchical Dirichlet processes formul tiple correlated time-varying corpora. *KDD'10*: Proceeding soft he 16th ACM SIGKDD international conference on Knowledge discovery and data mining. 2010. P. 1079–1088.

5. Text Flow: Towards better understanding of evolving topics in text / W. Cui et. al. *IEEE transactions on visualization and computer graphics*. 2011. Vol. 17, No. 12. P. 2412–2421.

6. Vuli´c I., Smet W., Moens M.-F. Cross-language information retrieval models based on latent topic models trained with document-aligned comparable corpora. *Information Retrieval*. 2012. P. 1–38.

7. Vulic I., Smet W., Tang J., Moens M.-F. Probabilistic topic modeling in multi lingual settings: an overview of its methodology and applications. *Information Processing & Management*. 2015. Vol. 51, No. 1. P. 111–147.

8. Originator or propagator?: Incorporating social role theory into topic models for Twitter content analysis / X. W. Zhao et. al. *CIKM'13*: Proceeding soft the 22Nd ACM International Conference on Conference on Information and Knowledge Management, NewYork. NY, USA: ACM, 2013. P. 1649–1654.

9. Varshney D., Kumar S., Gupta V. Modeling information diffusion in social networks using latent topic information. Intelligen Computing Theory / ed. D.-S. Huang, V. Bevilacqua, P. Premaratne. *Springer International Publishing*. 2014. Vol. 8588 of Lecture Notes in Computer Science. P. 137–148.

10. Pinto J. C. L., Chahed T. Modeling multi-topic information diffusion in social networks using latent Dirichlet allocation and Hawkes processes. *SITIS'14*:

Tenth International Conference on Signal-Image Technology & Internet-Based Systems. 2014. P. 339–346.

11. Bodrunova S., Koltsov S., Koltsova O., Nikolenko S. I., Shimorina A. Interval semisupervised LDA: Classifying need lesin a haystack / ed. F. C. Espinoza, A. F. Gelbukh, M. Gonzalez-Mendoza. *Springer International Publishing.* 2013. Vol. 8265 of Lecture Notes in Computer Science. P. 265–274.

12. Rubin T. N., Chambers A., Smyth P., Steyvers M. Statistical topic models formultilabel document classification. *Machine Learning*. 2012. Vol. 88, No. 1-2. P. 157–208.

13. Zhou S., Li K., Liu Y. Text categorization based on topic model. *International Journal of Computational Intelligence Systems*. 2009. Vol. 2, No. 4. P. 398–409.

14. Wang H., Zhang D., Zhai C. Structural topic model for latent topical structure analysis. *HLT'11*: Proceeding soft the 49th Annual Meeting of the Association for Computational Linguistics. Vol. 1. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011. P. 1526–1535.

15. Hofmann T. Probabilistic latent semantic indexing. *SIGIR'99*: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, USA: ACM, 1999. P. 50 –57.

16. Hospedales T., Gong S., Xiang T. Video behaviour mining using a dynamic topic model. *International Journal of Computer Vision*. 2012. Vol. 98, No. 3. P. 303–323.

17. Krestel R., Fankhauser P., Nejdl W. Latent Dirichlet allocation for tag recommendation. *RecSys'09*: Proceeding soft the third ACM conference on Recommender systems. ACM, 2009. P. 61–68.

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

19. Lee S. S., Chung T., McLeod D. Dynamic item recommendation by topic modeling for social networks. *Information Technology: New Generations:* Eighth International Conference on. IEEE, 2011. P. 884–889.

20. Yeh J.-h., Wu M.-l. Recommendation based on latent topics and social network analysis. Proceeding soft the 2010 Second International Conference on Computer Engineering and Applications. Vol. 1. IEEE Computer Society, 2010. P. 209–213.

21. Daud A., Li J., Zhou L., Muhammad F. Knowledge discovery through directed probabilistic topic models: a survey. *Frontiers of Computer Science in China*. 2010. Vol. 4, No. 2. P. 280–301.

22. Blei D. M. Probabilistic topic models. *Communication soft the ACM*. 2012. Vol. 55, No. 4. P. 77–84.

23. Розенфельд Л., Морвиль П. Информационная архитектура в Интернет. Изд. 2-е. СПб: Символ-Плюс, 2005. 544 с.

24. Marchionini G. Exploratory search: From finding to understanding. *Communication of the ACM*. 2006. Vol. 49, No. 4. P. 41–46.

25. White R. W., Roth R. A. Exploratory Search: Beyond the Query-Response Paradigm. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan and Claypool Publishers, 2009.

26. Маннинг К.Д., Рахгаван П., Шютце Х. Введение в информационный поиск. Вильямс, 2011.

27. Asuncion A., Welling M., Smyth P., Teh Y. W. On smooth in gandinference for topic models. Proceeding soft the International Conference on Uncertainty in Artificial Intelligence. 2009. P. 27–34.

28. Balikas G., Amini M., Clausel M. On a topic model for sentences. *SIGIR'16*: Proceeding soft he 39th International ACM SIGIR Conference on Research and Development in Information Retrieval. NewYork, NY, USA: ACM, 2016. P. 921–924.

29. Аналіз цілей сталого розвитку в університеті ХНУРЕ (The University Impact Rankings). URL: https://nure.ua/branch/viddil-benchmarkingu-ta-veb-

menedzhmentu/mizhnarodni-rejtingi/the-university-impact-rankings (дата звернення: 15.04.2020).

30. Аналіз бенчмаркінгу в університеті ХНУРЕ (QS World University Rankings). URL: https://nure.ua/branch/viddil-benchmarkingu-ta-veb-menedzhmentu/mizhnarodni-rejtingi/qs-word-university-rankings (дата звернення: 15.03.2020).

31. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие / Е. И. Большакова и др. М.: МИЭМ, 2011. 272 с.

32. Dempster A. P., Laird N. M., Rubin D. B. Maximum likelihood from in complete data viathe EM-algorithm. *J. of the Royal Statistical Society. Series B.* 1977. No. 34. P. 1-38.

33. Тихонов А.Н., Арсенин В.Я. Методы решения некорректных задач. М.: Наука, 1986.

34. Воронцов К.В. Аддитивная регуляризация тематических моделей коллекций текстовых документов. *Доклады РАН*. 2014. Т.456, №3. С. 268-271.

35. Indexing by Latent Semantic Analysis / S. Deerwester et. al. *Journal of the American Society for Information Science*. 1990. Vol. 41, No. 6. P. 391–407.

36. Landauer T., Dumais S. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *JPsychological Review*. 1997. Vol. 104. P. 211–240.

37. Lemaire B., Denhière G. Cognitive Models based on Latent Semantic Analysis. *ICCM'2003*: Tutorial givenat the 5th International Conference on Cognitive Modeling, Bamberg, Germany, 9 April 2003. 2003.

38. Landauer T. K., Foltz P. W., Laham D. Introduction to Latent Semantic Analysis. *Discourse Processes*. 1998. No. 25. P. 259–284.

39. Приклад роботи методу тематичного моделювання. URL: http://topicmodels.west.uni-koblenz.de/ckling/tmt/svd_ap.html (дата звернення:29.03.2020).

40. Blei D. M., Ng A. Y., Jordan M. L. Latent Dirichlet allocation. *Journal of Machine Learning Research*. 2003. Vol. 3. P. 993–1022.

41. Lu Y., Mei Q., Zhai C. Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA. *Information Retrieval*. 2011. Vol. 14, No. 2. P. 178–203.

42. Potapenko A. A., Vorontsov K. V. Robust PLSA performs better than LDA. *ECIR-2013*: 35th European Conference on Information Retrieval, Moscow, Russia, 24-27 March 2013. Lecture Notes in Computer Science (LNCS), SpringerVerlag-Germany, 2013.P. 784–787.

43. Asuncion A., Welling M., Smyth P., Teh Y. W. On smoothing and inference for topic models. Proceeding soft the International Conference on Uncertainty in Artificial Intelligence. 2009. P. 27–34.

44. Girolami M., Kaban A. On an equivalence between PLSI and LDA. *SIGIR'03*: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval. 2003. P. 433–434.