МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри КНІ
_____ Скарга-Бандурова І.С.
«____»_____ 20__ р.

ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА

ПОЯСНЮВАЛЬНА ЗАПИСКА

НА ТЕМУ:

Засоби аналізу авторства тексту по частоті появи нових слів

Освітньо-кваліфікаційний рівень "бакалавр"
Напрям 122 – "комп'ютерні науки"

Керівник проекту: _____ (підпис) Скарга-Бандурова І.С. (ініціали, прізвище)

Консультант з охорони праці: _____ (підпис) Критська Я.О. (ініціали, прізвище)

Студент: _____ (підпис) Михайлюченко О.І. (ініціали, прізвище)

Група: КН-14д

Сєвєродонецьк 2018

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки

Кафедра Комп'ютерних наук та інженерії

Освітньо-кваліфікаційний рівень Бакалавр

Напрям підготовки 6.050101 – "комп'ютерні науки"
(шифр і назва)

Спеціальність _____
(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри _____
_____ І.С. Скарга-Бандурова
«_____» _____20____р.

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА

Михайлюченку Олексію Івановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Засоби аналізу авторства тексту по частоті появи нових слів

керівник проекту (роботи) д.т.н., доц. Скарга-Бандурова І.С.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від " " 201_ р. № ____

2. Термін подання студентом роботи

3. Вихідні дані до роботи матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз та постановка задачі; огляд існуючих засобів розробки додатків, реалізація додатку.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Комп'ютерна презентація

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	ст.викл. Критська Я.О.		

7. Дата видачі завдання _____

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання, збір матеріалів	14.05.2018 – 17.05.2018	
2	Огляд літератури й обгрунтування необхідності розробки	17.05.2018 – 20.05.2018	
3	Розробка технічного завдання	20.05.2018 – 22.05.2018	
4	Проектування структури додатка	22.05.2018 – 24.05.2018	
5	Розробка додатку на ОС Windows	24.05.2018 – 30.05.2018	
6	Інформаційне наповнення додатку	30.05.2018 – 03.06.2018	
7	Перевірка реалізованого фунціоналу	03.06.2018 – 05.06.2018	
8	Охорона праці та безпека в надзвичайних ситуаціях	05.06.2018 – 07.06.2018	
9	Оформлення пояснювальної записки	07.06.2018 – 09.06.2018	

Студент _____ Михайлюченко О.І.
(підпис) (прізвище та ініціали)

Керівник _____ Скарга-Бандурова І.С.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту (роботи) бакалавра: 70с., 9 рис., 6 табл., 24 бібліографічних джерел посилань, 2 додатка.

Об'єкт розробки: Додаток для ОС Windows програма з використанням мови Python.

Мета роботи: Розробка додатку для ОС Windows з використанням мови Python.

В проекті виконано:

1. У розділі «Аналіз засобів розробки додатків для ОС Windows з використанням мови Python» було виконано зрівняння аналогічних додатків, був проведений огляд предметної області, були поставлені задачі щодо розробки системи.

2. У розділі «Моделі апроксимації статистичних даних.» було розглянуто апроксимацію статистичних даних для об'єктів, які підпорядкуватися Законами Зіпфа та визначення її якості.

3. У розділі «Проектування додатку для визначення авторства текстів з використанням мови Python» була описана функціональність, структура та описані алгоритми к додатку.

4. У розділі «Охорона праці та безпека в надзвичайних ситуаціях» був проведений аналіз небезпечних виробничих факторів. І на основі цього аналізу були запропоновані заходи усунення цих факторів

Отримано наступні результати: додаток аналізатор авторства тексту по частоті появи нових слів для ОС Windows.

Ключові слова: додаток, текст, Python, авторство, апроксимація.

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А,. м. Сєвєродонецьк, 93400.

ЗМІСТ

ВСТУП

Сучасне інформаційне суспільство використовує обчислювальні машини різного роду практично у всіх сферах життєдіяльності і, перш за все, в наукових дослідженнях.

У своєму сучасному втіленні комп'ютери і супутні їм інформаційні системи являють собою ідеальне технічне рішення для задач обробки великих обсягів статистичних даних і вирішення складних обчислювальних задач, необхідних, зокрема, в освітньому процесі, лінгвістичних і криміналістичних дослідженнях для ідентифікації авторства тексту по частоті появи нових слів, а також для вирішення більш загальної задачі – класифікації текстової інформації.

Однією із сфер застосування методик є сфера освіти. Школярі та студенти з появою глобальної мережі Інтернет все рідше самі виконують реферати, курсові та доповіді, вважаючи за краще не витрачати на це час і просто завантажити готові роботи з мережі Інтернет. Використання підходів для визначення авторства в цьому випадку дозволить більш об'єктивно оцінювати учнів.

Метою дипломної роботи є зробити аналіз методів ідентифікації авторства текстів, що забезпечує підвищення точності визначення частоту появи нових слів, зменшення обсягу текстової вибірки і зниження тимчасових витрат на прийняття рішення, і створення програмного комплексу для ідентифікації авторства на її основі.

1 АНАЛІЗ ЗАСОБІВ РОЗРОБКИ ДОДАТКІВ ДЛЯ ОС WINDOWS З ВИКОРИСТАННЯМ МОВИ PYTHON

1.1 Огляд предметної області

Питання встановлення автора тексту і часу його написання виникає в різних областях і становить інтерес для філологів, літературознавців, юристів, криміналістів, істориків.

Серед широко відомих прикладів спірного авторства можна назвати активно обговорювався роман-епопею «Тихий Дон».

Довгий час для вирішення подібних питань використовувалися історико-документальні та філологічні методи дослідження. Для виявлення авторських особливостей застосовувалася методика, відповідно до якої суб'єктивно відбиралися зовнішні деталі авторського стилю (улюблені слова, терміни, вирази). Відзначимо, що такі дослідження трудомісткі, тому постає питання про створення формальних методів вирішення даного завдання. В даний час для встановлення авторства і датування текстів застосовуэться безліч підходів з теорії розпізнавання образів, математичної статистики та теорії ймовірностей. Уже в 60-70-і роки. XX ст. основне місце серед методик атрибуції текстів зайняли методи статистики.

Існують програмні продукти, що дозволяють враховувати різні лінгвостатистичні параметри, різнобічно характеризують текст. Але більшість з них досить ресурсовитратні. Були розроблені методики, що дозволяють з достатньою точністю визначати авторство тексту, ластерний на його різних характеристиках.

1.2 Методи аналізу авторства тексту

Шведські вчені розробили новий спосіб виявлення автора тексту – вони встановили, що роль «відбитків пальців» письменників може грати частота

народження нових слів в їх текстах. У статті [1], опублікованій в науковому виданні New Journal of Physics, група шведських фізиків з університету Умео під керівництвом Себастьяна Бернгардсона описала новий метод, що дозволяє на основі статистичних даних визначити автора тексту.

Дослідники перевіряли, як в текстах трьох письменників – Томаса Харді, Генрі Мелвілла і Девіда Лоуренса – реалізується так званий закон Ціпфа [2]. Цей закон, відкритий в 1935 році лінгвіст Джорджем Ципфом, говорить, що частота якого або слова в тексту обернено пропорційна його рангу – місця в списку слів тексту, відсортованих за частотою.

Так, наприклад, друге за частотою слово буде зустрічатися в тексті приблизно в два рази рідше, ніж перше, третє – в три рази рідше і так далі.

Шведські фізики в своїй статті показали, що цей закон не так універсальний, як вважав Ціпфа. Вони виявили, що частота появи нових слів у міру зростання обсягу тексту змінюється в різних авторів по-різному, причому ця закономірність НЕ залежить від конкретного тексту, а тільки від автора.

Статистичний аналіз показав, що закономірність залишається постійною в будь-яких текстах одного і того ж автора – романах, главах з романів, оповідань, і може служить своєрідним «відбитками пальців».

Автори дослідження, спостерігаючи за Цими статистичними закономірностями, висунули ідею так зване «ластерні» – уявного нескінченного тексту, в якому описаний світ очима того чи іншого автора. «Створюючи твір, автор» витягує «шматки тексту з цієї великої» материнської книги «і перекладає їх на папір, зберігаючи, однак, частотні характеристики концептів в цій ластерні», - пишуть вчені.

Статистичні методи визначення авторства відомі досить давно, і шведські дослідники просто запропонували ще один, відносно простий спосіб, сказала лінгвіст Єлизавета Билініной з Утрехтського університету (Нідерланди).

«Традиційно вважається, що авторський стиль добре характеризується розподілом службових слів, і іншими незначними, і через погано помітним оку, а добре помітним статистикою характеристиками – середня довжина пропозиції, кількість вступних слів», - сказала співрозмовниця агентства.

За її словами, Бернгардсон і його співавтори спробували представить деяку частотну «карту» творів, і знайшли, що вона постійна для кожного учасника, навіть без всяких других «хитрощів».

1.3 Закони Зіпфа

Перший закон Зіпфа «ранг – частота». Вибирається будь-яке слово і підраховується, скільки разів воно зустрічається в тексті. Ця величина називається частота входження слова. Вимірюється частота кожного слова тексту. Деякі слова будуть мати однакову частоту, тобто входити в текст рівну кількість разів. Згрупуємо їх, взявши тільки одне значення з кожної групи. Розташуємо частоти в міру їх змен шення та пронумеруємо. Порядковий номер частоти називається ранг частоти. Так, найбільш часто зустрічаються слова матимуть ранг 1, наступні за ними – 2 і т.д.

Імовірність зустріти слово шляхом випадкового вибору, буде дорівнювати відношенню частоти входження цього слова до загальної кількості слів у тексті.

$$P = \frac{\delta}{n},$$

де P – імовірність, δ - частота входження слова, n - число слів.

Зіпф виявив цікаву закономірність. Виявляється, якщо помножити ймовірність виявлення слова в тексті на ранг частоти, то отримана величина (C) приблизно постійна:

$$C = \frac{\delta \cdot R}{n},$$

де R - ранг частоти.

Якщо трохи перетворити формулу, то можна побачити, що це функція **y = k / x** і її графік – рівнобічна гіпербола. Отже, за першим законом Зіпфа, якщо найпоширеніше слово зустрічається в тексті, наприклад, 100 раз, то наступне за частотою слово навряд чи зустрінеться 99 разів. Частота входження другого за популярністю слова, з високою часткою ймовірності, виявиться на рівні 50.

Значення константи в різних мовах по-різному, але всередині однієї мовної групи залишається незмінно, який би текст ми не взяли. Так, наприклад, для англійських текстів константа Зіпфа дорівнює приблизно 0,1. Російські тексти з точки зору законів Зіпфа не виняток. Для російської мови коефіцієнт Зіпфа вийшов рівним 0,06-0,07.

Другий закон Зіпфа «кількість – частота». Розглядаючи перший закон, факту, що різні слова входять в текст з однаковою частотою не розглядався. Зіпф встановив, що частота і кількість слів, що входять в текст з цією частотою, теж пов’язані між собою.

Якщо побудувати графік, відклавши по одній осі (осі X) частоту входження слова, а по інший (осі Y) – кількість слів у даній частоті, то вийшла крива буде зберігати свої параметри для всіх без винятку створених людиною текстів! Як і в попередньому випадку, це твердження вірне в межах однієї мови. Однак і міжмовні відмінності невеликі. На якій би мові текст не був написаний, форма кривої Зіпфа залишиться незмінною. Можуть трохи відрізнятися лише коефіцієнти, що відповідають за нахил кривої (в логарифмічному масштабі, за винятком декількох початкових точок, графік – пряма лінія).

Закони Зіпфа універсальні. Вони можуть бути застосовані не тільки до текстів. Характеристики популярності вузлів в мережі Інтернет – теж відповідають законам Зіпфа. Не виключено, що в законах відбивається «людське» походження об’єкта.

Якщо скористатися першим законом Зіпфа і побудувати графік залежності рангу від частоти, то дослідження показують, що найбільш значущі слова лежать в середній частині діаграми. Слова, які трапляються дуже часто, в основному виявляються приводами, займенниками, в англійській – артикля і т.п. Рідко зустрічаються слова теж, в більшості випадків, не мають вирішального значення інформації.

Закони Зіпфа універсальні. В принципі, вони можуть бути застосовані не тільки до текстів. Характеристики популярності вузлів в мережі Інтернет – теж відповідають законам Зіпфа. Не виключено, що в законах відбивається «людське» походження об'єкта.

Що дають закони Зіпфа? Як з їх допомогою отримати слова, що відображають зміст тексту? Якщо скористатися першим законом Зіпфа і побудувати графік залежності рангу від частоти, то дослідження показують, що найбільш значущі слова лежать в середній частині діаграми. Слова, які трапляються дуже часто, в основному виявляються приводами, займенниками, в англійській – артикля і т.п. Рідко зустрічаються слова теж, в більшості випадків, не мають вирішального значення інформації.

Формула другого Закону Зіпфа

Зіпф встановив, що частота і кількість слів, що входять в текст з цією частотою, теж пов'язані між собою [3]. Спеціальна формула:

$$\sum_{i=1}^{n}(a_1 - a_3 + \frac{b_1 - b_3}{x_i} + \Delta_1 - \Delta_3) > \sum_{i=1}^{n}(a_1 - a_2 + \frac{b_1 - b_2}{x_i} + \Delta_1 - \Delta_2),$$

$$x_1 \le x \le x_2$$

де $a_1, a_2, b_1, b_2, \Delta_1, \Delta_2$ — коефіцієнти гіперболічної апроксимації і похибки апроксимації для автора К;

$a_3, b_3 \Delta_3$ — коефіцієнти гіперболічної апроксимації і похибка апроксимації для автора М;

$x_1 \le x \le x_2$ — діапазон для кількості слів настроюється в процесі аналізу графіка.

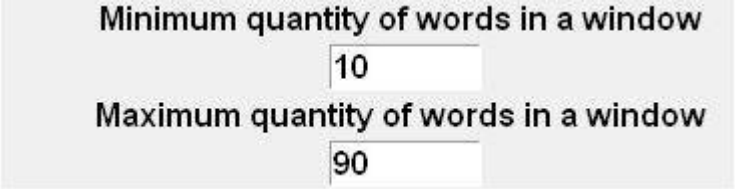


Рисунок 1.1 – Кількість слів від 10 до 90 за другим Закону Зіпфа

1.4 Метод і міра Хмельова

У травні 1999 р. на сайті російської фантастики з'явилася стаття Д. Хмелева, в якій був запропонований новий метод встановлення авторства текстів. Даний метод дозволяв з високою точністю класифікувати тексти по авторству на основі порівняння чисел появи ластерних букв. Незважаючи на успішну реалізацію методу і велику популярність, детальних досліджень в цьому напрямку практично не проводилося.

Ключовою ідеєю методу Хмелева є підрахунок обробка послідовностей елементів тексту. Розпізнаванню передує навчання системи. Навчання виробляється на текстах заданої множини авторів. Для кожного учасника підраховується матриця-еталон вживань всіх пар розглянутих елементів в його текстах. При розпізнаванні авторства довільного тексту підраховується аналогічна матриця і порівнюється з усіма наявними матрицями-еталонами. Автор, який володіє найбільш схожою матрицею-еталоном, буде приблизно автором даного тексту.

В якості запобіжного порівняння використовується величина:

$$L = -\sum_{i=1}^{k}\sum_{j=1}^{k} m_{1ij} \cdot \ln\left(\frac{m_{1ij}}{n_{1ij}} / \frac{m_{2ij}}{n_{2ij}}\right) \qquad (1.1)$$

де m1ij – число переходів з і елемента в j в уже згадуваному тексті; n1i – загальне число переходів з і-го елемента; m2ij, n2i – аналогічні числа для матриці того автора, з яким проводиться порівняння; k – число елементів (якщо підраховуються буквосполучення, то k = 32). Значення L тим менше по модулю, ніж менше відмінність між матрицями.

1.5 Аналіз програмних засобів, що використовуються для аналізу авторства тексту

1.5.1 Лінгвоаналізатор

«Лінгвоаналізатор» [4] – це програма математичного аналізу структури тексту. Працює онлайн. Спеціалізується на творах Російської Фантастики. Програма визначає близькість вхідного тексту до одного з авторів і в результаті видає трьох найімовірніших авторів, для кожного вказуючи три найбільш близьких твори.

Як стверджує автор програми, дана модель пройшла перевірку на матеріалі понад вісімдесяти авторів із загальним обсягом творів 128 Мб і довела свою ефективність.

Крім цього, автори попереджають користувачів, що програма краще працює з великим об’ємом текстів, так як текст автора може піти від свого звичного стилю.

Дана програма не аналізує ідеї, фабулу і зміст тексту, а застосовує методику атрибуції, яка спирається на математичну модель, де враховані саме формальні характеристики:

1) Число службових слів, таких як прийменники, сполучники, частки і т.д.;

2) Морфеми слів (приставки, коріння, суфікси, закінчення) і їх послідовності;

3) Складність використовуваних граматичних конструкцій;

4) Словник, який використовується автором.

Основні характеристики та переваги:

– аналізатор радить авторів і літературу, схожу за стилем написання з розглянутою;

– після кожного аналізу заповнюється анкета, і творцеві проекту видно статистика правильних і неправильних результатів програми;

– детальний аналіз, наведена відсоткова статистика встановлення авторства;

– програма досить легка в управлінні, не вимагає якихось спеціальних навичок і знань;

– ввічлива форма звернення.

Недоліки:

– потрібно занадто великий розмір тексту;

– непривабливий інтерфейс;

– програма не оновлювалася з 2001 року, відповідно база письменників не поповнюється;

– програма аналізує тільки тексти з жанру Фантастики.

– правильно встановлює авторів тільки тих творів, які є в базі даних.

1.5.2 Стилеаналізатор

Система «Стилеаналізатор» [5] – Програма передбачає лінгвістичний аналіз історичних текстів і авторського стилю літературних творів при допомозі статистичних методів. Система використовує методи нейронних мереж і ієрархічної кластеризації для вирішення завдання визначення авторства. В якості заходи порівняння матриць частот появи ознак пропонується використовувати запобіжний Кульбака і міру хі-квадрат. Під частотним ознакою розуміється будь-яка ознака стилю тексту, що допускає можливість знаходження частоти його появи в тексті.

Основні характеристики та переваги:

– статистичний аналіз текстів (підрахунок значень ознак, факторний аналіз, в тому числі метод головних компонентів, ластерний аналіз, методи байєсівської класифікації),

– інформаційний аналіз і класифікацію (дерева рішень),

– логічний аналіз і тестове розпізнавання,

– нейронні мережі прямого поширення,

– самоорганізуються карти Кохонена (Self-Organizing Maps – SOM-мережі),

– класифікацію на основі суффіксних дерев.

– побудова гістограм розподілів ознак з накладенням кривої, ластерний розподіл для візуального аналізу.

Недоліки:

– трудомісткість. Число завантажених текстів, яке безпосередньо впливає на якість пошуку, вимагає великих ресурсів від обчислювальної системи (великий обсяг пам’яті і потужний процесор);

– неможливість прогнозування успішного результату. Немає ніякого критерію того, чи в правильному напрямку рухається пошук;

– немає механізмів, що визначають, скільки часу залишилося до кінця роботи алгоритму, до того моменту, коли подальший пошук не принесе своїх результатів;

Метод Хмельова і його модифікації перемагають як в швидкості навчання, так і в якості класифікації. Нейронні мережі дають порівнянне якість, але сильно програють в швидкості. Дерева рішень забезпечують найгіршу якість класифікації, але при цьому дають наочний вид рішення і по ходу проводять відбір найбільш інформативних ознак.

1.5.3 Авторознавець

Система «Авторознавець» [6] – заснована на застосуванні нейронних мереж в поєднанні з методом опорних векторів. Підсумкове рішення з питання атрибуції тексту приймається ансамблем класифікаторів за принципом мажоритарного голосування. В якості характерних ознак тексту для опису авторського стилю використовуються найбільш часті триграми символів і найбільш часті слова російської мови. Система містить набір DLL-бібліотек, які підключаються до текстового процесора Word for Windows і в головному меню з'являється новий пункт. Таким чином, дана програмна система дозволяє користувачу працювати в звичному для нього середовищі.

Основні характеристики та переваги:

– врахування особливостей російської мови при аналізі авторського стилю: морфологічні омонімії, особливостей словозміни і ла. Можливість и користування морфологічних і частотних словників;

– врахування особливостей електронних текстів, в тому числі і коротких повідомлень: відсутність розділових знаків, використання емотиконів, неправильних але розпізнаних символів і т.д.

– використання бази даних, заснованої на запропонованої ієрархічної моделі, для зберігання тексту і його характеристик на рівні символів, слів, пропозицій.

– можливість використання методів MLP, CCN, SVM для ідентифікації автора та винесення кінцевого рішення на основі об'єднання результатів роботи декількох методів за принципом більшості голосів.

– можливість використання методів one-class SVM і QSUM для перевірки тексту на однорідність;

– можливість визначення авторства як одного тексту, так і проведення комплексних досліджень по ідентифікації автора для безлічі текстів і різних обсягів текстових вибірок.

1.5.4 Атрибутор

«Атрибутор**»** [7] **–** дана програма є онлайн лінгвістичним процесором для машинного порівняння текстів і їх класифікації за параметрами індивідуального авторського стилю. Твори підбиралися так, щоб тексти різних письменників мали якомога більше відмінностей, а тексти одного письменника мали максимальні подібності. На даний момент система навчена порівнювати тільки тексти романів. Для атрибуції досить приблизно шість друкованих сторінок

Основні характеристики та переваги:

– ластер обробляє інформацію і видає результат.

– частково «дізнається» автора тексту, відсутньої в базу, за стилем творів автора, які перебувають в базі.

– програма досить легка в управлінні, не вимагає якихось спеціальних навичок і знань.

– для аналізу і оцінки індивідуального авторського стилю використовуються трьохбуквені поєднання – тріади.

Недоліки:

– У базу даних атрибутора потрапили в загалом романи і повісті російських письменників 19 – 20 століть.

– Для порівняння не приймаються тексти розміром менше 20 Кб (приблизно 20 сторінок)

1.5.5 СМАЛТ

Система **«**СМАЛТ**»** [8] – (Статистичні методи аналізу літературного тексту), заснований на алгоритмах автоматизації морфологічного і синтаксичного аналізу текстів Обробка текстів в розробленій системі виробляється в кілька етапів. На першому кроці виконується автоматизоване розбиття вихідного тексту на лексичні одиниці, серед яких виділяються

частина (або розділ), абзац, пропозицію, слово. На другому здійснюється морфологічний розбір тексту. На третьому – синтаксичний розбір.

Обробка текстів в даній системі проводиться поетапно:

1) виконання автоматизованого розбиття вихідного тексту на: розділ, абзац, пропозиція, слово;

2) здійснення автоматичної обробки тексту і його морфологічний розбір;

3) синтаксичний аналіз;

4) Виконання користувачем операцій з бази даних з аналізу текстів.

Недолік запропонованого методу полягає в тому, що завдання визначення авторства доводиться зводити до задачі побудови якісного і швидкого синтаксичного аналізатора. Остання із завдань є не менш важкою і до сих пір не вирішена на необхідному рівні.

В основі формальних методів атрибуції текстів лежить уявлення про те, що зі зростанням обсягу тексту параметри, що характеризують авторський стиль, стають стійкими з ймовірнісної точки зору, що дозволяє встановлювати авторство по стабільно повторюється формальним характеристикам тексту. Тому більш висока якість атрибуції досягається для текстів великого об'єму, і менш точний результат виходить для текстів маленького обсягу (табл. 1.1).

Таблиця 1.1 - Порівняння програмних засобів атрибуції текстів

Назва	Методи	Зміни параметрів введення	Засоби аналізу тексту	Необхідний обсяг тексту	Точність, %	Застосування до вирішення реальних завдань
«Лінгво-аналізатор»	Ентропійний підхід, марковскі ланцюга	Ні	Графем., стат. аналіз	40000-100000 символів	84-89	Ні

«Атрибутор»	Марковський ланцюг	Ні	Стат. аналіз	>20000 символів	Не від.	Ні
«СМАЛТ»	Критерії Стьюдента, Колмогорова-Смирнова, ластерний аналіз, мережі Хеммінга	Ні	Графем., морф., синт., стат. аналіз, підтримка до революційної орфографії	500 слів для визначення однорідності	Не від.	Так
«Стиле-аналізатор»	Марковський ланцюг	Так	Графем., стат. аналіз, робота с розмічеными текстами	30 000-40 000 символів	90-98	Так
«Авторознавець»	Нейронні мережі, метод опорних векторів, QSUM	Так	Графем., морф., стат. аналіз	20 000-25 000 символів	95-98	Так
				100 символів	76	

1.6 Формулювання завдань роботи

В дипломному проекті поставлена задача здійснити розробку і зробити аналіз ідентифікації авторства текстів, що забезпечує підвищення точності визначення частоту появи нових слів, зменшення обсягу текстової вибірки і зниження тимчасових витрат на прийняття рішення, і створення програмного комплексу для ідентифікації авторства на її основі.

Завдання:

- проаналізувати предметну область та існуючі програмні засоби, які її описують;
- спроектувати додаток для визначення авторства тексту по частоті появи нових слів;

– виконати аналіз тексту з використанням розробленого додатку.

1.7 Технічне завдання на розробку додатка

Назва розробки - Засоби аналізу авторства тексту по частоті появи нових слів.

Призначення розробки

Додаток на ОС Windows призначений для надання інформації людям для їх кращого орієнтування в идентифікації авторста тексту з використанням категоризації текстів. Категоризація документів полягає у співставленні документів з колекції з однією або декількома групами (класами, кластерами) схожих між собою текстів (наприклад, по темі або стилем).

Вимоги до функціональних можливостей

Проектований додаток має виконувати наступні функції:

– Можливість роздільного аналізу по першому і другому закону Зіпфа кожного документа.

– Можливість окремо визначити жанр кожного документа (реалізація з перерозподілом потоку даних).

– Можливість внесення змін в документ безпосередньо в процесі аналізу (при наявності в документі не індентифікуемих символів).

Вимоги до інтерфейсу

Інтерфейс користувача системи має бути інтуїтивно зрозумілим. Навігаційні елементи і функціональні кнопки повинні забезпечувати однозначне розуміння, умовні позначення повинні відповідати загальноприйнятим.

Інтерфейс користувача має складатися і повинен забезпечувати:

- Наочне, інтуїтивно зрозуміле представлення інформації.
- Рухоме, динамічне вікно для вибору ділянки рангу або кількості слів безпосередньо в процесі аналізу.
- Колірна розмітка аналізованих документів з їх графічною реалізацією.

Графічні елементи навігації повинні бути забезпечені альтернативним підписом.

Апаратні та програмні вимоги

Розроблюваний додаток має невеликий розмір і без проблем встановиться на будь-якому персональному комп'ютері.

Таблиця 1.2 – Мінімальні та рекомендовані системні вимоги

Системний компонент	Мінімальні системні вимоги	Рекомендовані системні вимоги
ОС	Windows XP SP3	Windows 7, 8, 10
Процесор	з тактовою частотою 800 MHz	Intel Pentium 4 3,2GHz
ОЗУ	256 Мб	512 Мб – 2 Гб
Відеоадаптер	VGA сумісний адаптер	VGA сумісний адаптер
Місто на HDD	50 Мб	70 Мб
Пристрої введення	Миша / клавіатура	Миша / клавіатура
Архітектура з розрядністю	32 біт	32 біт або 64 біт

Висновки до розділу 1

Проведено аналіз існуючих методів, характеристик тексту, програмних засобів, які використовуються для ідентифікації автора вітчизняними та зарубіжними дослідниками. Визначено актуальні напрямки досліджень і розробок.

В розділі надано обгрунтування необхідності розробки, проаналізовані існуючі програмні засоби для визначення авторства тексту, поставлена задача здійснити розробку і зробити аналіз ідентифікації авторства текстів, що забезпечує підвищення точності визначення частоту появи нових слів, зменшення обсягу текстової вибірки і зниження тимчасових витрат на прийняття рішення, і створення програмного комплексу для ідентифікації авторства на її основі. Розроблене технічне завдання на розробку.

2 МЕТОДИ АПРОКСИМАЦІЇ СТАТИСТИЧНИХ ДАНИХ

2.1 Апроксимація функцій за методом найменших квадратів

В інженерній діяльності часто виникає необхідність описати у вигляді функціональної залежності зв'язок між величинами, заданими таблично або у вигляді набору точок (x_i , y_i), де $i = 0, ..., n$ [11]. Як правило, ці табличні дані отримані експериментально і мають похибки. При апроксимації бажано отримати відносно просту функціональну залежність (наприклад, многочлен), яка дозволила б «згладити» експериментальні похибки, обчислювати значення функції в точках які не містяться у вихідній таблиці (рис. 2.1). Ця функціональна залежність повинна з достатньою точністю відповідати початковій табличній залежності.

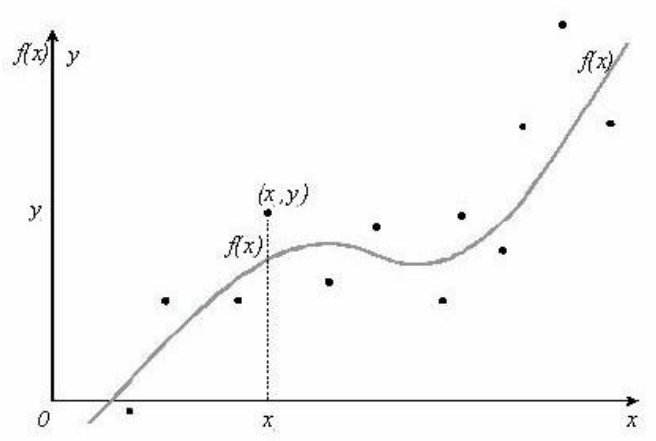


Рисунок 2.1 – Приклад набору точок та його апроксимації

В якості критерія точності найчастіше використовують критерій найменших квадратів, тобто визначають таку функціональну залежність $f(x)$, при якій $R = \sum_{i=0}^{0} (y_i - f(x_i))^2$ звертається в мінімум.

Розглянемо в якості функціональної залежності многочлен

$P_m(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m$, тоді $R = \sum_{i=0}^{0} (y_i - f(x_i))^2$.

Умови мінімуму - нульові приватні похідні по всім змінним $a_0, a_1, a_2 \cdots a_m$.

Тобто $\frac{dR}{da_k} = -\sum_{i=0}^{n}(y_i - a_0 - a_1 x_i - \cdots - a_m x_i)x_i^k = 0$, або

$\sum_{i=0}^{n}(y_i - a_0 - a_1 x_i - \cdots - a_m x_i)x_i^k = 0$, k = 0, 1, 2,…, m.

Збираємо коефіцієнти при невідомих $a_0, a_1, a_2 \cdots a_m$, отримуємо систему рівнянь:

$$a_0\sum_{i=0}^{n}x_i^k + a_1\sum_{i=0}^{n}x_i^{k+1} + a_2\sum_{i=0}^{n}x_i^{k+2} + \cdots + a_m\sum_{i=0}^{n}x_i^k y_i \text{ , } k = 0, 1, 2,\ldots, m$$

Надалі можна ввести позначення: $c_k = \sum_{i=0}^{n}x_i^k, b_k = \sum_{i=0}^{n}x_i^k y_i$ і переписати систему в розгорнутому вигляді:

$$\begin{cases} c_0 a_0 + c_1 a_1 + c_2 a_2 + \cdots c_m a_m = b_0 \\ c_1 a_0 + c_2 a_1 + c_3 a_2 + \cdots c_{m+1} a_m = b_1 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ c_m a_0 + c_{m+1} a_1 + c_{m+2} a_2 + \cdots c_{2m} a_m = b_m \end{cases}$$

Матриця даної системи називається матрицею Грамма. Вирішуючи цю систему лінійних рівнянь, отримуємо коефіцієнти $a_0, a_1, a_2 \cdots a_m$, які є шуканими параметрами емпіричної формули.

Розглянемо два окремих випадки m = 1 та m = 2.

1. Лінійна апроксимація (m = 1)

$$P_1(x) = a_0 + a_1 x$$

$$c_0 = \sum_{i=0}^{n}x_i^0 = n+1, c_1 = \sum_{i=0}^{n}x_i, c_2 = \sum_{i=0}^{n}x_i^2 y_i, b_0 = \sum_{i=0}^{n}y_i, b_1 = \sum_{i=0}^{n}x_i y_i$$

Таким чином система рівнянь має вигляд:

$$\begin{cases} (n=1)a_0 + \sum_{i=0}^{n} x_i a_1 = \sum_{i=0}^{n} y_i \\ \sum_{i=0}^{n} x_i a_0 + \sum_{i=0}^{n} x_i^2 a_1 = \sum_{i=0}^{n} x_i y_i \end{cases} \quad (2.1)$$

Систему (2.1) можна вирішити за методом Крамера.

$$\Delta = \begin{vmatrix} n+1 & \sum_{i=0}^{n} x_i \\ \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 \end{vmatrix},$$

$$a_0 = \frac{\begin{vmatrix} \sum_{i=0}^{n} y_i & \sum_{i=0}^{n} x_i \\ \sum_{i=0}^{n} x_i y_i & \sum_{i=0}^{n} x_i^2 \end{vmatrix}}{\Delta}, \qquad a_1 = \frac{\begin{vmatrix} n+1 & \sum_{i=0}^{n} y_i \\ \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i y_i \end{vmatrix}}{\Delta}$$

і, таким чином, отримати функцію $y = a_0 + a_1 x$

2. Квадратична апроксимація (m = 2).

$P_2(x) = a_0 + a_1 x + a_2 x^2$

Крім c_0, c_1, c_2, b_0, b_1 розраховуються

$$c_3 = \sum_{i=0}^{n} x_i^3, c_4 = \sum_{i=0}^{n} X_i^4, b_2 = \sum_{i=0}^{n} y_i x_i^2.$$

Розширена матриця системи рівнянь: $\left(\begin{array}{ccc|c} c_0 & c_1 & c_2 & b_0 \\ c_1 & c_2 & c_3 & b_1 \\ c_2 & c_3 & c_4 & b_2 \end{array}\right)$, вирішивши яку можна отримати знайдені коефіцієнти a_0, a_1, a_2.

Похибка наближення многочленом за методом найменших квадратів:

$$\Delta = \sqrt{\frac{1}{n+1} \sum_{i=0}^{n} \left(y_i - P_m(x_i)\right)^2}$$

Якщо експериментальні точки розташовуються уздовж деякої лінії, схожою за формою, наприклад, до графіка гіперболічної, показовою,

логарифмічною або інших функцій з невідомими параметрами вибирається в якості апроксимуючої. Потім проводиться *лінеаризація* цієї функції за допомогою заміни змінних і завдання зводиться до апроксимації залежності многочлена першого ступеня.

Наприклад:

1. Показова залежність: $y = ab^x$, приводиться до лінійного вигляду шляхом логарифмування $\ln y = \ln a + x \ln b$.

2. Степенева $y = ax^b$, аналогічно $\ln y = \ln a + b \ln x$.

3. Гіперболічна $y = \frac{x}{a + bx}$, приводиться до лінійного вигляду введенням нової змінної $Y = x / y$.

2.2 Визначення якості апроксимації

Якість моделей апроксимації може бути обчислена за допомогою наступних критеріїв [12]:

1) Середня помилка апроксимації

Середня помилка апроксимації - середнє відхилення розрахункових значень від фактичних:

$$\overline{A} = \frac{\sum \left| y_i - y_x \right| : y_i}{n} \cdot 100\%, \qquad (2.2)$$

де y_x - розрахункове значення за рівнянням.

Значення середньої помилки апроксимації до 15% свідчить про добре підібраній моделі рівняння.

2) Відносна помилка апроксимації

$$\overline{A} = \frac{\sum \left| y_t - y_i \right| : y_i}{n} \cdot 100\%, \qquad (2.3)$$

Помилка апроксимації в межах 5% -7% свідчить про хороший підбір рівняння до вихідних даних.

2.3 Апроксимація статистичних даних для об'єктів, що підпорядковуються законам Зіпфа

Найчастіше, для апроксимації статистичних даних для об'єктів, які підпорядкуватися Законами Зіпфа використовується гіперболічна функція виду [13]:

$$y = a + \frac{b}{a}, \tag{2.4}$$

де a, b – постійні коефіцієнти; x – статистичні дані аргументу функції (у вигляді списку); y – наближення значень функції до реальних даних отриманим методом найменших квадратів.

У нашому випадку вихідні дані задаються двома списками $x = \left[x_1 ... x_n\right], \left[y_1 ... y_n\right]$, де n – кількість даних у списках. Далі можна отримати функцію для визначення коефіцієнтів:

$$\mathrm{F}\left(a,b\right) = \sum_{i=0}^{n} y_i - \left(a + b / x_i \right)^2. \tag{2.5}$$

Коефіцієнти a, b знаходяться з наступної системи рівнянь:

$$\left.\begin{array}{l} \frac{dF(a,b)}{da} = 0 \\ \frac{dF(a,b)}{db} = 0 \end{array}\right\} \tag{2.6}$$

Рішення такої системи має наступний вигляд:

$$a = \frac{\sum_{i=1}^{n} y_i \cdot \sum_{i=1}^{n} \left(\frac{1}{x_i}\right)^2 - \sum_{i=1}^{n} \frac{1}{x_i} \cdot \sum_{i=1}^{n} \frac{y_i}{x_i}}{n\sum_{i=1}^{n} \left(\frac{1}{x_i}\right)^2 - \sum_{i=1}^{n} \frac{1}{x_i} \cdot \sum_{i=1}^{n} \frac{1}{x_i}} \quad (2.7)$$

$$b = \frac{n \cdot \sum_{i=1}^{n} \frac{y_i}{x_i} - \sum_{i=1}^{n} \frac{1}{x_i} \cdot \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} \left(\frac{1}{x_i}\right)^2 - \sum_{i=1}^{n} \frac{1}{x_i} \cdot \sum_{i=1}^{n} \frac{1}{x_i}} \quad (2.8)$$

Середня помилка апроксимації (2.2) Δ розраховується по формулі (2.9)

$$\Delta = \frac{\sum_{i=1}^{n} y_i - a - b / x_i}{n \cdot \sum_{i=1}^{n} y_i} \cdot 100 \quad (2.9)$$

У випадку, коли вихідні дані обираються з функції рівносторонньої гіперболи, a = 0, b = 10 і абсолютна похибка складає 0,004%. Значить функція mnkGP (x, y) працює правильно і її можна вставляти в прикладну програму.

2.3 Апроксимація даних поліномом

Згідно [14] для досліджуваних даних найкращим засобом апроксимації є поліном другого ступеня, оскільки в нього максимальний коефіцієнт достовірності R2.

Для апроксимації поліномом в Python є модуль scipy, але він не підтримує негативну ступінь d полінома.

Розглянемо код реалізації апроксимації даних поліномом.

```
#!/usr/bin/python
# coding: utf8
```

```
import scipy as sp
import matplotlib.pyplot as plt
def mnkGP(x,y):
              d=2 # ступінь полінома
              fp, residuals, rank, sv, rcond = sp.polyfit(x, y,
d, full=True) # Модель
              f = sp.poly1d(fp) # апроксмуюча функція
              print('Коефіцієнт -- a %s  '%round(fp[0],4))
              print('Коефіцієнт -- b %s  '%round(fp[1],4))
              print('Коефіцієнт -- c %s  '%round(fp[2],4))
              y1=[fp[0]*x[i]**2+fp[1]*x[i]+fp[2] for i in
range(0,len(x))] # значення функції a*x**2+b*x+c
              so=round(sum([abs(y[i]-y1[i]) for i in
range(0,len(x))])/(len(x)*sum(y))*100,4) # середня помилка
              print('Average quadratic deviation '+str(so))
              fx = sp.linspace(x[0], x[-1] + 1, len(x)) # можна
встановити замість len(x) більше число для інтерполяції
              plt.plot(x, y, 'o', label='Original data',
markersize=10)
              plt.plot(fx, f(fx), linewidth=2)
              plt.grid(True)
              plt.show()


x=[10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50, 54, 58, 62, 66,
70, 74, 78, 82, 86]
y=[0.1, 0.0714, 0.0556, 0.0455, 0.0385, 0.0333, 0.0294, 0.0263,
0.0238, 0.0217,
   0.02, 0.0185, 0.0172, 0.0161, 0.0152, 0.0143, 0.0135, 0.0128,
0.0122,
   0.0116] # дані для перевірки по функції y=1/x
mnkGP(x,y)
```

Як випливає з рис. 2.2, при апроксимації параболою даних, що змінюються по гіперболі середня помилка зростає, а вільний член квадратного рівняння звертається в нуль.

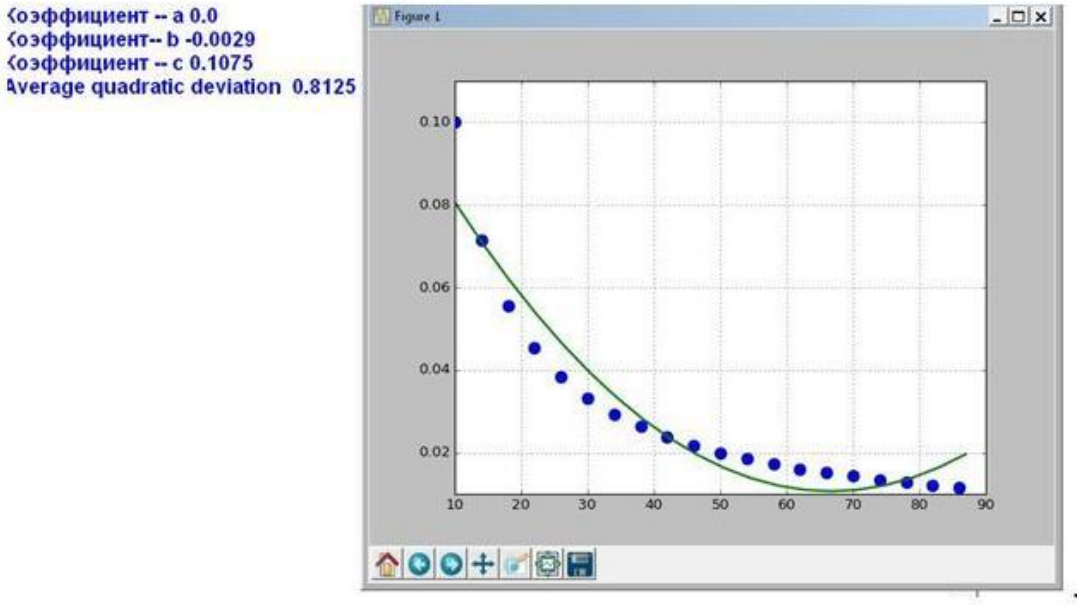


Рисунок 2.2 – Результат апроксимації параболою

Отримані функції будуть застосовані у наступному розділі для аналізу Законів Зіпфа.

Висновки до розділу 2

В розділі були сформульовані основні задачі, описаний алгоритм та розглянуті засоби апроксимації, які були обрані для реалізації додатку. Розглянуті методи апроксимації статистичних даних, що можуть використовуватися для рішення задачі порівняння текстів з використанням законів Зіпфа. Для подальших розрахунків використано гіперболічну функцію.

3 ПРОЕКТУВАННЯ ДОДАТКУ ДЛЯ ВИЗНАЧЕННЯ АВТОРСТВА ТЕКСТІВ З ВИКОРИСТАННЯМ МОВИ PYTHON

3.1 Мова програмування Python

Python - високорівнева інтерпретована мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і читання коду. Python надає конструкції, призначені для забезпечення можливості написання чітких програм як в маленькому, так і в великому масштабі.

Python підтримує різні парадигми програмування, зокрема об'єктно-орієнтоване, функціональне, імперативне і структурний. Також серед особливостей мови можна виділити автоматичну збірку сміття, динамічну типізацію, підтримка модульності, кроссплатформенность і велика кількість вже готових модулів, які дозволяють розробникам сконцентрувати увагу на вирішенні конкретної задачі. Також Python є мовою загального призначення, тому може застосовуватися практично в будь-якій області розробки ПО (standalone, клієнт-сервер, Web-додатки) і в будь-якій предметній області. Крім того, Python легко інтегрується з уже існуючими компонентами, що дозволяє впроваджувати Python в уже написані програми.

На даний момент існують три відомих реалізації середовища виконання для Python: CPython, Jython і Python.NET. Як можна здогадатися з назви, перша середа реалізована на мові C, друга на мові Java, а остання - на платформі .NET. Середовище виконання CPython зазвичай називається просто Python, і коли говорять про Python, то найчастіше мається на увазі саме ця реалізація. Ця реалізація складається з інтерпретатора і модулів розширення, написаних на мові C, і може використовуватися на будь-якій платформі, для якої доступний стандартний компілятор C. Крім того, існують вже скомпільовані версії середовища виконання для різних операційних систем, включаючи різні версії ОС Windows і різні дистрибутиви Linux. У цій

та наступних статтях буде розглядатися саме CPython, якщо інше не обумовлюється окремо. Середовище виконання Jython - це реалізація Python для роботи з віртуальною Java-машиною (JVM). Підтримується будь-яка версія JVM, починаючи з версії 1.2.2. Для роботи з Jython потрібна встановлена Java-машина (середовище виконання Java) і певне знання мови програмування Java. Вміти писати вихідний код на мові Java не обов'язково, проте доведеться мати справу с JAR-файлами і Java апплетамі, а також документацією в форматі JavaDOC. Яку версію середовища вибрати - залежить виключно від уподобань програміста, взагалі ж рекомендується тримати на комп'ютері і CPython, і Jython, так як вони не конфліктують між собою, а взаємно доповнюють один одного. Середа CPython працює швидше, так як немає проміжного рівня у вигляді JVM; крім того, оновлені версії Python спочатку випускають саме в вигляді середовища CPython.

Однак Jython може використовувати будь-який клас Java в якості модуля розширення і працювати на будь-якій платформі, для якої існує реалізація JVM. Обидва середовища виконання випущені під ліцензією, сумісною з відомою ліцензією GPL, тому можуть використовуватися для розробки як комерційного, так і вільного або безкоштовного ПО. Велика частина модулів розширення для Python також виходить в рамках ліцензії GPL і може вільно застосовуватися в будь-яких проектах, однак існують і комерційні розширення або розширення з більш строгими ліцензіями. Тому при використанні Python в комерційному проекті необхідно знати, які обмеження існують в ліцензіях модулів розширення.

3.2 Програма на Python для гіперболічної апроксимації з використанням закону Зіпфа

Згідно технічного завдання в програмі реалізовано функції роздільного аналізу по першому і другому закону Зіпфа кожного документа.

Розглянемо код реалізації апроксимації даних поліномом наведено у п.2.3. Текст програми наведений у Додатку А.

Для перевірки роботи програми використовувалися твори відомих англомовних письменників. У якості приклада проведено порівняльний аналіз творів Ден Браун «Код Давінчі» і «Ангели і демони» і Роберт Ладлема «Ідентифікація борна» [14].

3.3 Інтерфейс програми та ектсприменти

У перші два поля форми завантажуємо різні твори одного автора, а в третю іншого (рис. 3.1):

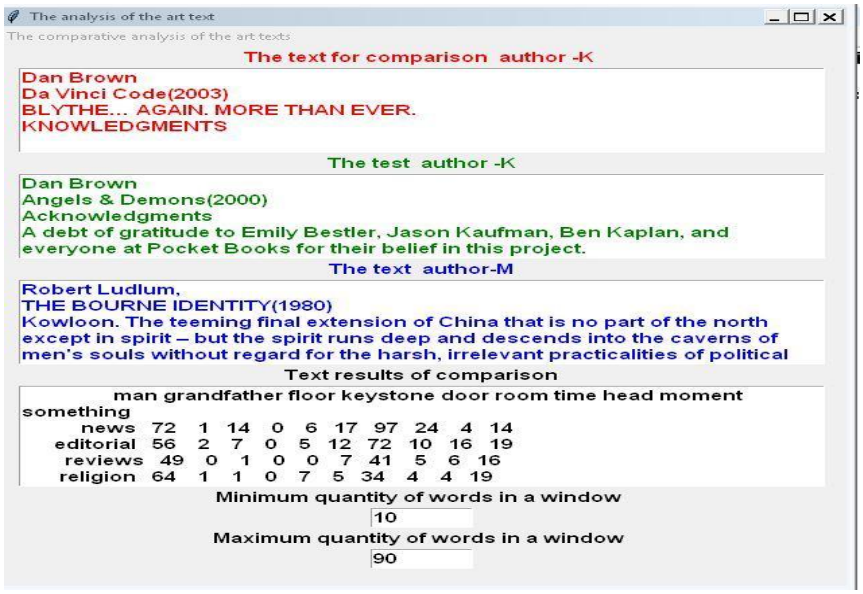


Рисунок 3.1 - Аналіз творів на авторство

Для визначення жанру творів вибираємо з тексту твору ключові іменники або модальні дієслова. Це можуть бути як окремі слова, так і словосполучення. З спеціально розмічених корпусів (я використовував Brown). По максимальному числу входжень відібраних слів визначається жанр (новини, редакційні повідомлення, огляди, релігія, хоббі, фентезі, наукова фантастика, пригоди, романи, гумористичні твори та ін.). Результат визначення жанру надано на рис. 3.2.

```
                man grandfather floor keystone door room time head moment
something
                news  72   1  14   0   6  17  97  24   4  14
           editorial  56   2   7   0   5  12  72  10  16  19
             reviews  49   0   1   0   0   7  41   5   6  16
            religion  64   1   1   0   7   5  34   4   4  19
             hobbies  16   2  22   0  10  14 127  54   5   9
                lore  88   1   9   0  14  24 174  38  16  21
      belles_lettres 219   1   8   0  15  24 225  26  36  49
          government  12   0   5   0   0   4 103   3   0   5
             learned  68   0   8   0  13  33 209  16  13  38
             fiction 111   1  18   0  47  63  99  54  36  45
             mystery 106   0  25   0  80  65  82  43  38  52
     science_fiction  17   0   1   0   1   2  30  15   3   8
           adventure 165   2  23   0  67  33 127  71  32  50
             romance  87   1  13   0  43  45  93  46  27  59
               humor  21   0   2   0   4  18  43  12  10  16
Text № -1- Theme-- belles_lettres. Concurrences- 603
```

Рисунок 3.2 – Результат визначення жанру твору

Як видно з рис. 3.3, твори, що підлягали дослідженню мають приблизно однаковий жанр, це важливо для їх подальшого порівняння.

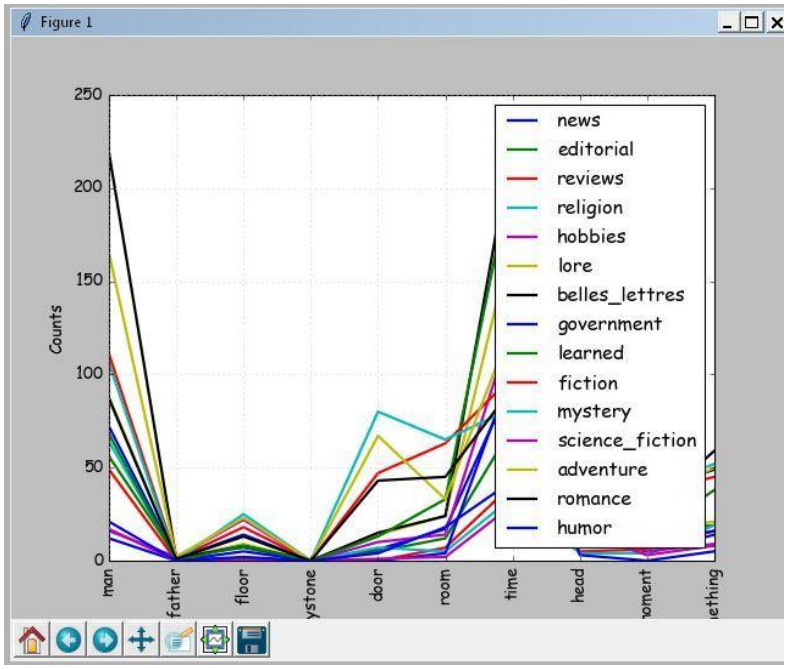


Рисунок 3.4 – Результат в графічному вигляді

Переконавшись, що всі тексти одного жанру можна починати аналіз авторства (рис. 3.5). Для розв'язання окремих завдань, наприклад для аналізу технічних текстів, можна створити свою базу тематик.

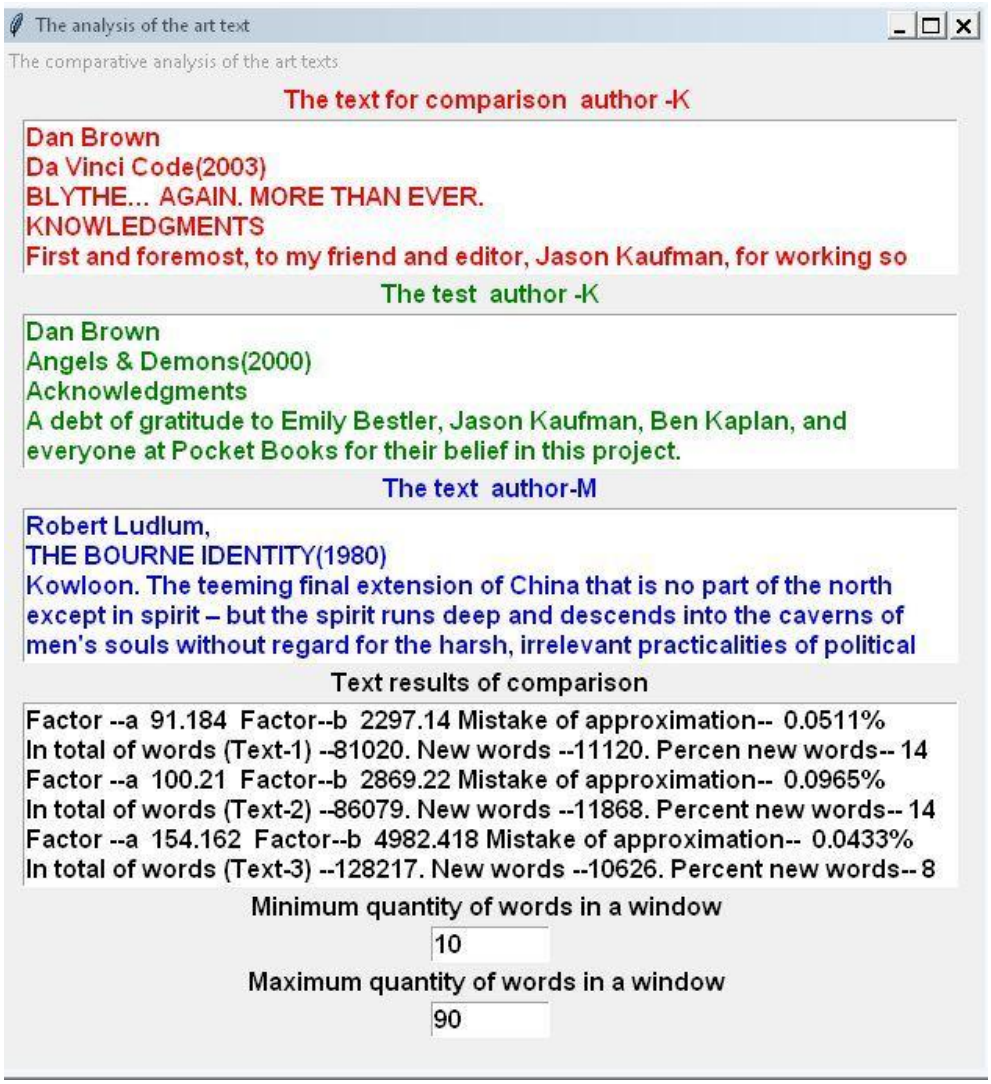


Рисунок 3.5 – Аналіз авторства

Програма будує фрагмент для кількості слів від 10 до 90 за другим Закону Зіпфа ─ «кількість - частота» [2]. В результаті отримуємо наступний розподіл:

```
Factor --a 91.184 Factor--b 2297.14 Mistake of approximation--
0.0511%
In total of words (Text-1) --81020. New words --11120. Percent
new words-- 14
Factor --a 100.21 Factor--b 2869.22 Mistake of approximation--
0.0965%
In total of words (Text-2) --86079. New words --11868. Percent
new words-- 14
Factor --a 154.162 Factor--b 4982.418 Mistake of approximation--
0.0433%
In total of words (Text-3) --128217. New words --10626. Percent
new words-- 8
Average distances between art products of the author K--25.062
```

```
Average distance between art products of the authors K and M--
138.25
```

З наведеної роздруківки і графіка (рис. 3.6) видно індивідуальність авторів: (K) - зелена і червона криві відповідають стилю Дена Брауна, а (M) - синя крива - Роберта Ладлема.

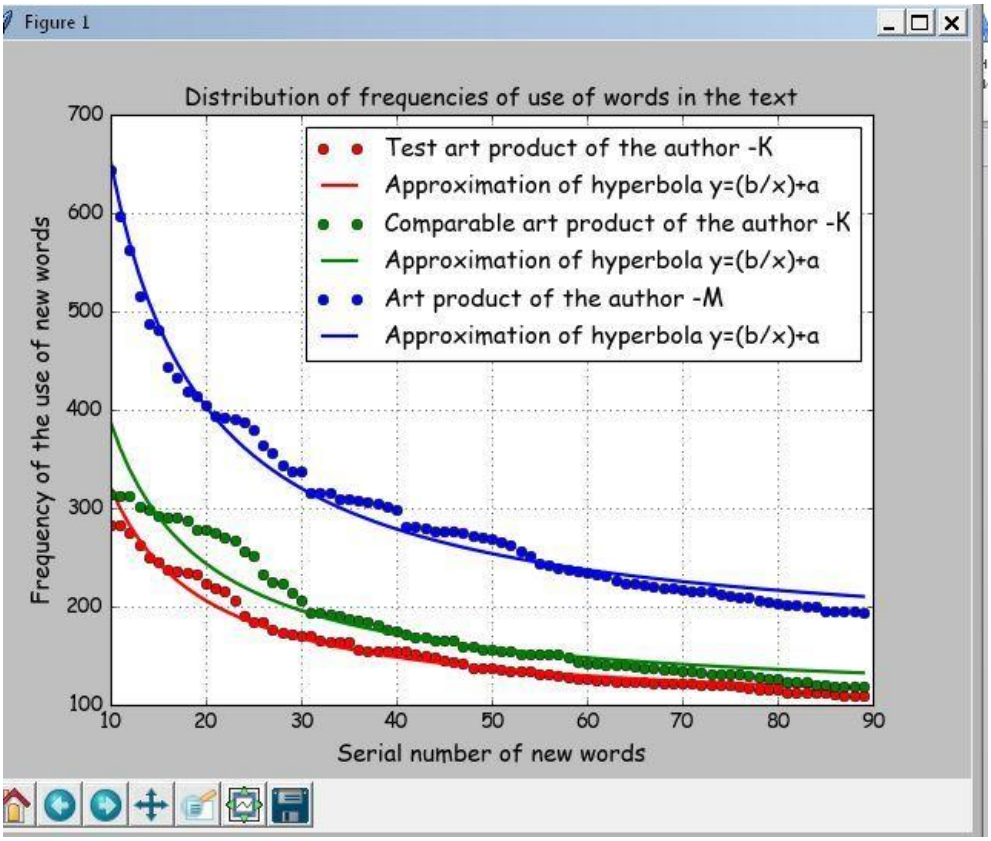


Рисунок 3.7 - Розподіл частот використання слів в тексті

Для розглянутого випадку середня відстань між апроксимуючими гіперболами автора K становить 25,062, а між першим твором автора K і твором автора M - 138,25. Для покращення результатів можна використати огляд виключно дієслів [15] або шаблони [16].

Висновки до розділу 3

Розглянуто реалізація на Python методу визначення авторства текстів по частоті вживання нових слів. Наведено формулу для порівняльного аналізу трьох текстів два з яких одного учасника, а третій іншого. Наведено приклад для порівняльного аналізу творів Дена Брауна і Роберта Ладлема.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Аналіз стану умов праці

4.1.1 Вимоги до приміщень

Згідно з ДСН 3.3.6.042-99 [17] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм - не менше 20 куб. м. У табл. 4.1. наведені розміри приміщення для якого виконуються розрахунки у даному розділі.

Таблиця 4.1 - Розміри приміщення.

Найменування	Значення
Довжина, м	5
Ширина, м	5
Висота, м	3
Площа, $м^2$	25
Об'єм, $м^3$	75

Виходячи з вищевикладеного зроблено висновок, що дане приміщення цілком відповідає зазначеним нормам.

4.1.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за ДСанПіН 3.3.2.007-98 [20] (табл. 4.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 4.2 - Характеристики робочого місця

Найменування параметра	Фактичне Значення	Нормативне Значення
Висота робочої поверхні, мм	750	680 ÷ 800
Висота простору для ніг, мм	730	не менше 600
Ширина простору для ніг, мм	660	не менше 500
Глибина простору для ніг, мм	700	не менше 650
Висота поверхні сидіння, мм	470	400 ÷ 500
Ширина сидіння, мм	400	не менше 400
Глибина сидіння, мм	400	не менше 400
Висота поверхні спинки, мм	600	не менше 300
Ширина опорної поверхні спинки, мм	500	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	800	700 ÷ 800

4.2 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

4.2.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 4.3). Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання НПАОП 0.00-1.28-10 [21], яке встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Переважно роботи за проектами виконують у кабінетах чи інших приміщеннях, де використовують різноманітне електрообладнання,

зокрема персональні комп’ютери (ПК) та периферійні пристрої. Основними робочими характеристиками персонального комп'ютера є:

– робоча напруга U=+220В +-5%;

– робочий струм I=2А;

– споживана потужність Р=350 Вт.

Таблиця 4.3 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна Оцінка	Нормативні Документи
1	2	3	4
Фізичні:			
підвищена або знижена вологість повітря	-//-	2	[17]
підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини	-//-	4	[18] [19]
Психофізіологічні:			
нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів; - виконання роботи за темою диплома, тестування; - оформлення роботи	4	[20] [21]
фізичні (статичне - сидіння)	порушення умов праці (організації місця праці: сидіння користувача) та організації робочого часу (безперервна робота)	2	[20] [21]

Робочі місця мають відповідати вимогам державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [20].

4.2.2 Пожежна безпека

Висока щільність елементів в електронних схемах призводить до значного підвищення температури окремих вузлів (80…100 ℃). При проходженні електричного струму по провідниках і деталей виділяється тепло, що в умовах їх високої щільності може привести до перегріву, і може служити причиною запалювання ізоляційних матеріалів. Слабкий опір ізоляційних матеріалів дії температури може викликати порушення ізоляції і привести до короткого замикання між струмоведучими частинами обладнання (шини, електроди).

Для гасіння пожеж в офісному приміщенні пропонується використовувати порошкові або вуглекислотні вогнегасники, так як вони є універсальними.

Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), надійно захищені діелектричними щитками та/або сітками з метою недопущення потрапляння працівника під напругу.

В приміщенні наявна затверджена «План-схема евакуації з кабінету (приміщення)».

Горючими матеріалами в приміщенні, де розташовані ЕОМ, є:

1) поліамід - матеріал корпусу мікросхем, горюча речовина, температура самозаймання 420℃;

2) полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335℃, температура самозаймання 530℃;

3) склотекстоліт ДЦ - матеріал друкарських плат, важкогорючий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;

4) пластикат кабельний №489 - матеріал ізоляції кабелів, горючий матеріал, показник горючості більше 2.1;

5) деревина - будівельний і обробний матеріал, з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, температура запалювання 255℃, температура самозаймання 399℃.

Простори усередині приміщень в межах, яких можуть утворюватися або знаходиться пожежонебезпечні речовини і матеріали відповідно до НАПБ Б.03.002-2007 [22] відносяться до пожежонебезпечної зони класу П-IIа. Це обумовлено тим, що в приміщенні знаходяться тверді горючі та важкозаймисті речовини та матеріали. Приміщенню, у якому розташоване робоче місце, присвоюється II ступень вогнестійкості.

Причинами можливого загоряння і пожежі можуть бути:

1) несправність електроустановки;

2) конструктивні недоліки устаткування;

3) коротке замикання в електричних мережах;

4) запалювання горючих матеріалів, що знаходяться в безпосередній близькості від електроустановки.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол [23].

4.2.3 Електробезпека

Виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три провідна мережа, шляхом прокладання

фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК, укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

4.3 Гігієнічні вимоги до параметрів виробничого середовища

4.3.1 Мікроклімат

Мікроклімат робочих приміщень - це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. В даному приміщенні проводяться роботи, що виконуються сидячи і не потребують динамічного фізичного напруження, то для нього відповідає категорія робіт 1а. Отже оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають ДСН 3.3.6.042-99 [17] і наведені в табл. 4.4:

Таблиця 4.4 – Норми мікроклімату робочої зони об’єкту

Період Року	Категорія Робіт	Температура C^0	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	Легка-1а	22-24	40-60	0,1
Тепла	Легка-1а	23-25	40-60	0,1

4.3.2 Освітлення

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше -1/8, в побутових - 1/10:

$$S_b = \left(\frac{1}{5} / \frac{1}{10}\right) * S_n \quad (4.1)$$

де S_b – площа віконних прорізів, м^2;

S_n – площа підлоги, м^2 .

$S_n = a \cdot b = 5 \cdot 5 = 25$ м^2 ,

$S = 1/8 \cdot 25 = 3{,}125$ м^2 .

Приймаємо 2 вікна площею S=1,6 м^2 кожне.

Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 5 м, ширина 5 м, світильниками ЛПО2П, оснащеними лампами типа ЛБ (дві по 80 Вт) з світловим потоком 5400 лм кожна. Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників n виробляється по формулі (4.2):

$$n = \frac{E * S * Z * K}{F * U * M} \quad (4.2)$$

де E - нормована освітленість робочої поверхні, визначається нормами – 300 лк; S - освітлювана площа, м^2 ; S = 25 м^2 ; Z - поправочний коефіцієнт світильника (Z = 1,15 для ламп розжарювання та ДРЛ; Z = 1,1 для люмінесцентних ламп) приймаємо рівним 1,1; K - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5; U - коефіцієнт

використання, залежний від типу світильника, показника індексу приміщення і т.п. - 0,575 М - число люмінесцентних ламп в світильнику - 2; F - світловий потік лампи - 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 * 25 * 1{,}1 * 1{,}5}{5400 * 0{,}575 * 2} \approx 2{,}0$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, які складаються з двох люмінесцентних ламп загальною потужністю 160 Вт, напругою - 220 В.

4.4 Вентилювання

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти), тобто при V приміщення > 40 м^3 на одного працюючого допускається природна вентиляція. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП.

Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

1) Заходи безпеки під час експлуатації персонального комп’ютера та периферійних пристроїв передбачають:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;

- експлуатацію сертифікованого обладнання;

- дотримання заходів електробезпеки;

- забезпечення оптимальних параметрів мікроклімату;

- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення);

- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціювання повітря:

а) якщо об'єм приміщення 20 $м^3$, то потрібно подати не менш як 30 $м^3$ /год повітря;

б) якщо об'єм приміщення у межах від 20 до 40 $м^3$, то потрібно подати не менш як 20 $м^3$/год повітря;

в) якщо об'єм приміщення становить понад 40 $м^3$, допускається природна вентиляція, у випадку, коли немає виділення шкідливих речовин.

2) Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:

- постійно стежити за справним станом електромережі, розподільних щитків, вимикачів, штепсельних розеток, лампових патронів, а також мережевих кабелів живлення, за допомогою яких електроприлади під'єднують до електромережі;

- постійно стежити за справністю ізоляції електромережі та мережевих кабелів, не допускаючи їхньої експлуатації з пошкодженою ізоляцією;

- не тягнути за мережевий кабель, щоб витягти вилку з розетки;

- не закривати меблями, різноманітним інвентарем вимикачі, штепсельні розетки;

- не підключати одночасно декілька потужних електропристроїв до однієї розетки, що може викликати надмірне нагрівання провідників, руйнування їхньої ізоляції, розплавлення і загоряння полімерних матеріалів;

- не залишати включені електроприлади без нагляду;

4.5.1 Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом [24], приміщення в якому проводяться всі роботи відноситься до першого класу (без підвищеної небезпеки). Під час роботи використовуються електроустановки з напругою живлення 36 В, 220 В, та 360 В. Опір контуру заземлення повинен мати не більше 4 Ом.

Послідовність розрахунку.

1) Визначається необхідний опір штучних заземлювачів $R_{шт.з}$.:

$$R_{шт.з} = \frac{R_д * R_{пр.з}}{R_{пр.з} - R_д} \quad (4.3)$$

де $R_{пр.з}$. - опір природних заземлювачів;

$R_д$ - допустимий опір заземлення.

Якщо природні заземлювачі відсутні, то $R_{шт.з}$.=$R_д$.

Підставивши числові значення у формулу (А.3), отримуємо:

$$R_{шт.з} = \frac{4 * 40}{40 - 4} \approx 4Ом$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту ρ, Ом·м. Приблизне значення питомого опору глини приймаємо ρ=40 Ом*м (табличне значення).

3) Розрахунковий питомий опір ґрунту, Ррозр, Ом·м, визначається відповідно для вертикальних заземлювачів ρрозр.в, і горизонтальних Ррозр.г, Ом·м за формулою:

$$P_{розр.} = \Psi * p \quad (4.4)$$

де ψ - коефіцієнт сезонності для вертикальних заземлювачів I кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів $Р_{розр.в}$=1,7 і горизонтальних $Р_{розр.в}$ =5,5 Ом·м

$Р_{озр.в}$ = 1, 7 * 40 = 68 Ом/м

$Р_{озр.г}$ = ·5,5 * 40 = 220 Ом/м

4) Розраховується опір розтікання струму вертикального заземлювача Rв, Ом, за (4.5).

$$R_B = \frac{P_{розр.в}}{2*\pi*1_B} * \left(\ln \frac{2*1_B}{d_{CT}} + \frac{1}{2} * \ln \frac{4*t+1_B}{4*t-1_B} \right) \quad (4.5)$$

де $l_в$ - довжина вертикального заземлювача (для труб - 2−3 м; $l_в$=3 м);

$d_{ст}$ - діаметр стержня (для труб - 0,03−0,05 м; $d_{ст}$=0,05 м);

t - відстань від поверхні землі до середини заземлювача, яка визначається за ф. (4.6):

$$t = h_B + \frac{1_B}{2} \quad (4.6)$$

де $h_в$ - глибина закладання вертикальних заземлювачів (0,8 м); тоді

$$t = 0.8 + \frac{3}{2} = 2{,}3м$$

$$R_B = \frac{68}{2*\pi*3} * \left(\ln \frac{2*3}{0{,}05} + \frac{1}{2} * \ln \frac{4*2{,}3+3}{4*2{,}3-3} \right) = 18{,}5Ом$$

5) Визначається теоретична кількість вертикальних заземлювачів n штук, без урахування коефіцієнта використання ηв:

$$n = \frac{2*R_B}{R_д} = \frac{2*18{,}5}{4} = 9{,}25 \quad (4.7)$$

6) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання $n_в$, шт:

$$n_B = \frac{2*R_B}{R_д * \eta_B} = \frac{2*18{,}5}{4*0{,}57} = 16{,}2 \approx 16 \quad (4.8)$$

7) Визначається довжина з'єднувальної стрічки горизонтального заземлювача lc, м:

$$l_c = 1{,}05 * L_B * (n_B - 1) \quad (4.9)$$

де $l_в$ - відстань між вертикальними заземлювачами, (прийняти за $L_в$ = 3м);

$n_в$ - необхідна кількість вертикальних заземлювачів.

$$l_c = 1{,}05 * 3 * (16 - 1) \approx 48м$$

8) Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки) $R_г$, Ом:

$$R_Г = \frac{P_{розр.г}}{2*\pi*l_c} * \ln\frac{2*l_c^2}{d_{ст}*h_Г} \quad (4.10)$$

де $d_{см}$ - еквівалентний діаметр смуги шириною b, $d_{см}$ =0,95b, b = 0,15 м;

$h_г$ - глибина закладання горизонтальних заземлювачів (0,5 м);

$l_с$ - довжина з'єднувальної стрічки горизонтального заземлювача $l_с$, м

$$R_Г = \frac{220}{2*\pi*48} * \ln\frac{2*48^2}{0{,}95*0{,}15*0{,}5} = 8{,}1Ом$$

9) Визначається коефіцієнт використання горизонтального заземлювача $\eta_с$. відповідно до необхідної кількості вертикальних заземлювачів $n_в$. Коефіцієнт використання з'єднувальної смуги $\eta_с$=0,3 (табличне значення).

10) Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{заг} = \frac{R_В * R_Г}{R_В * \eta_c + R_Г * n_В * \eta_В} \quad (4.11)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: $R_{заг}$ Ом, а саме:

$$R_{заг} = \frac{18{,}5*8{,}1}{18{,}5*0{,}3+8{,}1*16*0{,}57} = 1{,}9 \le R_д$$

Висновки до розділу 4

В розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

ВИСНОВКИ

Об'єктом розробки даного дипломного проекту є додаток для ОС Windows – програма для визначення авторства тексту по частоті появи нових слів з використанням мови Python.

Були поставлені та вирішуються наступні завдання:

1) Вибір методу і програмних засобів для розробки програми.

2) Виконано порівняльний аналіз аналогічних програм.

3) Розглянута та проаналізована високорівнева мова програмування Python.

4) Обґрунтована актуальність створення додатка на ОС Windows.

5) Проведено розробку додатка.

Додаток більше зорієнтований для викладачів, студентів та школярів.

Були сформульовані основні задачі та описані методи та засоби, які були обрані для реалізації додатку. Також були враховані недоліки аналогічних додатків які були розглянені. Додаток аналізатор був створений відповідно до технічного завдання на розробку та протестований.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Мета-книга і залежні від розміру властивості письмової мови. Бернхардссон С. , Корреа де Роча Л.Є. и Міннхаген П. Изд. IOP и Deutsche Physikalische Gesellschaft. Новий журнал фізики , том 11. // URL: http://iopscience.iop.org/article/10.1088/1367-2630/11/12/123015

2. Закони Д. Зіпфа. // URL: http://tpl-it. wikispaces.com/Законы+Зипфа+%28Ципфа%29 (дата звернення 10.06.2018).

3. Формула другого Закону Зіпфа к додатку. // URL: https://habr.com/post/323206/ (дата звернення 10.06.2018).

4. Хмельов Д. Коротка історія розробки методик визначення авторського стилю. Лінгвоаналізатор. // URL: http://rusf.ru/books/analysis/history.htm/ (дата звернення 06.06.2018).

5. Словникової-аналітичний блок системи «Стилеаналізатор» / В. В. Піддубний, О. Г. Шевельов, А. С. Кравцова, А. А. Фатих // Наукова творчість молоді: матеріали XIV Всеросійської науково-практичної конференції, 15-16 квітня 2010 року. Томськ, 2010. Ч. 1. С. 138-140.

URL: http: //vital.lib.tsu.ru/vital/access/manager/Repository/vtls: 000420899

6. Батура Т.В. Інститут систем інформатики А.П. Ершова. Формальні методи визначенням авторства тексту. // URL: https://cyberleninka.ru/article/v/formalnye-metody-opredeleniya-avtorstva-tekstov

7. Морозов Н.А. Лінгвістичні спектри: засіб для відрізнення плагіатом від істинних творів того чи іншого відомого нового автора // URL: http://www.textology.ru/library/book.aspx? BookId = 1 & textId = 3

8. Статистичні методи аналізу літературного тексту // URL: http://smalt.karelia.ru (дата звернення 06.06.2018).

5. Шеннон К. Роботи по теорії інформації і кібернетики. М .: Изд-во іноземної літератури, 1963. 830 с.

9. Яглом А.М., Яглом І.М. Імовірність і інформація. 3-е изд. М .: Наука, 1973. 511 с.

10. Алферов А.П., Зубов А.Ю., Кузьмін А.С. та ін. Основи криптографії. М .: Геліос АРВ, 2002. 480 с.

11. Родіонова Е.С. Методи атрибуції художніх текстів // Структурна та прикладна лінгвістика: Изд-во СПбГУ, 2008. Вип. 7. 127 с.

9. Батура Т.В. Формальні методи визначення авторства текстів. // Вісник НГУ.Сер. Інформаційні технології, 2012.

10. Шевельов О.Г. Порівняльний аналіз ефективності алгоритмів розпізнавання авторства текстів по частотах переходів // Кібернетика, 2005.

11. Аппроксимация функций по методу наименьших квадратов. // URL: http://mvm-math.narod.ru/Lec_PM5.pdf (дата звернення 06.06.2018).

12. Аппроксимация функций. Основные понятия и определения. // URL: http://files.khadi.kharkov.ua/laboratoriji/item/download/11122_a325857b69f3d193804f5aec8014f413.html (дата звернення 06.06.2018).

13. Простая программа на Python для гиперболической аппроксимации статистических данных // URL: https://habr.com/post/322954/#comment_83142 (дата звернення 06.06.2018).

14. Тараненко Ю. Программа на PYTHON для определения авторства текста по частоте появления новых слов // URL: https://habr.com/users/scorobey/posts/page6/ (дата звернення 06.06.2018).

15. Exploring Zipf’s Law with Python, NLTK, SciPy, and Matplotlib // URL: https://dzone.com/articles/exploring-zipf%E2%80%99s-law-python

16. S. Bird, Ewan Klein, E. Loper Natural Language Processing with Python. O’Reilly, 2009.

17. ДСН 3.3.6.042-99. Санітарні норми мікроклімату виробничих приміщень. Міністерство охорони здоров'я України (МОЗ). Постанова № 42 від 01.12.1999

18. ГОСТ 12.1.030-81 ССБТ. Електробезпека. Захисне заземлення. Занулення.

19. ГОСТ 13109-97. Норми якості електричної енергії в системах електропостачання загального призначення.

20. ДСанПІН 3.3.2.007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Міністерство охорони здоров'я України (МОЗ). Затверджено постановою № 7 головного державного санітарного лікаря України 10 грудня 1998 р.

21. НПАОП 0.00-1.28-10. Про погодження матеріалів правил охорони праці під час експлуатації електронно-обчислювальних машин. ДЕРЖАВНИЙ КОМІТЕТ УКРАЇНИ З ПРОМИСЛОВОЇ БЕЗПЕКИ, ОХОРОНИ ПРАЦІ ТА ГІРНИЧОГО НАГЛЯДУ. Наказ №65 від 23.06.2010.

22. НАПБ Б.03.002-2007. Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою. Наказ МНС № 833 від 03.12.2007 року.

23. ГОСТ 12.1.044-89. Система стандартів безпеки праці. Вогнестійкість. Номенклатура показників і методи їх визначення (ІСО 4589-84).

24. НПАОП 40.1-1.01-97. Правила безпечної експлуатації електроустановок. Наказ № 257 державного комітету України по нагляду за охороною праці від 6 жовтня 1997 р.

ДОДАТОК А

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import nltk
from nltk import *
from nltk.corpus import brown
stop_words= nltk.corpus.stopwords.words('english')
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
mpl.rcParams['font.family'] = 'fantasy'
mpl.rcParams['font.fantasy'] = 'Comic Sans MS, Arial'
from nltk.stem import SnowballStemmer
stop_symbols = '.,!?:;"-\n\r()'
def comor_text():
        # функция стемминга NLTK - быстрее яем словарна
лемитизация
        stemmer = SnowballStemmer('english')
        # контроль корректности данных
        if len(txt.get(1.0,END))!=1 and
len(txt1.get(1.0,END))!=1 and len(txt2.get(1.0,END))!=1:

mrus=[txt.get(1.0,END),txt1.get(1.0,END),txt2.get(1.0,END)]
                mr=3 # переменная для раздельного анализа
графиков
        elif len(txt.get(1.0,END))!=1 and
len(txt1.get(1.0,END))!=1 and len(txt2.get(1.0,END))==1:
                mrus=[txt.get(1.0,END),txt1.get(1.0,END)]
                mr=2
        elif len(txt.get(1.0,END))!=1 and
len(txt1.get(1.0,END))==1 and len(txt2.get(1.0,END))==1:
                mrus=[txt.get(1.0,END)]
                mr=1
        else:
                  txt3.insert(END,"There are no all texts")
```

```
                return
        # стемминг, отбор стоп слов и создание частотных
словарей
        for text in mrus:
                v=([stemmer.stem(x) for x in
[y.strip(stop_symbols) for y in text.lower().split()] if x and
(x not in stop_words)])
                #частотный словарь частота употребления слова -
ранг
                my_dictionary=dict([])
                z=[]
                for w in v:
                        if w in my_dictionary:
                                my_dictionary[w]+=1
                        else:
                                my_dictionary[w]=1
                max_count=int(txt5.get(1.0,END))
                min_count=int(txt4.get(1.0,END))
                if len(my_dictionary)<max_count:
                        txt3.insert(END,"It is not enough of
words for the analysis ")
                        return
                #частотный словарь частота употребления слова -
колличество слов
                my_dictionary_z=dict([])
                for key,val in my_dictionary.items():
                        if val in my_dictionary_z:
                                my_dictionary_z[val]+=1
                        else:
                                my_dictionary_z[val]=1
                        z.append(val)
                z.sort(reverse=True)
                # получение исходных данных для построения
графиков частотного распределения
                e=z[ min_count:max_count]
```

```
                ee=[my_dictionary_z[val] for val in z][
min_count:max_count]
                ee=np.arange(len(my_dictionary))[
min_count:max_count]
                if text==mrus[0]: # расчёт гиперболическойй
аппроксимации -a,b для первого документа + % новых слов

zz=round((float(len(my_dictionary))*100)/(float(len(v))),0)
                        tt=('In total of words (Text-1) --%i.
New words --%i. Percen new words-- %i'%( len(v),len(
my_dictionary),int(zz)))
                        xData1 = ee
                        yData1 = e
                        z=[1/w for w in  ee]
                        z1=[(1/w)**2 for w in  ee]
                        t=[ round(e[i]/ee[i],4)  for i in
range(0,len(ee)) ]
                        a=round((sum(e)*sum(z1)-
sum(z)*sum(t))/(len(ee)*sum(z1)-sum(z)**2),3)
                        b=round((len(ee)*sum(t)-
sum(z)*sum(e))/(len(ee)*sum(z1)-sum(z)**2),3)
                        y1=[round(a+b/w ,4) for w in ee]
                        s=[round((y1[i]-e[i])**2,4) for i in
range(0,len(ee))]
                        sko=round(round((sum(s)/(len(ee)-
1))**0.5,4)/(sum(y1)/len(ee)),4)
                        tg='Factor --a  '+str(a)+'  Factor--b
'+str(b)+' Mistake of approximation--  '+str(sko)+"%"+"\n"+tt
                        txt3.delete(1.0, END)
                        txt3.insert(END,tg)
                        txt3.insert(END,'\n')
                        y1Data1=y1
                elif text==mrus[1]:# расчёт аппроксимации -a,b
для втого документа + % новых слов

zz=round((float(len(my_dictionary))*100)/(float(len(v))),0)
```

```
                        tt=('In total of words (Text-2) --%i.
New words --%i. Percent new words-- %i'%( len(v),len(
my_dictionary),int(zz)))
                        xData2 = ee
                        yData2=e
                        z=[1/w for w in  ee]
                        z1=[(1/w)**2 for w in  ee]
                        t=[ round(e[i]/ee[i],4)  for i in
range(0,len(ee)) ]
                        a=round((sum(e)*sum(z1)-
sum(z)*sum(t))/(len(ee)*sum(z1)-sum(z)**2),3)
                        b=round((len(ee)*sum(t)-
sum(z)*sum(e))/(len(ee)*sum(z1)-sum(z)**2),3)
                        y1=[round(a+b/w ,4) for w in ee]
                        s=[round((y1[i]-e[i])**2,4) for i in
range(0,len(ee))]
                        sko=round(round((sum(s)/(len(ee)-
1))**0.5,4)/(sum(y1)/len(ee)),4)
                        tg='Factor --a  '+str(a)+'  Factor--b
'+str(b)+' Mistake of approximation--  '+str(sko)+"%"+"\n"+tt
                        txt3.insert(END,tg)
                        txt3.insert(END,'\n')
                        y1Data2=y1
                elif text==mrus[2]:# расчёт аппроксимации -a,b
для третьего документа + % новых слов

zz=round((float(len(my_dictionary))*100)/(float(len(v))),0)
                        tt=('In total of words (Text-3) --%i.
New words --%i. Percent new words-- %i'%( len(v),len(
my_dictionary),int(zz)))
                        xData3 = ee
                        yData3=e
                        z=[1/w for w in  ee]
                        z1=[(1/w)**2 for w in  ee]
                        t=[ round(e[i]/ee[i],4)  for i in
range(0,len(ee)) ]
```

```
                        a=round((sum(e)*sum(z1)-
sum(z)*sum(t))/(len(ee)*sum(z1)-sum(z)**2),3)
                        b=round((len(ee)*sum(t)-
sum(z)*sum(e))/(len(ee)*sum(z1)-sum(z)**2),3)
                        y1=[round(a+b/w ,4) for w in ee]
                        s=[round((y1[i]-e[i])**2,4) for i in
range(0,len(ee))]
                        sko=round(round((sum(s)/(len(ee)-
1))**0.5,4)/(sum(y1)/len(ee)),4)
                        tg='Factor --a  '+str(a)+'  Factor--b
'+str(b)+' Mistake of approximation--  '+str(sko)+"%"+"\n"+tt
                        txt3.insert(END,tg)
                        txt3.insert(END,'\n')
                        y1Data3=y1
        if mr==3: # построение графиков для первого и третьего
документа + среднее расстояние между их аппроксимацией
                r12=round(sum([abs(yData1[i]-yData2[i]) for i in
range(0,len(xData1))])/len(xData1),3)
                txt3.insert(END,"Average distances between art
products of the author K--"+ str(r12))
                txt3.insert(END,'\n')
                r13=round(sum([abs(yData1[i]-yData3[i]) for i in
range(0,len(xData1))])/len(xData1),3)
                txt3.insert(END,"Average distance between art
products of the authors K and M--"+ str(r13))
                txt3.insert(END,'\n')
                plt.title('Distribution of frequencies of use of
words in the text', size=14)
                plt.xlabel('Serial number of new words',
size=14)
                plt.ylabel('Frequency of the use of new words',
size=14)
                plt.plot(xData1, yData1, color='r', linestyle='
', marker='o', label='Test art product of the author -K')
                plt.plot(xData1, y1Data1, color='r',linewidth=2,
label='Approximation of hyperbola y=(b/x)+a')
```

```
                plt.plot(xData2, yData2, color='g', linestyle='
', marker='o', label='Comparable art product of the author -K')
                plt.plot(xData2, y1Data2, color='g',linewidth=2,
label='Approximation of hyperbola y=(b/x)+a')
                plt.plot(xData3, yData3, color='b', linestyle='
', marker='o', label='Art product of the author -M')
                plt.plot(xData3, y1Data3, color='b',linewidth=2,
label='Approximation of hyperbola y=(b/x)+a')
                plt.legend(loc='best')
                plt.grid(True)
                plt.show()
        elif mr==2:# построение графиков для первого и второго
документа + среднее расстояние между их аппроксимацией
                r12=round(sum([abs(yData1[i]-yData2[i]) for i in
range(0,len(xData1))])/len(xData1),3)
                txt3.insert(END,"Average distances between art
products of the author K--"+ str(r12))
                txt3.insert(END,'\n')
                plt.title('Distribution of frequencies of use of
words in the text', size=14)
                plt.xlabel('Serial number of new words',
size=14)
                plt.ylabel('Frequency of the use of new words',
size=14)
                plt.plot(xData1, yData1, color='r', linestyle='
', marker='o', label='Test art product of the author -K')
                plt.plot(xData1, y1Data1, color='r',linewidth=2,
label='Approximation of hyperbola y=(a/x)+b')
                plt.plot(xData2, yData2, color='g', linestyle='
', marker='o', label='Comparable art product of the author -K')
                plt.plot(xData2, y1Data2, color='g',linewidth=2,
label='Approximation of hyperbola y=(a/x)+b')
                plt.legend(loc='best')
                plt.grid(True)
                plt.show()
```

```
        elif mr==1: # построение графика для любого загруженного
документа
                plt.title('Distribution of frequencies of use of
words in the text', size=14)
                plt.xlabel('Serial number of new words',
size=14)
                plt.ylabel('Frequency of the use of new words',
size=14)
                plt.plot(xData1, yData1, color='r', linestyle='
', marker='o', label='Test art product of the author -K')
                plt.plot(xData1, y1Data1, color='r',linewidth=2,
label='Approximation of hyperbola y=(a/x)+b')
                plt.grid(True)
                plt.show()
def choice_text():# загрузка документов в поля формы
                try:
                        op = askopenfilename()
                        f=open(op, 'r')
                        st=f.read()
                        f.close()
                        if len(txt.get(1.0,END))==1:
                                txt.insert(END,st)
                        elif len(txt1.get(1.0,END))==1:
                                txt1.insert(END,st)
                        elif len(txt2.get(1.0,END))==1:
                                txt2.insert(END,st)
                except:
                        pass
def array_text_1 ():# чтение данных из поля уже в UNICODE
        if len(txt.get(1.0,END))!=1:
                u=txt.get(1.0,END)
        else:
               txt3.insert(END,"There are no text №1")
               return
        op=1
        processing_subjects (u,op)
```

```
def array_text_2 ():# чтение данных из поля уже в UNICODE
        if  len(txt1.get(1.0,END))!=1:
                u=txt1.get(1.0,END)
        else:
               txt3.insert(END,"There are no text №2")
               return
        op=2
        processing_subjects (u,op)
def array_text_3 ():# чтение данных из поля уже в UNICODE
        if len(txt2.get(1.0,END))!=1:
                u=txt2.get(1.0,END)
        else:
               txt3.insert(END,"There are no  text №3")
               return
        op=3
        processing_subjects (u,op)
def processing_subjects (u,op):# определние жанра текста (
NLTK+corpusbrown)
        q= nltk.word_tokenize(u)
        qq=[w for w in q if len(w)>2]
        z=nltk.pos_tag(qq)
        m=[w[0].lower() for w in z if w[1]=="NN"]
        d={}
        for w in m:
                if w in d:
                        d[w]+=1
                else:
                        d[w]=1
        pairs = list(d.items())
        pairs.sort(key=lambda x: x[1], reverse=True)
        modals=[]
        wq=10
        for i in pairs[0:wq]:
                modals.append(i[0])
        cfd = nltk.ConditionalFreqDist(
                        (genre, word)
```

```
                        for genre in brown.categories()
                        for word in
brown.words(categories=genre))
        #задание жанров для определения
        genres=['news', 'editorial', 'reviews', 'religion',
'hobbies', 'lore', 'belles_lettres',
        'government', 'learned', 'fiction', 'mystery',
'science_fiction', 'adventure', 'romance', 'humor']
        sys.stdout = open('out.txt', 'w')
        cfd.tabulate(conditions=genres, samples=modals)
        sys.stdout.close()# перенаправление потоков
        f=open('out.txt', 'r')
        w=f.read()
        txt3.insert(END,w)
        f.close()
        sys.stdout = open('out.txt', 'w')
        cfd.tabulate(conditions=genres, samples=modals)
        sys.stdout.close()
        f=open('out.txt', 'r')
        b=0
        u={}
        for i in f:
                b=b+1
                if b>=2:
                        d=i.split()
                        c=d[1:len(d)]
                        e=[int(w) for w in c]
                        u[d[0]]=sum(e)
        for key, val in u.items():
                if val == max(u.values()):
                      tex="Text № -%i- Theme-- %s. Concurrences-
%i"%(op,key,val)
        txt3.insert(END,tex)
        txt3.insert(END,'\n')
        f.close()
        cfd.plot(conditions=genres, samples=modals)
```

```
def close_win():
    tk.destroy()
# интерфейс tkinter + меню+индивидуальная цветоовая разметка
текстов+ центрирование формы
import tkinter as T
from tkinter.filedialog import *
import tkinter.filedialog
import fileinput
tk=T.Tk()
tk.geometry('630x630')
main_menu = Menu(tk)
tk.config(menu=main_menu)
file_menu = Menu(main_menu)
main_menu.add_cascade(label="The comparative analysis of the art
texts", menu=file_menu)
file_menu.add_command(label="Choice of the texts",
command=choice_text)
file_menu.add_command(label="Definition of subjects of the text-
1", command=array_text_1)
file_menu.add_command(label="Definition of subjects of the text-
2", command=array_text_2)
file_menu.add_command(label="Definition of subjects of the text-
3", command=array_text_3)
file_menu.add_command(label="Definition of the author of the
text", command=comor_text)
file_menu.add_command(label="Exit  from the program",
command=close_win)
lab =Label(tk, text="The text for comparison  author -K ",
font=("Arial", 12,  "bold "),foreground='red')
lab.pack()
txt= Text(tk, width=66,height=5,font=("Arial", 12,  "bold
"),foreground='red',wrap=WORD)
txt.pack()
lab1 = Label(tk, text="The test  author -K",font=("Arial", 12,
"bold "),foreground='green')
lab1.pack()
```

```
txt1= Text(tk, width=66,height=5,font=("Arial", 12,  "bold
"),foreground='green',wrap=WORD)
txt1.pack()
lab2 = Label(tk, text="The text  author-M", font=("Arial", 12,
"bold "),foreground='blue')
lab2.pack()
txt2= Text(tk, width=66,height=5,font=("Arial", 12,  "bold
"),foreground='blue',wrap=WORD)
txt2.pack()
lab3 = Label(tk, text="Text results of comparison",
font=("Arial", 12,  "bold"),foreground='black')
lab3.pack()
txt3= Text(tk, width=66,height=6,font=("Arial", 12,
"bold"),foreground='black',wrap=WORD)
txt3.pack()
lab4 = Label(tk, text="Minimum quantity of words in a window ",
font=("Arial", 12,  "bold"),foreground='black')
lab4.pack()
txt4= Text(tk, width=8,height=1,font=("Arial", 12,
"bold"),foreground='black',wrap=WORD)
wd=10
txt4.pack()
txt4.insert(END,wd)
lab5 = Label(tk, text="Maximum quantity of words in a window ",
font=("Arial", 12,  "bold"),foreground='black')
lab5.pack()
txt5= Text(tk, width=8,height=1,font=("Arial", 12,
"bold"),foreground='black',wrap=WORD)
wd=90
txt5.pack()
txt5.insert(END,wd)
tk.title('The analysis of the art text')
x = (tk.winfo_screenwidth() - tk.winfo_reqwidth())
/4#центрирование формы
y = (tk.winfo_screenheight() - tk.winfo_reqheight()) /
16#центрирование формы
```

```
tk.wm_geometry("+%d+%d" % (x, y))#центрирование формы
tk.mainloop()
```

ДОДАТОК Б

Слайди презентації

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

Напрям підготовки 6.050101 – комп'ютерні науки

Тема дипломного проекту:

«Засоби аналізу авторства тексту по частоті появи нових слів»

Студент: Михайлюченко О.І.
Керівник: Скарга-Бандурова І.С.

Сєвєродонецьк, 2018 р. 1

Рисунок Б.1 – Титульний слайд

Мета роботи:

- аналіз методів ідентифікації авторства текстів, що забезпечує підвищення точності визначення частоту появи нових слів, зменшення обсягу текстової вибірки і зниження тимчасових витрат на прийняття рішення, і створення програмного комплексу для ідентифікації авторства на її основі

2

Рисунок Б.2 – Мета роботи

1. Порівняння програмних засобів атрибуції текстів

Назва	Методи	Зміна пар-рів	Засоби аналізу тексту	Необхідний обсяг тексту, симв.	Точність	Реальні звадання
Лінгвоаналізатор	Ентропійний підхід, марківські ланцюги	Ні	Графем., стат. аналіз	40000-100000	84-89	Ні
Атрибутор	марківські ланцюги	Ні	Стат. аналіз	>20000	Не від.	Ні
СМАЛТ	Статистичні критерії, мережі Хеммінга	Ні	Графем., морф., синт., стат. аналіз	500 слів для визначення однорідності	Не від.	Так
Стилеаналізатор	марківські ланцюги	Так	Графем., стат. аналіз, робота с розміченими текстами	30 000-40 000	Не від.	Так
Авторознавець	Нейронні мережі, метод опорних векторів	Так	Графем., морф., стат. аналіз	100	90-98	Так

3

Рисунок Б.3 – Порівняння програмних засобів

Аналіз предметної області

Закони Зіпфа

- якщо помножити ймовірність виявлення слова в тексті на ранг частоти, то отримана величина (С) приблизно постійна:

$$C = \frac{\delta \cdot R}{n}$$

- де R - ранг частоти.
- За першим законом Зіпфа, якщо найпоширеніше слово зустрічається в тексті, наприклад, 100 раз, то наступне за частотою слово навряд чи зустрінеться 99 разів.
- Частота входження другого за популярністю слова, з високою часткою ймовірності, виявиться на рівні 50.

4

Рисунок Б.4 – Аналіз предметної області

2. Апроксимація статистичних даних для об'єктів, що підпорядковуються законам Зіпфа

- Найчастіше, для апроксимації статистичних даних для об'єктів, які підпорядкуватися Законами Зіпфа використовується гіперболічна функція виду

$$y = a + \frac{b}{a}$$

- де a, b – постійні коефіцієнти; x – статистичні дані аргументу функції (у вигляді списку); y – наближення значень функції до реальних даних отриманим методом найменших квадратів.

5

Рисунок Б.5 – Апроксимація статистичних даних по законам Зіпфа

Апроксимація статистичних даних для об'єктів, що підпорядковуються законам Зіпфа

- У нашому випадку вихідні дані задаються двома списками $x=[x_1,..x_n]$, $y=[y_1,..,y_n]$, де n – кількість даних у списках.
- Далі отримано функцію для визначення коефіцієнтів:

$$F(a,b) = \sum_{i=0}^{n} y_i - (a + b / x_i))^2$$

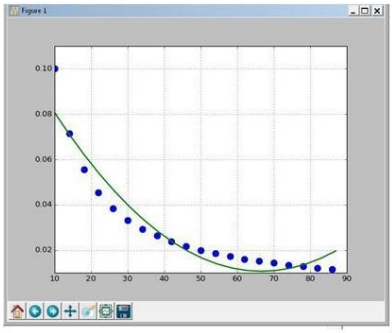
6

Рисунок Б.6 – Апроксимація статистичних даних по законам Зіпфа

Апроксимація даних поліномом

Код реалізації апроксимації даних поліномом

```
#!/usr/bin/python
# coding: utf8
import scipy as sp
import matplotlib.pyplot as plt
def mnkGP(x,y):
        d=2 # ступінь полінома
        fp, residuals, rank, sv, rcond = sp.polyfit(x, y, d, full=True) # Модель
        f = sp.poly1d(fp) # апроксмуюча функція
        print('Коефіцієнт -- a %s  '%round(fp[0],4))
        print('Коефіцієнт -- b %s  '%round(fp[1],4))
        print('Коефіцієнт -- c %s  '%round(fp[2],4))
        y1=[fp[0]*x[i]**2+fp[1]*x[i]+fp[2] for i in range(0,len(x))] # значення функції a*x**2+b*x+c
        so=round(sum([abs(y[i]-y1[i]) for i in range(0,len(x))])/(len(x)*sum(y))*100,4) # середня помилка
        print('Average quadratic deviation '+str(so))
        fx = sp.linspace(x[0], x[-1] + 1, len(x)) # можна встановити замість len(x) більше число для інтерполяції
        plt.plot(x, y, 'o', label='Original data', markersize=10)
        plt.plot(fx, f(fx), linewidth=2)
        plt.grid(True)
        plt.show()
x=[10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50, 54, 58, 62, 66, 70, 74, 78, 82, 86]
y=[0.1, 0.0714, 0.0556, 0.0455, 0.0385, 0.0333, 0.0294, 0.0263, 0.0238, 0.0217,
  0.02, 0.0185, 0.0172, 0.0161, 0.0152, 0.0143, 0.0135, 0.0128, 0.0122,
  0.0116] # дані для перевірки по функції y=1/x
mnkGP(x,y)
```



7

Рисунок Б.7 – Апроксимація даних поліномом

3. Проектування додатку для визначення авторства текстів з використанням мови Python

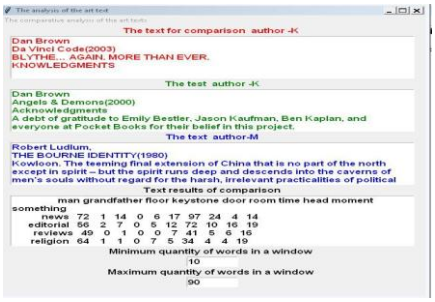
- Згідно технічного завдання в програмі реалізовано функції роздільного аналізу по першому і другому закону Зіпфа кожного документа.

- Для перевірки роботи програми використовувалися твори відомих англомовних письменників. У якості приклада проведено порівняльний аналіз творів Ден Браун «Код Давінчі» і «Ангели і демони» і Роберт Ладлема «Ідентифікація борна».

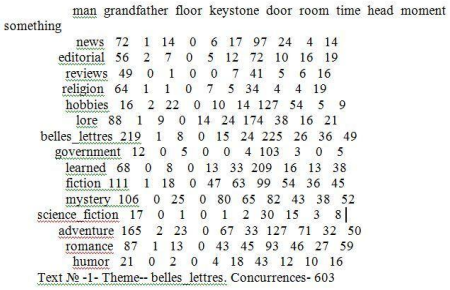
8

Рисунок Б.8 – Проектування додатку

Інтерфейс програми та експерименти



Аналіз творів на авторство

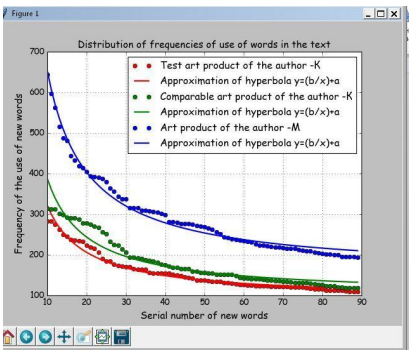


Результат визначення жанру твору

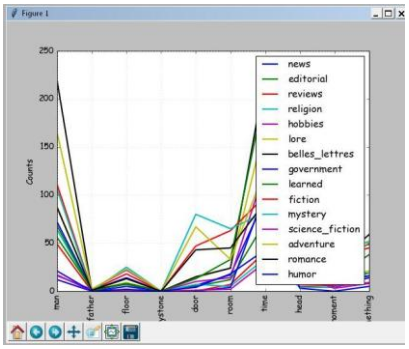
9

Рисунок Б.9 – Інтерфейс програми та експерименти

Інтерфейс програми та експерименти



Розподіл частот використання слів в тексті



Результат в графічному вигляді

Для розглянутого випадку середня відстань між апроксимуючими гіперболами автора K становить 25,062, а між першим твором автора K і твором автора M - 138,25.

10

Рисунок Б.10 – Інтерфейс програми та експерименти

Висновки

- Об'єктом розробки даного дипломного проекту є додаток для ОС Windows – програма для визначення авторства тексту по частоті появи нових слів з використанням мови Python.
- Були поставлені та вирішуються наступні завдання:
  1) Вибір методу і програмних засобів для розробки програми.
  2) Виконано порівняльний аналіз аналогічних програм.
  3) Розглянута та проаналізована високорівнева мова програмування Python.
  4) Обґрунтована актуальність створення додатка на ОС Windows.
  5) Проведено розробку додатка.
- Додаток більше зорієнтований для викладачів, студентів та школярів.
- Були сформульовані основні задачі та описані методи та засоби, які були обрані для реалізації додатку. Також були враховані недоліки аналогічних додатків які були розглянені. Додаток аналізатор був створений відповідно до технічного завдання на розробку та протестований.

11

Рисунок Б.11 – Висновки