

## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Метод автоматизованого виправлення граматичних помилок у цифрових текстах

Галузь знань 12 – Інформаційні технології  
Шифр і назва галузі знань  
Спеціальність 122 – Комп'ютерні науки  
Шифр і назва спеціальності  
Освітня програма Комп'ютерні науки  
Назва освітньої програми

Виконав: студент 4 курсу, група КН-18-1 П.В.о В.О. Підлапушиний  
Курс, група виконавця Підпис Ініціали, прізвище  
Керівник: викладач кафедри КН О.Собко О.В. Собко  
Науковий ступінь, посада Підпис Ініціали, прізвище  
Нормоконтроль: к.т.н., доцент кафедри КН Р.О. Р.О. Багрій  
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:


Зав. кафедри КН, д.т.н., професор

13 червня 2022 р.

О.В. Бармак  
Підпис Ініціали, прізвище

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
Факультет інформаційних технологій  
Кафедра комп'ютерних наук  
Освітній ступінь бакалавр  
Галузь знань 12 – Інформаційні технології  
Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ  
Завідувач кафедри комп'ютерних наук

  
(підпис)  
д.т.н., професор О.В. Бармак  
« 25 » березня 2022 року

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «Метод автоматизованого виправлення граматичних помилок у цифрових текстах»

2. Завдання видано студенту Підлапушному Владиславу Олеговичу  
(прізвище, ім'я, по батькові)

3. Керівник роботи викладач кафедри КН Собко Олена Віталіївна  
(посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від « 01 » березня 2022 р. № 18

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

Метою кваліфікаційної роботи бакалавра є створення методу автоматизованого виправлення граматичних помилок у цифрових текстах. Також слід виконати розробку системи автоматизованого виправлення граматичних помилок у цифрових текстах. Розроблювана система має здійснювати обробку текстів та визначати в них граматичні помилки. Також система має містити візуальне відображення місць в тексті, що містять помилки, та виконувати підбір пропозицій щодо виправлень.

Виконавець: студент 4 курсу, група КН-18-1 ПВ В.О. Підлапушний  
Курс, група виконавця Підпис Ініціали, прізвище

Керівник: викладач кафедри КН О. Собко О.В. Собко  
Науковий ступінь, посада Підпис Ініціали, прізвище

## Анотація

Тема кваліфікаційної роботи бакалавра: «Метод автоматизованого виправлення граматичних помилок у цифрових текстах»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-18-1 Підланушиний Владислав Олегович

Керівник кваліфікаційної роботи бакалавра: викладач кафедри КН Собко Олена Віталіївна

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
66	28	10	37	4

Метою кваліфікаційної роботи бакалавра є створення та прикладна програмна реалізація методу автоматизованого виправлення граматичних помилок у цифрових текстах. Для розробки інформаційної системи було використано мову програмування C#, систему керування базами даних MS SQL Server, платформу .NET.

Розроблена система призначена для користувачів, що мають необхідність швидкого виправлення граматичних помилок в текстах, реалізований програмний продукт може бути особливо корисним для осіб, що створюють тексти для поширення в соціальних мережах, статті, рецензії, тощо. Інструментами для практичного використання системи є: обробка текстів та визначення в них граматичних помилок, візуальне відображення місць в тексті, що містять помилки, підбір пропозицій щодо виправлень та відсоткове значення помилок в тексті.

Ключові слова: семантика, ключові слова, n-грами, аналіз тексту.

Виконавець: студент 4 курсу, група КН-18-1

Курс, група виконавця

  
Підпис

В.О. Підланушиний  
Ініціали, прізвище

## Зміст

Перелік скорочень .....	3
Вступ.....	4
Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій.....	5
1.1 Аналіз інформаційних моделей.....	5
1.2 Огляд теоретичних підходів до розв’язку подібних задач .....	9
1.3 Аналіз існуючих програмних рішень.....	11
1.4 Аналіз сучасних засобів створення програмного забезпечення .....	17
1.5 Мета, задачі та вимоги до реалізації інформаційної системи .....	22
Розділ 2 Проектування інформаційної системи .....	24
2.1 Метод автоматизованого виправлення граматичних помилок у цифрових текстах .....	24
2.2 Інформаційна структура програмного застосунку для автоматизованого виправлення граматичних помилок .....	26
2.2.1 Проектна архітектура системи та взаємозв’язок компонентів.....	26
2.2.2 Інформаційна модель.....	29
2.3 Вибір засобів розробки інформаційної системи .....	34
2.3.1 Вибір мови програмування .....	35
2.3.2 Вибір редактора програмного коду.....	36
2.3.3 Вибір СКБД .....	38
Розділ 3 Програмна реалізація інформаційної системи .....	39
3.1 Структура та функціональне призначення програмних складових системи.....	39
3.2 Особливості реалізації програмних складових системи .....	40
3.3 Тестування інформаційної системи .....	46
3.4 Інструкція користувача.....	53
Висновки .....	61
Перелік посилань.....	63
Додатки	

**Перелік скорочень**

<b>Скорочення, термін, позначення</b>	<b>Пояснення</b>
БД	База даних
ІС	Інформаційна система
ІТ	Інформаційні технології
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки
ПЗ	Пояснювальна записка
ПП	Програмний продукт
СКБД	Система керування базами даних
ХНУ	Хмельницький національний університет.
JVM	Java Virtual Machine

## Вступ

**Мета кваліфікаційної роботи бакалавра** – створення та прикладна програмна реалізація методу автоматизованого виправлення граматичних помилок у цифрових текстах.

**Об'єкт дослідження** – процес створення і редагування цифрових текстів.

**Предмет дослідження** – інформаційні технології, моделі, методи та засоби для автоматизованого виправлення граматичних помилок у цифрових текстах.

З широким поширенням текстової інформації у вигляді цифрових документів особливо актуальним є перевірка та виправлення граматичних помилок з метою створення якісного контенту. Адже текст, в якому є помилки здатен відштовхувати читача та знижувати загальну затребуваність автора таких текстів.

Для створення цифрових документів, що не містять граматичних помилок, автор повинен володіти відповідними знаннями – правилами написання слів, їх зв'язку у реченнях, доцільного використання термінів, що належать до тих чи інших стилів мовлення.

При створенні цифрових текстів автори можуть допускати не тільки граматичні помилки, а й стилістичні, лексичні, пунктуаційні. Допомагають знаходити та виправляти такі помилки редактори та коректори. Вони можуть проводити різного рівня редагування – від пошуку граматичних помилок, до стилістичних. Дана робота є доволі трудомістка, та вимагає великої бази знань для коректного виправлення помилок у цифрових текстах. Для проведення даної роботи існує ряд програмних застосунків, які дозволяють робити це автоматизовано. Кожен з них має свої переваги та недоліки, тому повністю покладатися на правильність роботи цих застосунків не варто.

Таким чином має сенс проведення дослідження щодо існуючих методів та засобів для автоматизованого виправлення граматичних помилок у цифрових текстах, а також розробка власного методу, що вирішує дану задачу.

## **Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій**

### **1.1 Аналіз інформаційних моделей**

З розвитком інформаційних технологій все більше інформації, яка нас оточує перетворюється в цифровий вигляд. Музика, фотографії, фільми, література – все масово зберігається і відтворюється із застосуванням електронних пристроїв. Особливо великі обсяги інформації зберігаються в цифрових текстах – це літературні твори, новини, статі, підручники, різного роду книги. Особливо великі об'єми інформації зберігаються тільки у цифровому вигляді. До прикладу, існує таке поняття як «цифрова поезія», яке має на увазі існування літературних творів тільки у цифровому вигляді [1].

Цифровий текст – це такий текст, який створено для читання на різного роду пристроях таких як смартфони, планшети, комп'ютери. Такі тексти можуть зберігатися на різних носіях – жорсткий диск, флеш-пам'ять, CD-диски.

Варто зазначити, що все більше людей віддає перевагу саме цифровому тексту, адже він легкий у сприйманні, доступний для будь-якої грамотної людини [2]. Вимоги до таких текстів також є. Цифрові тексти повинні розкривати повною мірою суть того, про що описується в них, мати певну структуру та звісно бути правильно написані.

Під правильністю написання мається на увазі відповідність написаних слів, словосполучень та речень правилам граматики, лексики, орфографії тощо.

Слово – найменша одиниця мови, яка є самостійною одиницею мови і виконує функцію позначення якогось явища, події, об'єкту, ознаки, дії, тобто їх називання [3]. Слова об'єднуються в словосполучення, а ті в свою чергу в речення.

Словосполучення – це об'єднання двох або більше слів, які пов'язані між собою граматичним зв'язком або за змістом. В словосполученнях завжди існує два типи слів – головне та залежні. Словосполучення відіграють роль конкретизації, детальнішого опису предмету, явища, дії тощо [4].

Речення – це об'єднання одного або декількох слів, які виражають закінчену думку. За допомогою речень люди формулюють свої думки і вони є основним способом спілкування. Слова у реченні обов'язково повинні бути пов'язані граматично та за змістом [5]. Речення бувають різних типів:

- Розповідні – містить розповідь про якісь факти, об'єкти чи події.
- Питальні – речення, що спонукає людину до отримання відповіді на запитання.
- Спонукальні – речення, які людину спонукають до певної дії.

Також речення бувають прості та складні. Простими називаються такі речення, які складаються з однієї граматичної основи, а складні – з двох і більше [5].

Грамматика – це сукупність правил для окремої мови, за якими слова об'єднуються в змістовні речення [6]. Іншими словами, грамматика, як наука вивчає будову мови. Центральною одиницею граматики є слово.

Лексика – це склад мови, сукупність слів, що описують певну сферу спілкування або стиль мови. За допомогою сукупності певних слів можна описувати тексти за наступними стилями [7]:

- Науковий – використовується для написання різного роду підручників, статей, насичений науковими термінами.
- Офіційно-діловий – використовується в діловодстві.
- Художній – призначений для написання літературних творів.
- Розмовний – стиль повсякденних розмов, усного спілкування.
- Публіцистичний – використовується для написання статей у журналах, газетах, а також використовується на телебаченні.
- Сакральний – стиль, що використовується у релігійній сфері життя людей.

Орфографія – це система правил написання слів, що є загальноприйнята для певної мови. Орфографія відповідає за правильність написання слів, їх перенесення з одного рядка в інший, правильного написання слів при їх відмінюванні тощо [8].

Для перевірки правильності написання слів використовують орфографічні словники. Вони містять перелік слів у нормативному написанні, тобто такому, що склався в певний історичний період та признається суспільством як правильним [9]. Якщо раніше словники існували тільки у паперовому вигляді, то сьогодні створено багато електронних словників, до яких мають доступ усі охочі користувачі. Словники також поділяються на категорії, наприклад, шкільні, загальні, галузеві тощо. Такі словники містять певний набір слів, що орієнтований на певну аудиторію.

На сьогоднішній день доступна велика кількість українських орфографічних словників і у електронній формі. Також варто зазначити, що існують і інші види словників, такі як:

- синонімічні;
- фразеологічні;
- орфоепічні;
- тощо.

Словники створені з метою уникнення помилок у написанні текстів, а також окремих слів, так як помилки бувають різних типів [10]:

- граматичні – вжите слово у неправильній чи спотвореній формі;
- лексичні – слово вжито у значенні, що йому не притаманно;
- пунктуаційні – невірні розставлені розділові знаки;
- стилістичні – порушено стиль викладу тексту.

В свою чергу граматичні помилки поділяються на:

- морфемні;
- морфологічні;
- синтаксичні;
- словотвірні;
- орфографічні.

Варто зазначити, що деякі помилки, які умовно можна назвати мовними, виявляються тільки в контексті, на відміну від граматичних помилок, для яких

не потрібен контекст, так як помилка може бути присутня в середині самого слова [11].

Для ефективного виправлення орфографічних помилок у текстах необхідним є виконання наступних завдань [12]:

- постійне повторення сучасних підходів до написання слів;
- виявлення орфографічних помилок та зіставлення помилки з відповідним правилом написання;
- обґрунтування вибору написання слова;
- вибір зі словників слів на відповідні орфографічні правила, що підтверджують правильність його написання;
- вибір з існуючих текстів слів, правильність написання яких порушено;
- редагування помилково написаних слів як в усній, так і в письмовій формі;
- повний орфографічний розбір тих слів, в яких були виявлені помилки.

Якщо навести приклад конкретної професії, в обов'язки якої входить виправлення помилок, а саме редактора, то для його ефективної роботи необхідним є знання не тільки правил граматики, а й предметної області, а також уміння не тільки виправляти помилки, а й покращувати, оптимізувати текст [13]. Це пов'язано з тим, що при написанні текстів автори можуть допускати банальні помилки у датах, цифрах, як от обдруківки. Задача редактора виявити такого роду помилки та виправити їх, не порушивши суті написаного автором.

Редактор може виконувати різного роду перевірки написаного – від лише перевірки граматичних помилок, до перевірки стилістичних, лексичних помилок. При чому перевірка останніх потребує від редактора постійної перевірки себе і автора у правильності викладеного матеріалу, звірка з джерелами [13].

Можна стверджувати, що процес виявлення та виправлення помилок доволі складний, адже робота над текстом може бути різної глибини – від перевірки граматичних помилок, і до перевірки стилістичних та лексичних помилок. Це потребує від перевіряючого великого обсягу знань – як базових правил написання слів, так і поглиблених. Тому особливо актуальним є

автоматизація процесу виявлення помилок у цифрових текстах для спрощення роботи людини. З іншої сторони можна стверджувати, що цей процес не можна цілком довірити машині, він повинен контролюватися людиною, адже машина не завжди може враховувати особливості правопису слів певної мови.

## 1.2 Огляд теоретичних підходів до розв'язку подібних задач

Системи, які займаються перевіркою орфографії – це допоміжні програми, що шукають в цифрових текстах слова, які мають помилки. Для розв'язку задачі виявлення та виправлення помилок у цифрових текстах існує декілька підходів, проте всі вони мають схожий принцип:

- виокремлення слів, з яких складається текст;
- порівняння кожного слова зі словником;
- додаткові дії, наприклад приведення слова у називну форму однини для виявлення збігів зі словником.

Існують і іншого роду системи перевірки орфографії, у яких для виявлення помилок використовується підхід побудови N-грам.

N-грамою називається послідовність з N елементів [14]. Цими елементами можуть бути звуки, букви, склади, слова. N-грами можуть мати різну кількість елементів, найбільш розповсюджена кількість елементів, яку використовуються для реалізації різноманітних задач, це 2-4 елементи. Відповідно до кількості елементів N-грами мають відповідні назви [14]:

- уніграми – один елемент;
- біграми – два елементи в послідовності;
- триграми – три елементи в послідовності;
- тетраграми – чотири елементи в послідовності;
- тощо.

Побудова N-грам відбувається за принципом об'єднання N елементів з послідуочим зміщенням на один елемент для побудови наступної N-грами.

Наприклад, у реченні «Густа тінь у воротах повітки здавалась чорною» можна побудувати наступні біграми:

- густа тінь;
- тінь у;
- у воротах;
- воротах повітки;
- повітки здавалась;
- здавалась чорною.

Відповідно тетраграми можна виділити наступні:

- густа тінь у;
- тінь у воротах;
- у воротах повітки;
- воротах повітки здавалась;
- повітки здавалась чорною.

N-грами бувають двох типів – лінійні, як приклад, що наведений вище та синтаксичні. Синтаксичні визначаються шляхами в деревовидній структурі синтаксичних залежностей. Для більшості задач побудова синтаксичних N-грам дає кращі результати, ніж лінійні [14].

N-грами використовуються для багатьох задач. В основному це задачі, пов'язані з обробкою природної мови. Найбільше застосування вони знаходять в задачах прогнозування наступного елемента послідовності, а також в задачах пошуку схожості текстів. У статті [15] автори пропонують для визначення подібності рядків застосовувати N-грами не для групи слів, а для групи символів, що в свою чергу дозволяє ефективніше оцінити синтаксичну подібність рядків відповідно до еталону.

У [16] автор пропонує використовувати побудову N-грам для визначення ймовірності використання певного слова чи словосполучення в задачі прогнозування слів для альтернативної комунікації.

У [17] в задачі виявлення настроїв у текстах соцмереж автори розглядають застосування класифікатора на базі юніграм або N-грам при машинному навчанні.

Також N-грами знаходять своє застосування у більш специфічних задачах. Наприклад, автори [18] розглядають застосування N-грам, як один із методів для виявлення шкідливого програмного забезпечення в комп'ютерних системах.

Іноді в задачах обробки природньої мови є необхідним пошук ключових слів та словосполучень, які в сукупності представляють сенс написаного тексту. Для пошуку ключових слів використовую наступні методи:

- дисперсна оцінка;
- Text Rank;
- TF-IDF;
- Keyphrase Extraction Algorithm;
- частотний аналіз.

Також багато авторів пропонують різні модифікації вищеперерахованих методів пошуку ключових слів, що покращують їхню ефективність.

З вище написаного можна зробити висновок про те, що підходи з побудовою N-грам є популярними у різного роду задачах, не лише для виявлення та виправлення помилок у цифрових текстах.

### **1.3 Аналіз існуючих програмних рішень**

Задача пошуку помилок у текстах є доволі популярною, тому для її вирішення існує багато готових рішень. У інтернеті можна знайти спеціальні сервіси для перевірки текстів на помилки. Також таку функцію можна знайти у деяких текстових редакторах. Далі розглянуто приклади таких застосунків.

LanguageTool – онлайн сервіс для перевірки граматики, стилю та орфографії текстів [19]. Сервіс надає свої послуги безкоштовно, проте зареєструвавшись, користувач можуть отримати ряд переваг, а саме:

- користувацькі налаштування інтерфейсу;
- створення персонального словнику;
- отримання рекомендацій щодо написання;
- можливість отримання оцінки текстів.

Також великою перевагою даного сервісу є те, що користувач може встановити розширення для свого браузера, а також інтегрувати даний інструмент у Google-документи.

Під час процесу перевірки система виділяє слова, які містять помилки та клацнувши на них, користувач може переглянути варіанти правильного написання (рисунок 1.1)

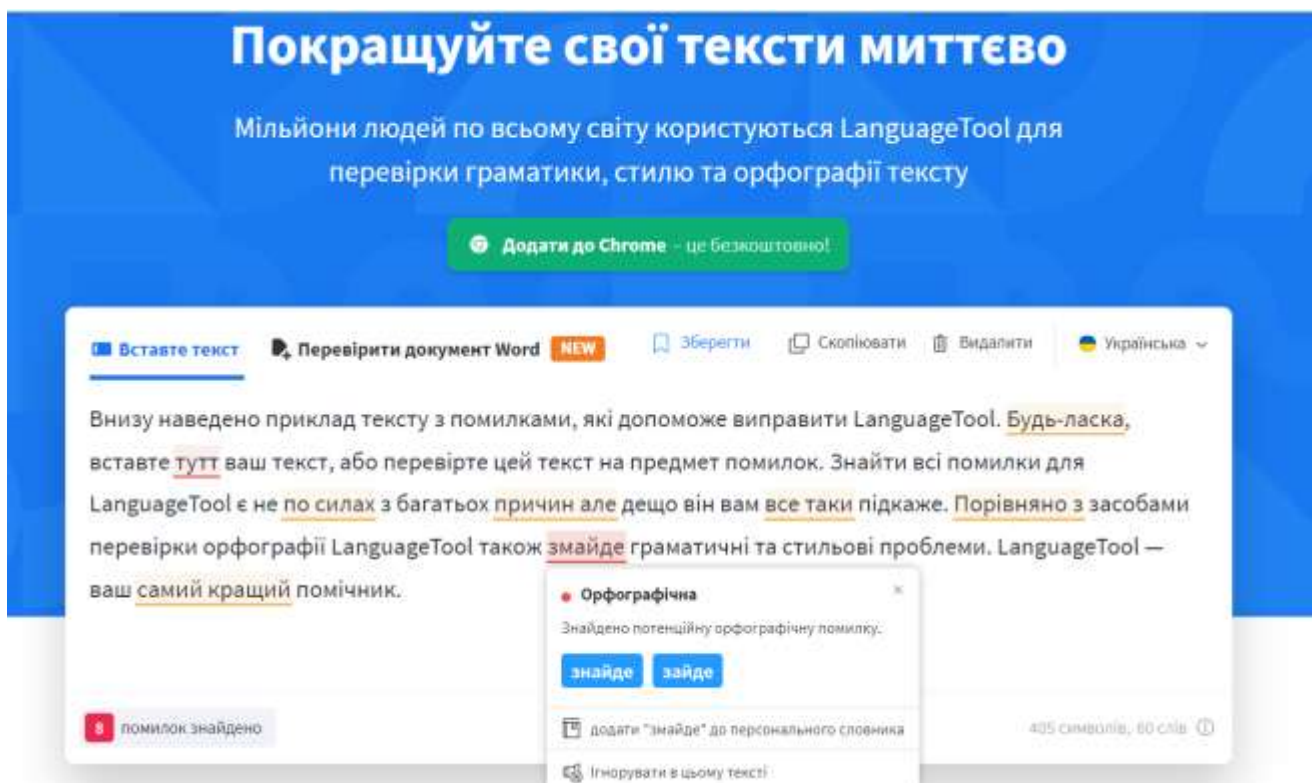


Рисунок 1.1 – Сервіс перевірки тексту на помилки LanguageTool [19]

Сервіс LanguageTool інтегрований також як основний інструмент для перевірки правопису і на інших сайтах, що позиціонують себе як сервіси виправлення помилок.

Сервіс перевірки орфографії Офіційного сайту Української мови – призначений для перевірки лише орфографічних помилок у текстах [20].

Підтримує тільки українську мову. Приклад виявлення помилок за допомогою цього сервісу зображено на рисунку 1.2.

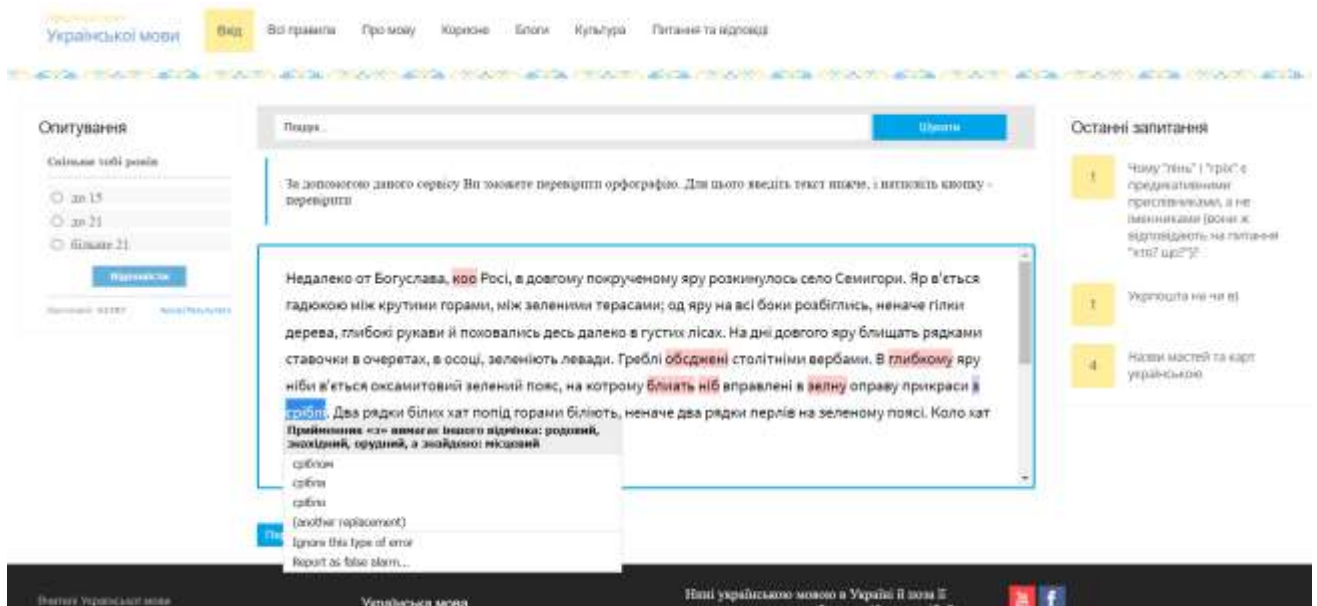


Рисунок 1.2 – Сервіс перевірки орфографії Офіційного сайту Української мови [20]

Перевагою даного сервісу є те, що для кожної помилки система надає пояснення чому виявлено помилку і варіанти її виправлення. Також можна додати слово в словник або надіслати рапорт розробникам, що помилку було виявлено невірно. Недоліком можна назвати те, що підтримується лише одна мова. Також відсутній кабінет користувача, де можна було би переглянути статистику чи налаштувати словники.

Сервіс перевірки орфографії сайту Мова – ДНК нації – простий сервіс перевірки орфографії, який перевіряє орфографічні, граматичні і лексичні помилки, в основі даного сервісу використовується інструмент LanguageTool [21]. Даний сервіс, як і йому подібні, підсвічує потенційні помилки та пропонує варіанти заміни (рисунок 1.3).

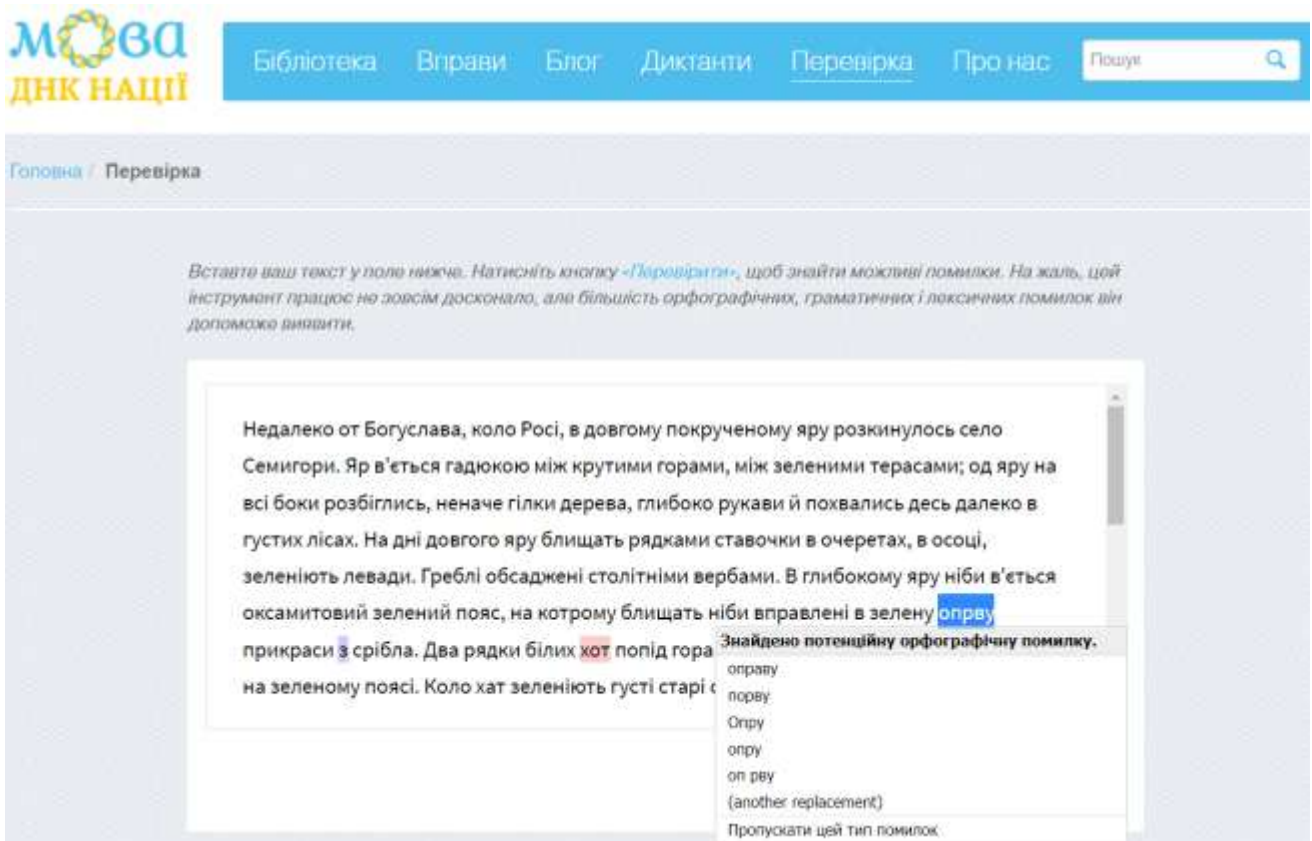


Рисунок 1.3 – Сервіс перевірки орфографії сайту Мова – ДНК нації [21]

Недоліком даного сервісу є те, що він також не має кабінету користувач, що спрощувало б подальшу роботу з текстами.

OnlineCorrector – розширення для браузера Google Chrome, що дозволяє виправляти помилки у Google Docs, що написані українською мовою [22]. Має доволі широкий асортимент можливостей: знаходить та виправляє граматичні помилки, лексичні, а також стилістичні помилки, видаляє лишні пробіли, пропонує чергування сполучників та виправляє поширені помилки написання слів через дефіс тощо. Також є можливість встановити розширення для Microsoft Office Word. Приклад виправлення помилок зображено на рисунку 1.4.

Недоліком даного розширення є те, що воно орієнтоване на обмежену кількість програм та не є безкоштовним.

Вбудованою функцією виправлення помилок у текстах оснащений текстовий редактор Microsoft Office Word [23]. Цікавою особливістю є те, що перевірка виконується автоматично тією мовою, якій відповідає розкладка клавіатури користувача в момент введення тексту. Якщо користувач вводить



Перевагами даної функції є те, що можна додавати слова у словник, щоб в подальшому вони не вважалися помилково написаними. Також функція перевірки правопису перевіряє узгодженість форм слів. В редакторі підтримується велика кількість мов для виправлення помилок.

Потужною онлайн-платформою для перевірки правильності написаних англійською мовою текстів є Grammarly. Унікальність цієї платформи полягає в тому, що вона не лише перевіряє граматику, а й надає рекомендації щодо чіткого та стислого викладу інформації, забарвлення її у відповідний емоційний тон, як от впевненість, ввічливість тощо [24].

Grammarly доступна на різних пристроях, як от настільні комп'ютера та мобільні телефони, вона здатна інтегруватися з браузерами та сервісами Google. Інтеграція з Microsoft Edge показано на рисунку 1.6.

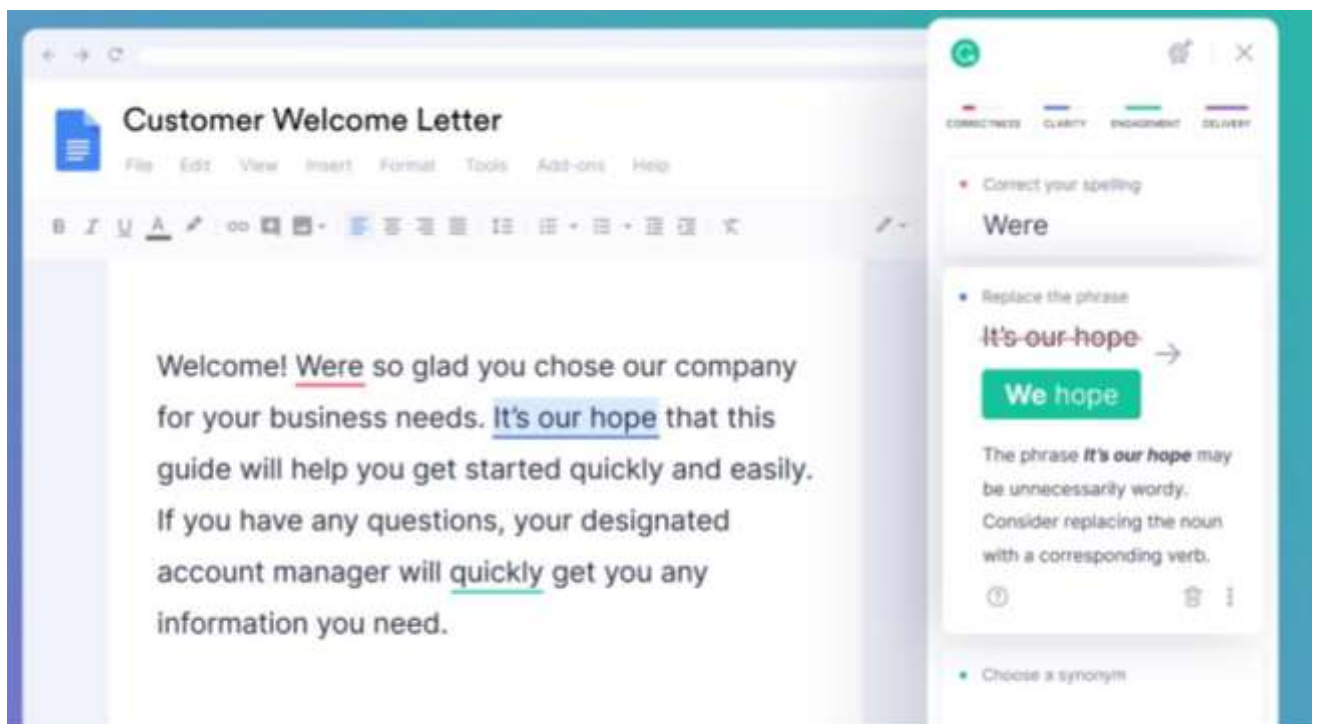


Рисунок 1.6 – Інтеграція Grammarly з Microsoft Edge [25]

Придбавши ліцензію Grammarly рівень написання текстів переходить на новий рівень. Крім виправлення помилок, Grammarly пояснює тип помилки. Grammarly аналізує весь текст і може пропонувати альтернативні слова на заміну до найбільш повторюваних у тексті. У деякому сенсі дана платформа може

цілком замінити людську роботу редактора чи коректора тексту [26]. Недоліком даної платформи є підтримка лише однієї мови.

Отже, як видно з проведеного аналізу існуючих програмних рішень, для користувачів доступні як найпростіші сервіси виявлення помилок, так і цілі платформи на базі штучного інтелекту, що не лише виявляють та виправляють помилки, а й здатні обґрунтувати виявлену помилку, пояснити тип помилки та підібрати правильні варіанти слова, а також вміють змістовно будувати текст відповідно до вимог користувача.

#### **1.4 Аналіз сучасних засобів створення програмного забезпечення**

Сьогодні ринок електронних пристроїв є доволі різноманітним. Одна людина може володіти одразу декількома пристроями, які надають їй чимало різних можливостей. Найбільшу частину сучасних електронних пристроїв на сьогодні складають персональні комп'ютери та смартфони. При чому іноді смартфони можуть замінити деякі функції комп'ютерів. Відповідно до цих типів пристроїв програмісти розробляють різного типу програмне забезпечення, як от:

- застосунки для смартфонів та планшетів на платформах Android та iOS;
- веб-застосунки, що є універсальними для будь-якого типу електронних пристроїв, проте мають свої особливості розробки для смартфонів та персональних комп'ютерів;
- віконні застосунки для персональних комп'ютерів на базі операційних систем Windows, MacOS, Linux тощо.

Далі розглянуто переваги та недоліки кожного з типів застосунків.

В усьому світі близько 4 мільярдів активних користувачів смартфонів. У 2021 році 65% всіх відвідувань Інтернету в США було здійснено з мобільних пристроїв.

Мобільний застосунок – це програма, яка завантажується та встановлюється на смартфон. Застосунки розробляються для певних операційних систем, таких як Android, Windows або iOS, і їх можна звантажити

на таких платформах, як Google Play Store або Apple App Store відповідно. Вони працюють на смартфоні та беруть свої дані з мережі, але часто можуть зберігати весь вміст локально, та працювати без підключення до мережі [27].

Застосунки для смартфонів розроблені для оптимальної роботи на певних операційних системах. Їх структура, користувальницький інтерфейс та функції оптимізовані для забезпечення максимально зручного користування. Відповідно, це є однією з найбільших переваг мобільних застосунків. Вони завантажуються швидше і в них легше перемикається з одного застосунка на інший. Також можна ефективно використовувати власні функції смартфона, такі як GPS, камери, телефонія, для цілей програми.

Користувачі можуть персоналізувати застосунок. Можна змінювати налаштування та зберігати особисті дані, наприклад, платіжні дані, щоб зробити покупки в Інтернеті зручнішими. Додатки також мають push-сповіщення, що є доволі цінним інструментом для того, щоб залучити більшу кількість користувачів [27]. Розробники можуть налаштувати сповіщення для того, щоб інформувати користувачів про спеціальні пропозиції, нові функції та оновлення. Ці форми персоналізації є великою перевагою порівняно з іншими типами застосунків.

Також слід врахувати той факт, що програму можна використовувати без доступу до мережі, необхідні дані можуть бути завантажені пізніше, при підключенні смартфона до мережі.

З мінусів можна відзначити те, що відсутність сумісності може негативно вплинути на асортимент і вартість застосунків. Кожна з вищеперерахованих операційних систем має власні вимоги, тому необхідно створювати різні версії програми, щоб охопити користувачів з різними смартфонами. Кожну версію необхідно оновлювати, а користувачі повинні бути в курсі того, що виходять нові версії для того, щоб і них був стимул, оновити застосунок і користуватися ним далі [27].

Веб-застосунки – це програмне забезпечення, яке не потрібно встановлювати на комп'ютері чи смартфоні, воно доступне через Інтернет.

Таким чином, можна значно спростити свою роботу, перебуваючи в будь-якому місці і не прив'язуватися до робочого місця [28].

Перевагами на практичному рівні є те, щоб отримати доступ до нього, не потрібно завантажувати будь-яке програмне забезпечення, а оновлення доступні без необхідності встановлення або оновлення. Використання застосунку не залежить від операційної системи, що встановлена на пристрої користувача.

Ще однією перевагою є безпека, тому що дані розташовані на сервері, доступ до якого можна отримати за допомогою Інтернету і доступний через можна отримати через вбудований браузер. Оскільки сервер є зовнішнім, користувачі не ризикують бути атакованими вірусами [29].

Великою перевагою є надзвичайна гнучкість та простота – користувач може взаємодіяти з будь-якого місця, не обмежуючись лише одним робочим місцем.

Великим недоліком веб-застосунків є те, що їх не можна використовувати без підключення до Інтернету. Для деяких функцій необхідно отримати інформацію та дані з серверів, на яких вони знаходяться [29].

Віконні застосунки – це відомий і найдавніший варіант роботи з персональним комп'ютером. На сьогоднішній день все ще велика кількість застосунків завантажуються на комп'ютер, наприклад Microsoft Office. І надають найбільш широкий та зручний для роботи функціонал. Серед іншого вони також формують наше сучасне розуміння програмного забезпечення.

Ось деякі переваги віконних застосунків. Після встановлення на відповідному пристрої для доступу до програми не потрібне постійне підключення до Інтернету, на відміну від веб-застосунків та мобільних застосунків. Дані користувача зберігаються локально на комп'ютері, тому є майже фактично контроль над програмою та інформацією, з якою вона працює. Великою перевагою віконних застосунків є те, що спеціалізовані застосунки дозволяють легко обробляти складні процеси [30].

Недоліки також очевидні. Віконні застосунки потрібно оновлювати вручну через певні проміжки часу, що може бути складно для користувачів з

низьким рівнем володіння персональним комп'ютером. Також важливим є зберігати резервні копії персональних даних, оскільки вся інформація зберігається на жорсткому диску, і якщо комп'ютер ламається, то деякі дані можуть бути втрачені. Також недоліком є те, що для роботи з застосунком необхідним є доступність в першу чергу самого пристрою [30].

Отже зважаючи на всі переваги та недоліки доцільним є розробка саме віконного застосунку, адже він надасть можливість зручної роботи та зберігання великих об'ємі інформації.

Для розробки віконних застосунків створено спеціального роду програмні платформи для спрощення розробки. На сьогодні найпопулярнішими платформами є Java та .NET.

Java – розроблена Джеймсом Гослінгом (Sun Microsystems, тепер Oracle) у 1995 році, є об'єктно-орієнтованою мовою загального призначення [31].

Одноименною є і платформа, яка використовується для програмування мовою Java. Відомою парадигмою та девізом платформи Java є «Напиши один раз, запускай будь-де». Цей девіз відображає відображає гнучкість та переносимість платформи та самої мови, яка за своєю суттю не залежить від операційної системи користувача.

Java найвідоміша завдяки своїй портативності та незалежності від платформи. Оскільки код Java скомпільовано у байт-код, його можна виконувати на будь-якій комп'ютері чи операційній системі без зміни синтаксису, достатньо встановити Virtual Machine [31].

Продуктивність Java є складною темою для дискусій серед програмістів. Наприклад, порівняно з мовою C# та платформою .NET, Java має довший час відгуку та створює більші навантаження на процесор. Справедливо відзначити, що в поєднанні з компіляторами Just-In-Time Java все ж таки забезпечує високу продуктивність. Java має багатопотокову архітектуру, що дозволяє виконувати декілька процесів одночасно.

Java має доволі високий рівень безпеки. Оскільки вона статично типізована, то це зменшує кількість помилок, пов'язаних із типом змінних.

Уразливості Java походять здебільшого через інтегровані бібліотеки та програми.

Платформа Java та сама мова найкраще підходить для [31]:

- розробки веб-застосунків;
- створення наукових програм та програмного забезпечення;
- створення програм для ОС Android;
- створення хмарних застосунків.

.NET Framework, або просто .NET – це інструмент розробки програмного забезпечення, створений Microsoft для власної системи продуктів і послуг. Система, яка охоплює сімейство платформ для розробки .NET, є доволі велика. Перевагою даної платформи є те, що по суті це служба з відкритим кодом, якою можуть користуватися інші розробники [32].

Платформа .NET – це сімейство фреймворків та інструментів розробки програмного забезпечення, які дозволяють створювати широкий спектр функцій і послуг, а також типів застосунків.

Хоча деякі інші фреймворки є фреймворками з відкритим кодом, проте багато хто вважає за краще використовувати .NET через переваги, які він пропонує.

Однією з найкращих переваг .NET є те, що він заснована на об'єктно-орієнтованому програмуванні. Це дозволяє поділяти розроблюване програмне забезпечення на менші частини, що потім дозволяє розробникам працювати над ними окремо. Після того, як робота над певними частинами виконана, вони можуть переходити до наступних. Коли всі частини програмного застосунку готові, їх можна об'єднати та ефективніше керувати ними [32].

Простота системи кешування .NET робить його надійним для тимчасового зберігання даних. Більше того, він дозволяє налаштувати реалізацію кешу, щоб розробники могли підвищити продуктивність та масштабувати за потреби.

.NET має кросплатформенну реалізацію, яку можна використовувати для розробки додатків на кількох операційних системах. Для розробників, які

віддають перевагу різним мовам програмування, це зовсім не буде проблемою, адже платформа підтримує розробку на декількох мовах програмування.

Недоліками даної платформи є те, що вона знаходиться під управлінням Microsoft, тому будь-які зміни чи обмеження, які може накласти компанія, вплинуть на розробки, що були виконані в рамках цієї платформи. Професійна розробка застосунків за допомогою цієї платформи вимагає покупку ліцензії, що є доволі дороговартісною. Також при оновленні платформи не завжди вчасно випускається оновлена документація до неї [32].

Отже, можна зробити висновки про те, що кожна з проаналізованих платформ має свої переваги та недоліки, проте .NET має більше переваг та буде більш доцільнішим використання саме цієї платформи для розробки віконного застосунку.

### **1.5 Мета, задачі та вимоги до реалізації інформаційної системи**

Метою кваліфікаційної роботи бакалавра є створення та прикладна програмна реалізація методу автоматизованого виправлення граматичних помилок у цифрових текстах. Для досягнення мети потрібно вирішити наступні задачі:

1. Провести аналіз предметної області, зокрема виконати огляд теоретичних підходів до розв'язання задачі виправлення граматичних помилок у цифрових текстах, а також провести аналіз існуючих програмних рішень.

2. Створити метод автоматизованого виправлення граматичних помилок у цифрових текстах, на базі якого створити інформаційну структуру програмного застосунку.

3. На базі проведеного аналізу існуючих засобів створення програмного забезпечення обрати комбінацію засобів для розробки програмної реалізації методу автоматизованого виправлення граматичних помилок у цифрових текстах.

4. Обраними засобами розробки створити та протестувати прикладну програмну реалізацію методу автоматизованого виправлення граматичних помилок у цифрових текстах.

Розроблена інформаційна система автоматизованого виправлення граматичних помилок у цифрових текстах на платформі.NET, має виконувати такі основні функції:

- Одержання текстового навчального контенту.
- Початкова обробка тексту навчального контенту (стоп-знаки, цифри, регістр).
- Пошук меж слів (роздільники – прогалини тощо).
- Побудова навчального масиву коректних слів.
- Побудова лексичного масиву слів за навчальним масивом коректних слів.
- Обрахунок значень кількості появ кожного з слів у лексичному масиві.
- Побудова матриці лексичних n-грам за даними лексичного масиву слів.
- Відображення n-грам з обраною користувачем кількістю осередків.
- Одержання тексту для перевірки.
- Початкова обробка тексту для перевірки.
- Побудова текстового масиву для перевірки.
- Перевірка наявності кожного з слів перевірючого текстового масиву у матриці лексичних n-грам.
- Візуальне виділення правильних слів, що наявні в матриці лексичних n-грам.
- Візуальне виділення неправильних слів, що відсутні в матриці лексичних n-грам.
- Формування й сортування масивів правильних слів-альтернатив до кожного з неправильних слів.
- Заміна користувачем неправильних слів на обрані слова із масивів правильних слів-альтернатив.

## Розділ 2 Проєктування інформаційної системи

### 2.1 Метод автоматизованого виправлення граматичних помилок у цифрових текстах

Схема методу автоматизованого виправлення граматичних помилок у цифрових текстах графічно представлена на рисунку 2.1.

Згідно пропонованого методу, вхідними даними є тестовий навчальний контент та безпосередньо сам текст для перевірки для автоматизованого виправлення граматичних помилок.

Умовно метод можна розбити на послідовність виконання кроків, першим з яких є формування навчального масиву коректних слів. На першому кроці спершу одержується текстовий навчальний контент, який проходить початкову обробку тексту (прибирання стоп-знаків, цифр, приведення до однакового регістру). Далі проходить пошук меж слів, згідно заданих роздільників (роздільники – прогалини тощо). По знайденим словам відбувається побудова навчального масиву коректних слів.

Другим кроком виконання є побудова матриці лексичних n-грам. У межах кроку відбувається побудова лексичного масиву слів за навчальним масивом коректних слів та обрахунок значень кількості появ кожного з слів у лексичному масиві. Далі за даними лексичного масиву слів здійснюється побудова матриці лексичних n-грам та відображення n-грам з обраною користувачем кількістю осередків.

Третім кроком виконання є пошук граматичних помилок. У рамках кроку користувач завантажує текст для перевірки, який проходить початкову обробку тексту та за обробленим текстом відбувається побудова текстового масиву для перевірки. Ну і далі здійснюється безпосередньо перевірка наявності кожного з слів перевіреного текстового масиву у матриці лексичних n-грам.

Четвертим кроком виконання буде виправлення граматичних помилок. У рамках кроку здійснюється візуальне виділення правильних слів, що наявні в

матриці лексичних n-грам, а також візуальне виділення неправильних слів, що відсутні в матриці лексичних n-грам (помилкових слів).



Рисунок 2.1 – Схема методу автоматизованого виправлення граматичних помилок у цифрових текстах

За отриманими даними формується й сортується масив правильних слів-альтернатив до кожного з неправильних слів. І останнім підпунктом є заміна користувачем неправильних слів на обрані слова із масивів правильних слів-альтернатив.

У якості вихідних даних буде виведено виправлений текст для користувача.

## **2.2 Інформаційна структура програмного застосунку для автоматизованого виправлення граматичних помилок**

### **2.2.1 Проектна архітектура системи та взаємозв'язок компонентів**

Загальна схема програмного застосунку для автоматизованого виправлення граматичних помилок у цифрових текстах представлена на рисунку 2.2.

Як видно з рисунку 2.2 – програмний застосунок складається із трьох ключових модулів: модуля формування навчального масиву коректних слів, модуля побудови матриці лексичних n-грам, модуля пошуку та виправлення граматичних помилок, а також бази даних.

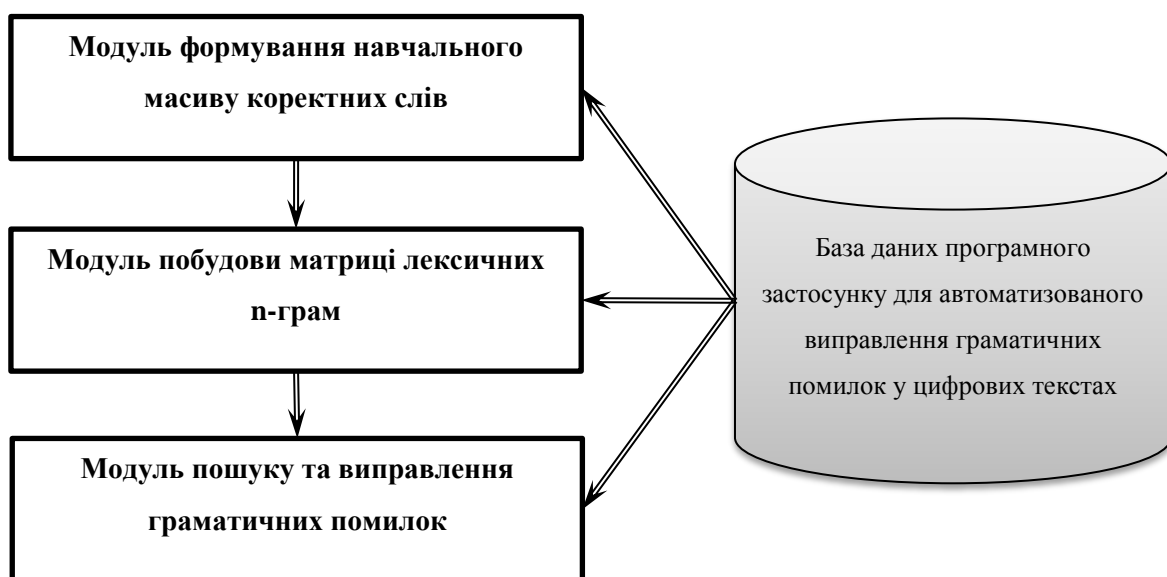


Рисунок 2.2 – Схема програмного застосунку для автоматизованого виправлення граматичних помилок у цифрових текстах

Більш детально про кожен ключовий модуль буде розписано нижче. На рисунку 2.3 проілюстровано схему модулю формування навчального масиву коректних слів.

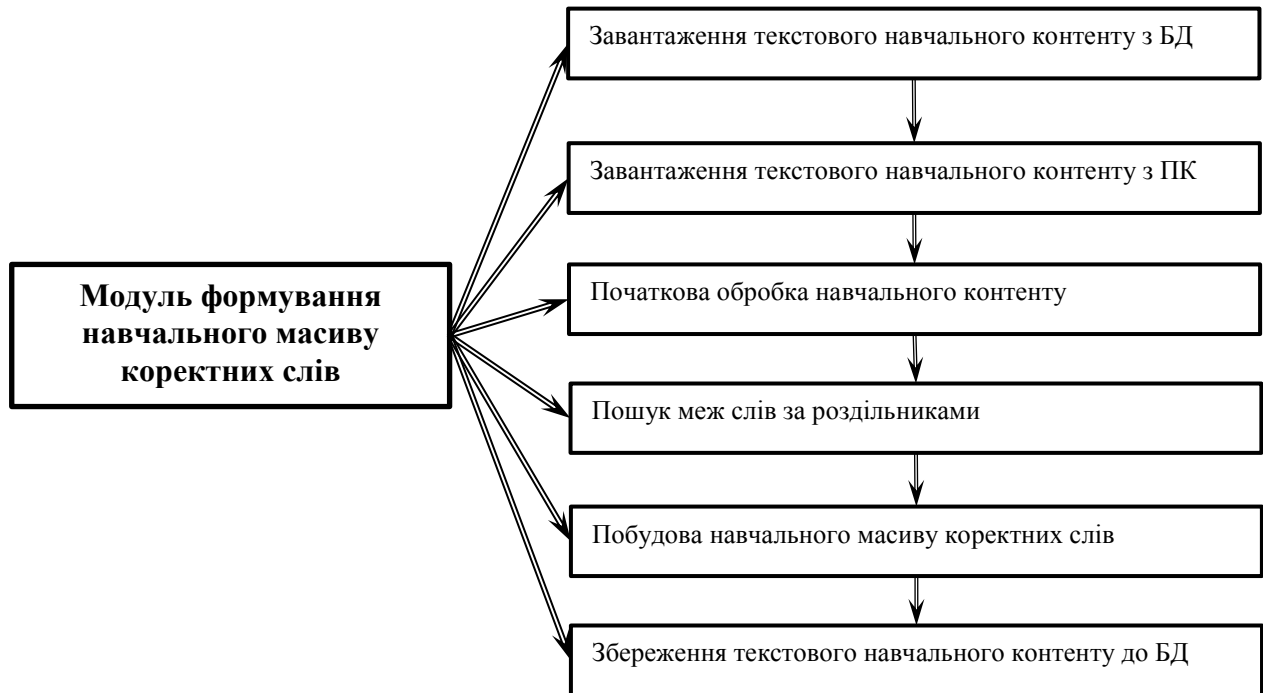


Рисунок 2.3 – Схема модулю формування навчального масиву коректних слів

У рамках модуля користувач має можливість завантажувати текстовий навчальний контент як з бази даних, так і з ПК. Також у рамках модуля є початкова обробка навчального контенту, пошук слів за вказаними користувачем роздільниками та по знайденим словам побудова навчального масиву коректних слів. Також є можливість виконати збереження тестового навчального контенту до БД.

На рисунку 2.4 більш детально проілюстровано схему модулю побудови матриці лексичних n-грам.

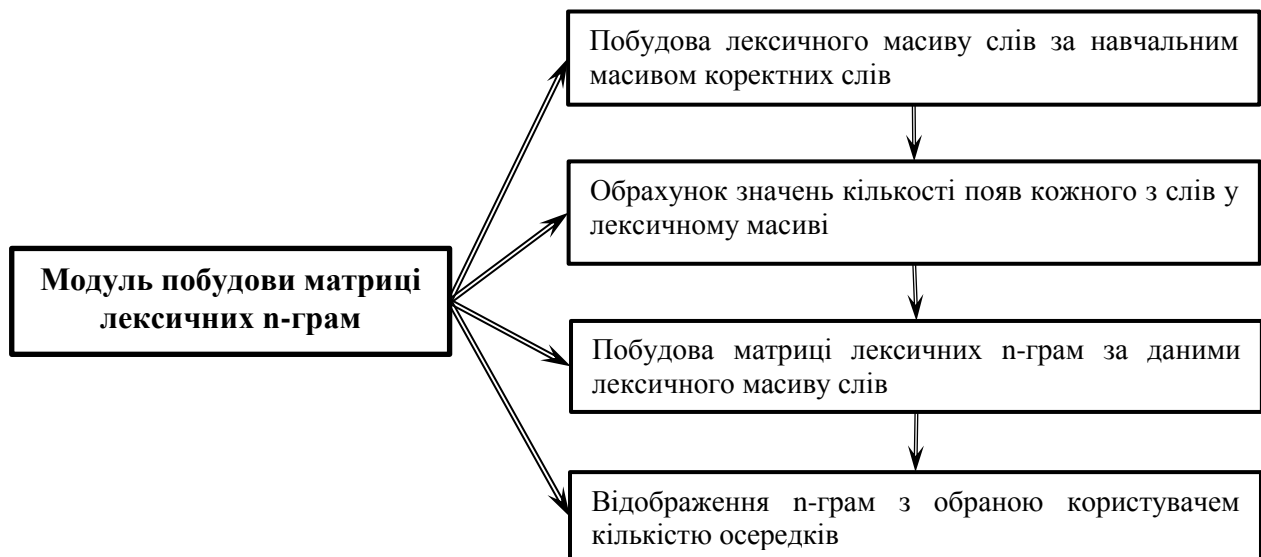


Рисунок 2.4 – Схема модулю побудови матриці лексичних n-грам

У рамках модулю побудови матриці лексичних n-грам відбувається побудова лексичного масиву слів за навчальним масивом коректних слів, сформований попереднім модулем. У подальшому обраховуються значення кількості появ кожного з слів у лексичному масиві та будується безпосередньо матриця лексичних n-грам за даними лексичного масиву слів. З матриці лексичних n-грам користувач може обрати кількість осередків, які будуть відображені користувачу.

На рисунку 2.5 детально проілюстровано схему модулю пошуку та виправлення граматичних помилок.

У рамках третього модуля відбувається пошук та виправлення граматичних помилок. Текст для аналізу можна завантажити як з БД так і з ПК. Далі подібно як у першому модулі – до тексту застосовується первинна обробка, після проведення якої відбувається побудова текстового масиву для перевірки на правильність. У подальшому здійснюється перевірка наявності кожного з слів перевіреного текстового масиву у матриці лексичних n-грам. Після виконаної перевірки для користувача буде сформовано візуальне представлення – виділення правильних слів, що наявні в матриці лексичних n-грам та виділення неправильних слів, що відсутні в матриці лексичних n-грам. Надалі здійснюється формування й сортування масивів правильних слів-альтернатив до кожного з

неправильних слів та заміна користувачем неправильних слів на обрані слова із масивів правильних слів-альтернатив.

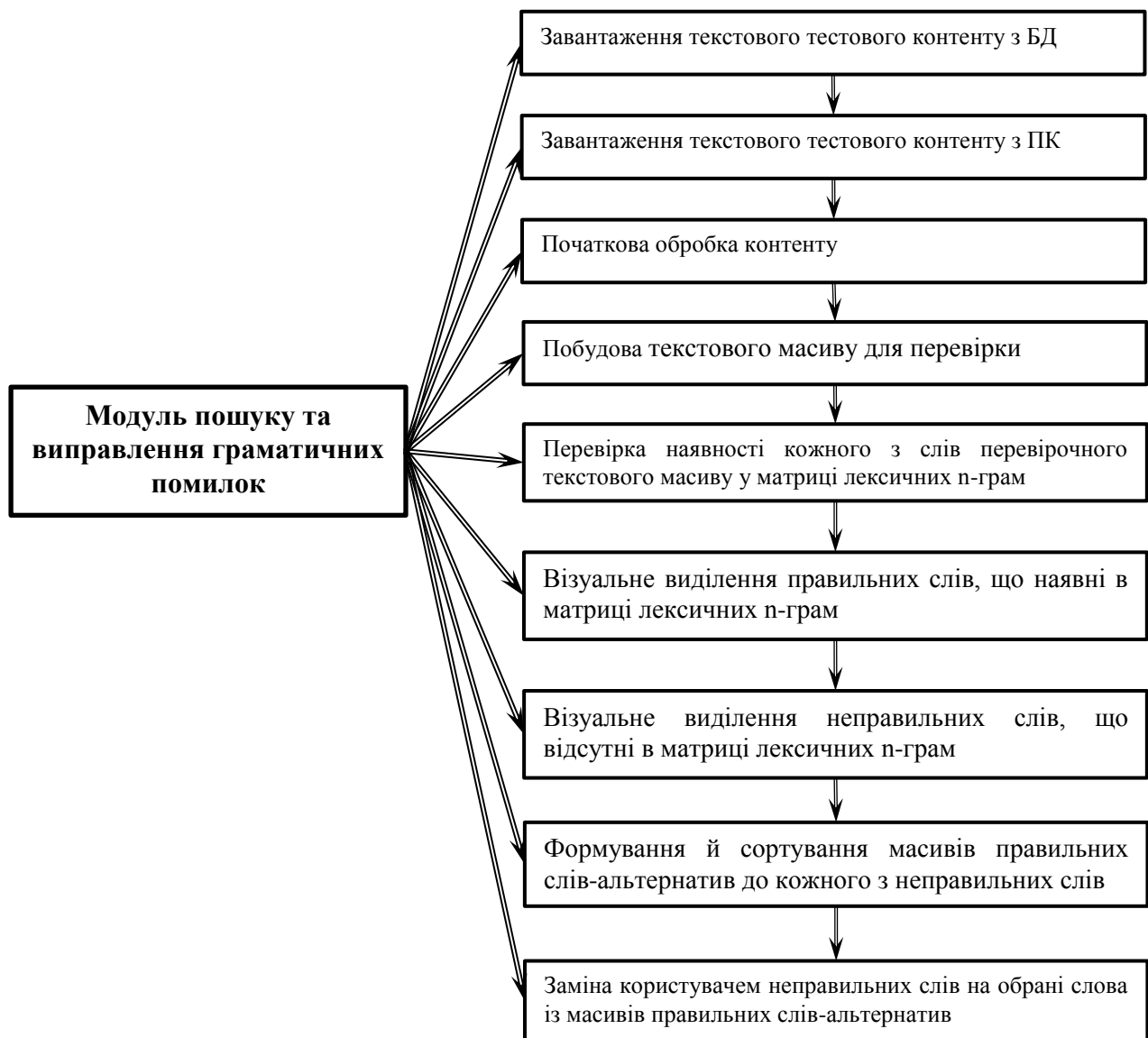


Рисунок 2.5 – Схема модулю пошуку та виправлення граматичних помилок

Стосовно структури БД розроблюваного застосунку буде описано нижче.

### 2.2.2 Інформаційна модель

Для забезпечення надійної роботи програмного застосунку на базі методу автоматизованого виправлення граматичних помилок у цифрових текстах

необхідно створити базу даних, в якій буде встановлено зв'язки між таблицями, що зберігатимуть інформацію, проміжні результати обчислень, тощо.

На рисунку 2.6 зображено схему бази даних програмного застосунку для автоматизованого виправлення граматичних помилок у цифрових текстах.

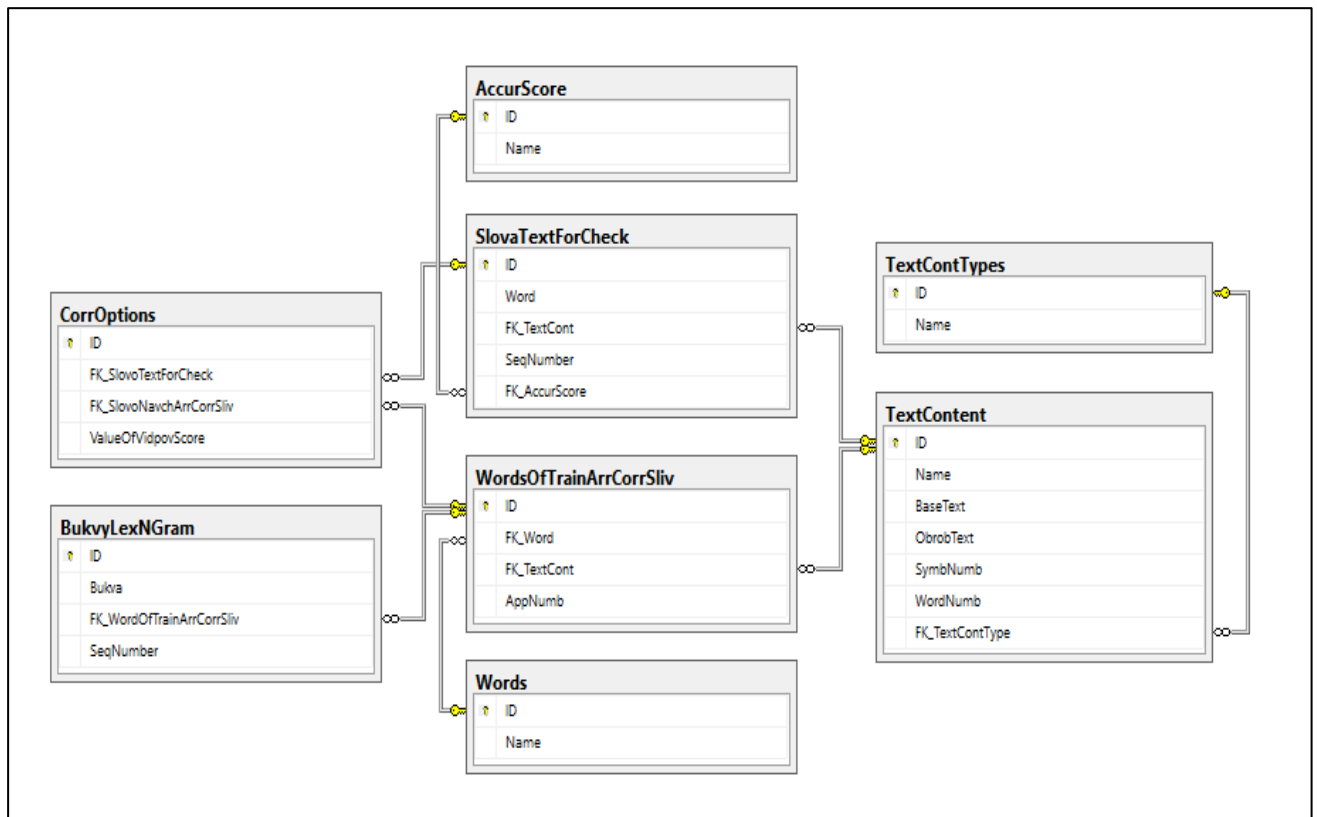


Рисунок 2.6 – Схема бази даних програмного застосунку для автоматизованого виправлення граматичних помилок у цифрових текстах.

Таблиця «TextContent» (таблиця 2.1) призначена для зберігання цифрових текстів та основної інформації про них. Таблиця «Words» (таблиця 2.2) призначена для збереження окремих слів текстового файлу. Таблиця «TextContTypes» (таблиця 2.3) містить дані щодо назв типів текстового контенту.

Таблиця 2.1 – Атрибути таблиці «TextContent»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Первинний ключ, числовий ідентифікатор для однозначного визначення запису таблиці
2.	Name	text	Назва цифрового тексту
3.	BaseText	text	Цифровий текст, що буде проаналізовано
4.	ObrobText	text	Цифровий текст для аналізу, що переведено в нормальну форму
5.	SymbNumb	int	Кількість символів, що містить текст
6.	WordNumb	int	Кількість слів, що містить текст
7.	FK_TextContType	int	Вторинний ключ, посилання на таблицю «TextContTypes» для співставлення із відповідним типом текстового контенту.

Таблиця 2.2 – Атрибути таблиці «Words»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Первинний ключ, числовий ідентифікатор для однозначного визначення запису таблиці
2.	Name	varchar(50)	Контент слова.

Таблиця 2.3 – Атрибути таблиці «TextContTypes»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Первинний ключ, числовий ідентифікатор для однозначного визначення запису таблиці
2.	Name	varchar(50)	Назва типу текстового контенту.

Таблиця «WordsOfTrainArrCorrSliv» (таблиця 2.4) створена для збереження інформації щодо слів навчального масиву коректних лексичних одиниць. Таблиця «BukvyLexNGram» (таблиця 2.5) зберігає окремо букви лексичних n-грам.

Таблиця 2.4 – Атрибути таблиці «WordsOfTrainArrCorrSliv»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Первинний ключ, числовий ідентифікатор для однозначного визначення запису таблиці.
2.	FK_Word	int	Вторинний ключ, посилання на таблицю «Words» для співставленням із відповідним словом.
3.	FK_TextCont	int	Вторинний ключ, посилання на таблицю «TextContTypes» для співставленням із відповідним типом текстового контенту.
4.	AppNumb	int	Числове значення кількості появ у тексті.

Таблиця 2.5 – Атрибути таблиці «BukvyLexNGram»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Первинний ключ, числовий ідентифікатор для однозначного визначення запису таблиці
2.	Bukva	varchar(50)	Буква слова
3.	FK_WordOfTrainArrCorrSliv	int	Вторинний ключ, посилання на таблицю «WordOfTrainArrCorrSliv» для співставленням із словом матеріалу.
4.	SeqNumber	varchar(50)	Числовий номер в послідовності.

Таблиця «AccurScore» (таблиця 2.6) створена для збереження назв оцінок правильності написання слів.

Таблиця 2.6 – Атрибути таблиці «AccurScore»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Первинний ключ, числовий ідентифікатор для однозначного визначення запису таблиці
2.	Name	varchar(50)	Назва оцінки правильності

Таблиця «SlovaTextForCheck» (таблиця 2.7) створена для збереження інформації щодо слів тексту для перевірки.

Таблиця 2.7 – Атрибути таблиці «SlovaTextForCheck»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Первинний ключ, числовий ідентифікатор для однозначного визначення запису таблиці
2.	Name	varchar(50)	Слово для подальшої перевірки
3.	FK_TextCont	int	Вторинний ключ, посилання на таблицю «TextContent» для співставлення із словом матеріалу.
4.	SeqNumber	int	Номер слова в послідовності.
5.	FK_AccurScore	int	Вторинний ключ, посилання на таблицю «AccurScore» для співставлення із відповідною оцінкою правильності написання слова.

Таблиця «CorrOptions» (таблиця 2.8) створена для збереження варіантів виправлень слів.

Таблиця 2.8 – Атрибути таблиці «CorrOptions»

№ п/п	Назва атрибуту	Тип даних	Опис
1.	ID	int	Первинний ключ, числовий ідентифікатор для однозначного визначення запису таблиці
2.	FK_SlovoTextForCheck	Int	Вторинний ключ, посилання на таблицю «Words» для співставлення із відповідним словом для перевірки.
3.	FK_SlovoNavchArrCorrSliv	Int	Вторинний ключ, посилання на таблицю «WordsOfTrainArrCorrSliv» для співставлення із відповідним словом навчального масиву коректних слів.
4.	ValueOfVidpovScore	float	Числове значення оцінки відповідності.

Отже, в результаті виконання підрозділу було реалізовано необхідну базу даних для створення програмного продукту на основі методу автоматизованого виправлення граматичних помилок у цифрових текстах. На основі даталогічної моделі БД створено необхідні для роботи таблиці та заповнено їх початковими вхідними даними.

### 2.3 Вибір засобів розробки інформаційної системи

Для розробки застосунку, розробники використовують набір компонентів, які допомагають їм в цьому. Згідно поставленого завдання, необхідним є розробка віконного застосунка на платформі .NET з використанням бази даних для збереження та роботи з даними, що необхідні для виконання завдання. Необхідним є обрати наступні компоненти:

- мова програмування, на якій буде створюватись застосунок відповідно до мов, що підтримує платформа .NET;

- редактор програмного коду, що представляє собою інтегроване середовище програмування;
- систему керування базами даних, яка забезпечить створення та роботу з базою.

Далі розглянуто вибір кожного з компонентів.

### 2.3.1 Вибір мови програмування

Платформа .NET підтримує розробку на декількох мовах програмування, як от:

- C#;
- VB.NET;
- C++;
- F#.

Проте, мовою, яка була спеціально створена для даної платформи є саме мова програмування C#, тому її варто обрати для розробки застосунку. Також ця мова має ряд переваг, далі їх розглянуто детальніше.

C# – це об'єктно-орієнтована й компонентно-орієнтована мова програмування, що була розроблена Андресом Хейлсбергом та його командою компанії Microsoft. Мова працює на платформі .NET [33]. C# можна охарактеризувати двома поняттями – це сучасність у функціях і простота коду. Ця мова програмування постійно зазнає оновлень сучасними елементами та підтримує різного роду інтеграції.

Мова C# легка для вивчення, оскільки, вивчивши невелику частину мови, можна одразу писати корисний код. Більш розширені функції можна вивчати, вже під час роботи [33].

C# дуже добре інкапсулює складний код. Програміст може використовувати функції C#, навіть не замислюючись про те, як вони працюють в середині. Це важливий принцип інкапсуляції об'єктно-орієнтованого програмування. Також можна використовувати не тільки стандартні бібліотеки

для написання коду, а й сторонніх розробників, адже ця мова програмування є доволі популярною, тому кількість таких сторонніх бібліотек дуже велика [33].

Найбільшою перевагою цієї мови є її здатність економити час розробки завдяки статично типізованим даним і легкій читабельності, що зменшує час налагодження.

Однією з переваг C# є те, що він автоматично збирає сміття. Програми написані на цій мові менш схильні до помилок, оскільки нелогічні змінні не можна використовувати як умови. У C# не існує глобальних змінних або функцій, оскільки всі вони належать до класу. Сфера застосування цієї мови досить широка:

- веб-застосунки;
- віконні застосунки;
- застосунки для баз даних;
- застосунки для веб-сервісів.

Отже, ця мова програмування якнайкраще підходить для реалізації програмного застосунку.

### **2.3.2 Вибір редактора програмного коду**

Для швидкого написання коду програмних застосунків використовується спеціальний редактор програмного коду. Він називається інтегрованим середовищем програмування.

Інтегроване середовище програмування повинне забезпечувати автоматизацію написання коду та інструменти для відлагодження коду. Більшість інтегрованих середовищ програмування підтримують декілька мов програмування [34]. Найпотужнішим середовищем програмування, що підтримує мову програмування C# є Microsoft Visual Studio.

Microsoft Visual Studio – це середовище програмування для розробників, що дозволяє писати та редагувати код. Visual Studio має редактор коду, який підтримує IntelliSense – цей компонент дозволяє пришвидшувати написання

коду, тому що може завершувати код замість програміста. Також середовище підтримує рефакторинг коду. Інші вбудовані інструменти включають конструктор для створення додатків із графічним інтерфейсом користувача, веб-дизайнер, конструктор класів і дизайнер схем баз даних [35].

Середовище надає можливість налаштувати свій інтерфейс (рисунок 2.7) під потреби розробника, даючи швидкий доступ до необхідних функцій. Також добре відображає структуру проекту, і спрощує перехід між його компонентами.

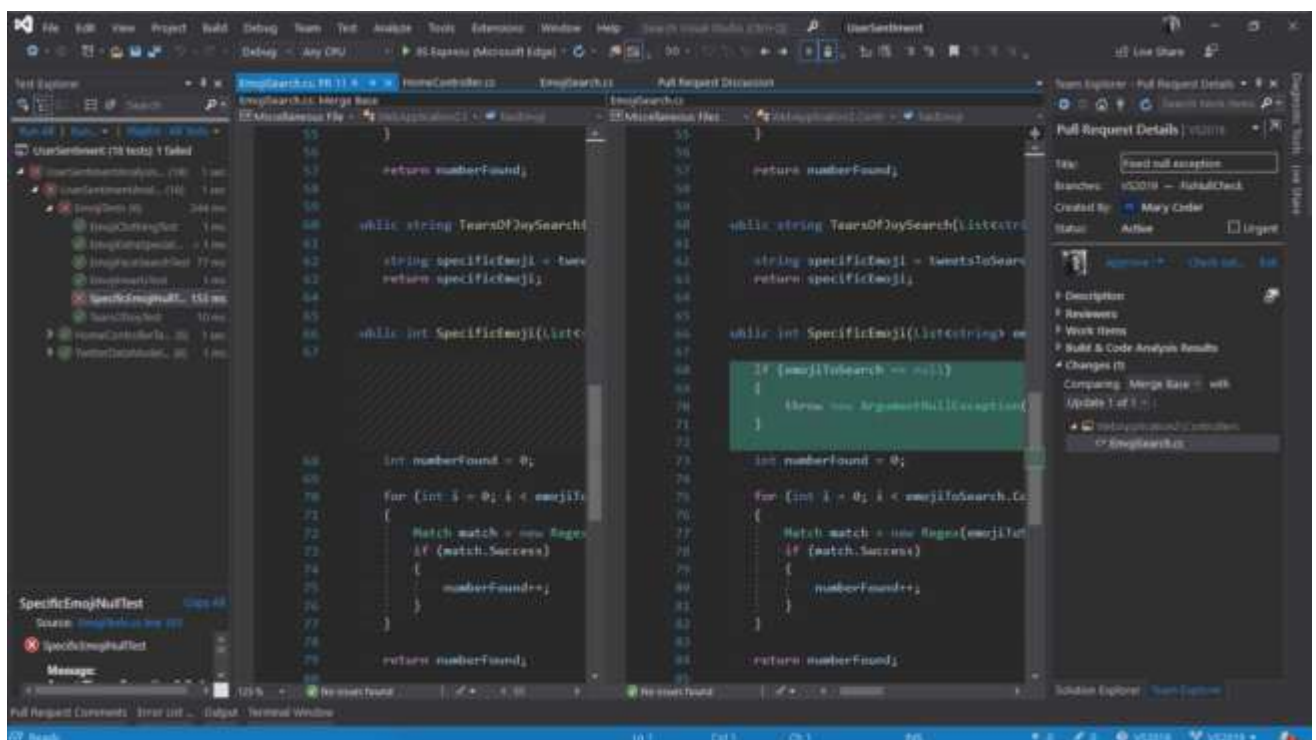


Рисунок 2.7 – Вигляд середовища програмування Microsoft Visual Studio [36]

Дане середовище програмування дозволяє проводити тестування створеного застосунку та виявляти його слабкі місця. Це дозволяє розробнику усунути їх ще на етапі створення.

Отже, Microsoft Visual Studio – ідеальний редактор програмного коду. Адже багато в чому автоматизує роботу розробника, а значить спрощує її та економить час. Також великою перевагою є те, що воно також є продуктом компанії Microsoft, що означає максимальну сумісність з платформою .NET та мовою програмування C#.

### 2.3.3 Вибір СКБД

Створення бази даних та для застосунку та робота з нею забезпечуються системою керування баз даних. В якості СКБД обрано MS SQL Server.

MS SQL Server – це програма, що призначена для зберігання та обробки даних в базі даних. Для того, щоб взаємодіяти з нею користувачі надсилають запити та отримують відповіді, можуть це робити як локально, так і через мережу. Отож MS SQL Server має ряд переваг, а саме [37]:

- має дві функції підвищення безпеки на рівні сервера: режим автентифікації Windows і змішаний режим;
- керування користувачами (призначення ролей і привілеїв для доступу до СКБД);
- доступна вбудована автоматична настройка продуктивності, настроювані функції оптимізації продуктивності, що забезпечують баланс між високою продуктивністю та цілісністю даних;
- можливість налаштування резервного копіювання з певною періодичністю, що також підвищує безпеку збережених даних;
- розвинений графічний інтерфейс, що спрощує взаємодію користувача з системою;
- максимальна інтеграція з інструментами для розробки компанії Microsoft.

Суттєвим недоліком можна вважати те, що дана СКБД орієнтована лише на роботу на операційній системі Windows, проте обраний віконний тип застосунку на платформі .NET нівелює цей недолік.

Отже, перераховані вище переваги даної СКБД дають право вважати, що вона найкращим чином підходить для роботи з базою даних в рамках програмної реалізації методу автоматизованого виправлення граматичних помилок у цифрових текстах.

## Розділ 3 Програмна реалізація інформаційної системи

### 3.1 Структура та функціональне призначення програмних складових системи

Діаграма класів програмного застосунку для автоматизованого виправлення граматичних помилок у цифрових текстах зображена на рисунку 3.1.

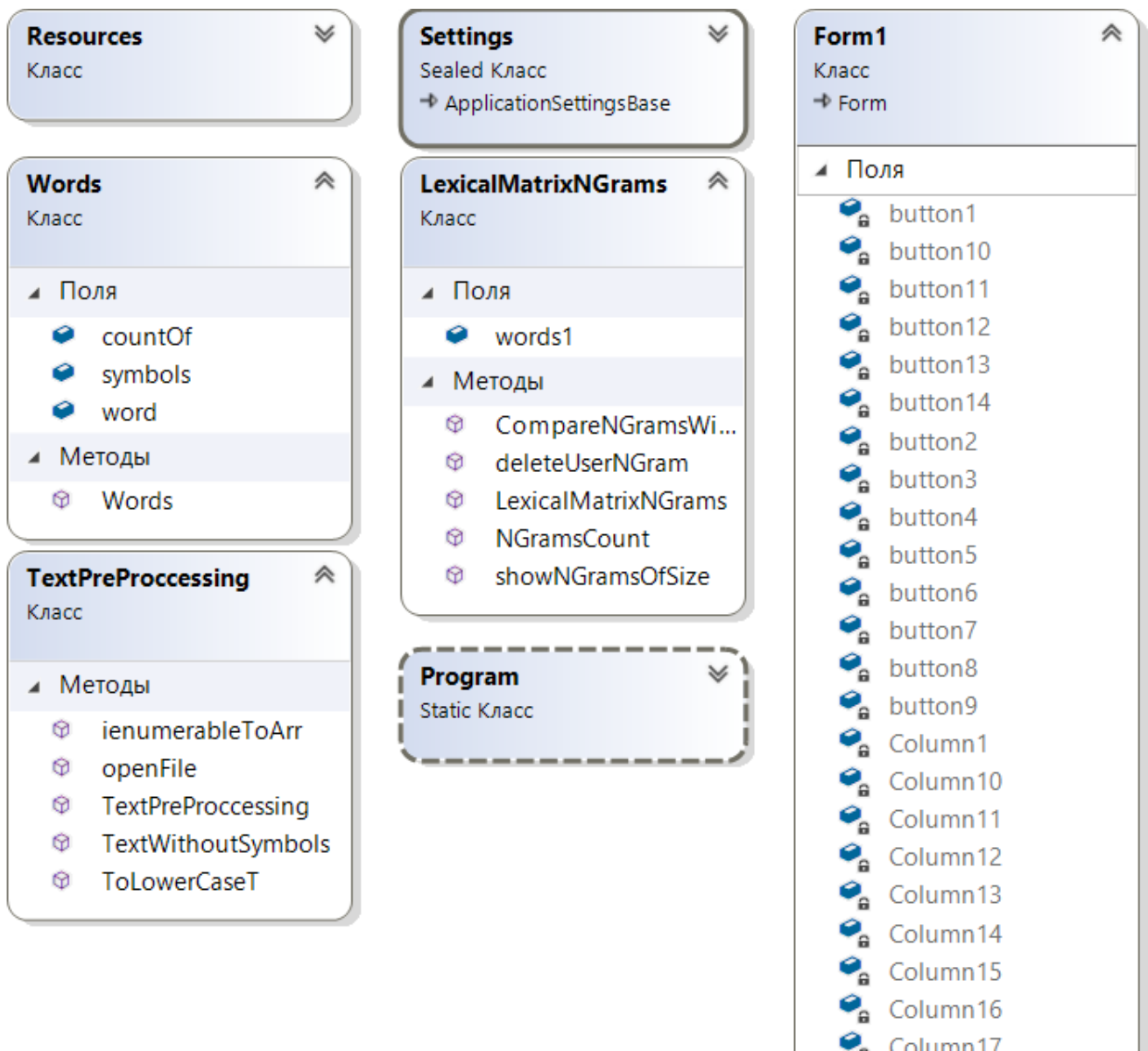


Рисунок 3.1 – Діаграма класів програмного застосунку для автоматизованого виправлення граматичних помилок у цифрових текстах

Клас `Words` призначений для збереження структури даних зручної для подальшого аналізу. Він містить саме слово, його масив символів, а також кількість зустрічань певної лексичної n-грами у тексті.

Клас `TextPreProcessing` містить метод відкриття тексту з файлу – `openFile()`, метод прибирання стоп-символів – `TextWithoutSymbols()`, метод переведення тексту у нижній регістр – `ToLowerCaseT()` та метод переведення перерахування у масив – `IEnumerableToArr()`.

Клас `LexicalMatrixNGrams` призначений для реалізації другого та третього модулів програмного продукту та містить методи видалення користувачем обраної n-грами (`deleteUserNGram`), відображення n-грам заданого розміру (`showNGramsOfSize`), підрахунок кількості n-грам у тексті (`NGramsCount`) та порівняння n-грам у тестовому тексті з n-грамами з бази даних (`CompareNGramsWithTest`). Решту функціонування програмного забезпечення реалізовано безпосередньо в обробниках подій натиснення на компоненти форми. Деталі реалізації описано у наступному пункті.

### 3.2 Особливості реалізації програмних складових системи

Згідно з вищесказаного, для реалізації програмного застосунку для автоматизованого виправлення граматичних помилок у цифрових текстах потрібно створити реалізації трьох ключових модулів. Перший модуль формування навчального масиву коректних слів зображено на рисунку 3.2.

Для завантаження тексту з файлової системи реалізовано метод `openFile()`, який повертає перерахування слів, зчитаних з файлу, зчитаного з ПК. Код методу подано нижче:

```
public IEnumerable<string> openFile() {
    Encoding code = Encoding.UTF8;
    IEnumerable<string> s = new string[] { };
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.Title = "Завантажити текстовий тестовий контент";
    openFileDialog.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*";
    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        s = File.ReadLines(openFileDialog.FileName, code);
    }
}
```

```

    }
    return s;
}

```

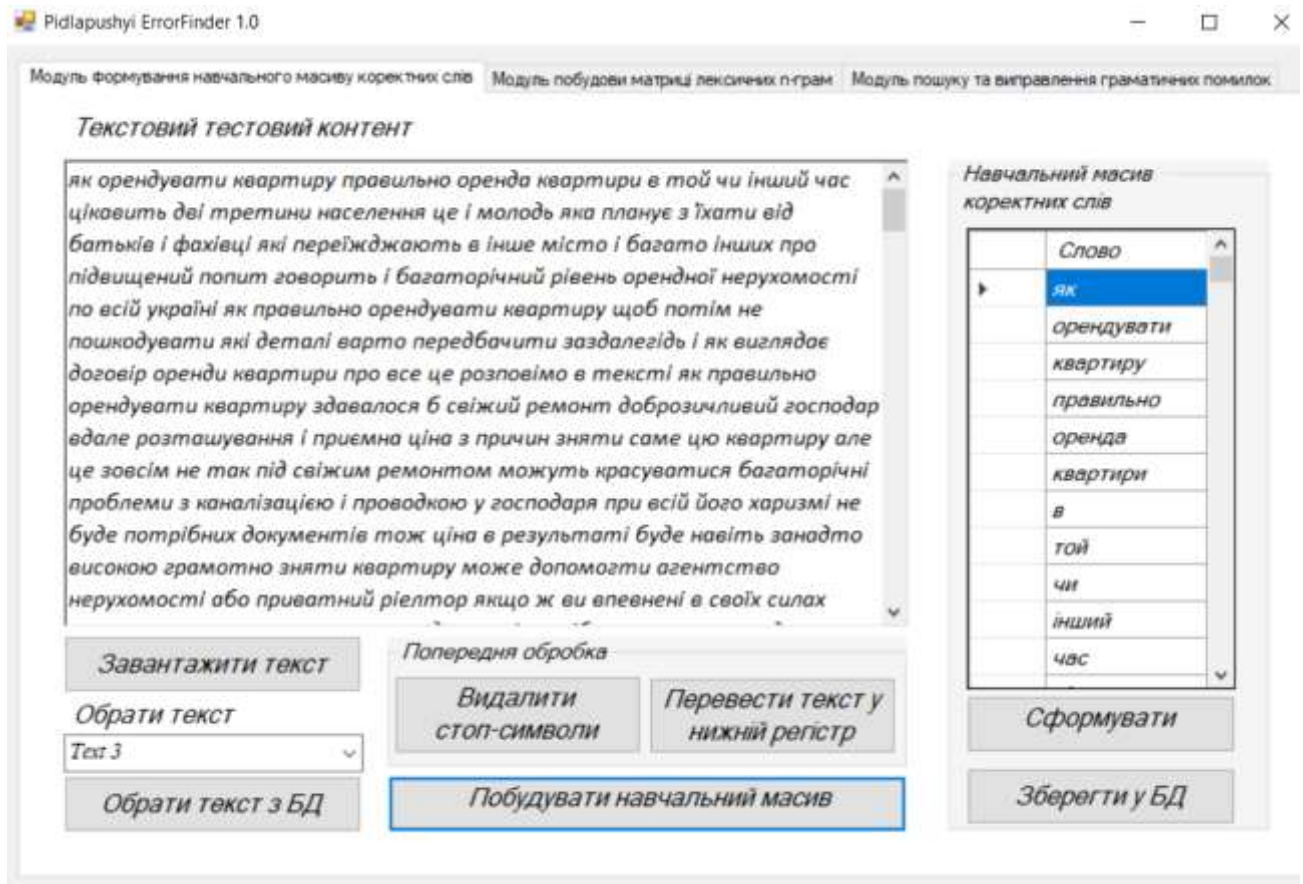


Рисунок 3.2 – Вигляд вкладки «Модуль формування навчального масиву коректних слів» програмного застосунку для автоматизованого виправлення граматичних помилок у цифрових текстах

Викликається метод при натисненні на кнопку «Завантажити текст». Вигляд діалогового вікна проілюстровано на рисунку 3.3.

У рамках реалізації першого модулю після завантаження тексту можна проводити з текстом маніпуляції, а саме: початкову обробку тексту та перетворення тексту у масив слів. У рамках початкової обробки тексту користувач може забрати цифри та стоп-символи, натиснувши на відповідну кнопку «Видалити стоп-символи». Код методу `TextWithoutSymbols()` для видалення стоп-символів проілюстровано нижче:

```

public IEnumerable<string> TextWithoutSymbols(String s)
{
    var words = s.Split(new char[] { ' ', '.', ',', '!', '?', '/', '-', '\'});
}

```

```

    ('(', '[', ']', '-', '!', '>', '<', '#', '=', '"', "'", '@', '+', '.', ',',
    '!', '"', ';', '>', '0', ':',
    '1', '2', '3', '4', '5', '6', '7', '8', '9'})
    .Where(w => !string.IsNullOrEmpty(w));

    return words;
}

```

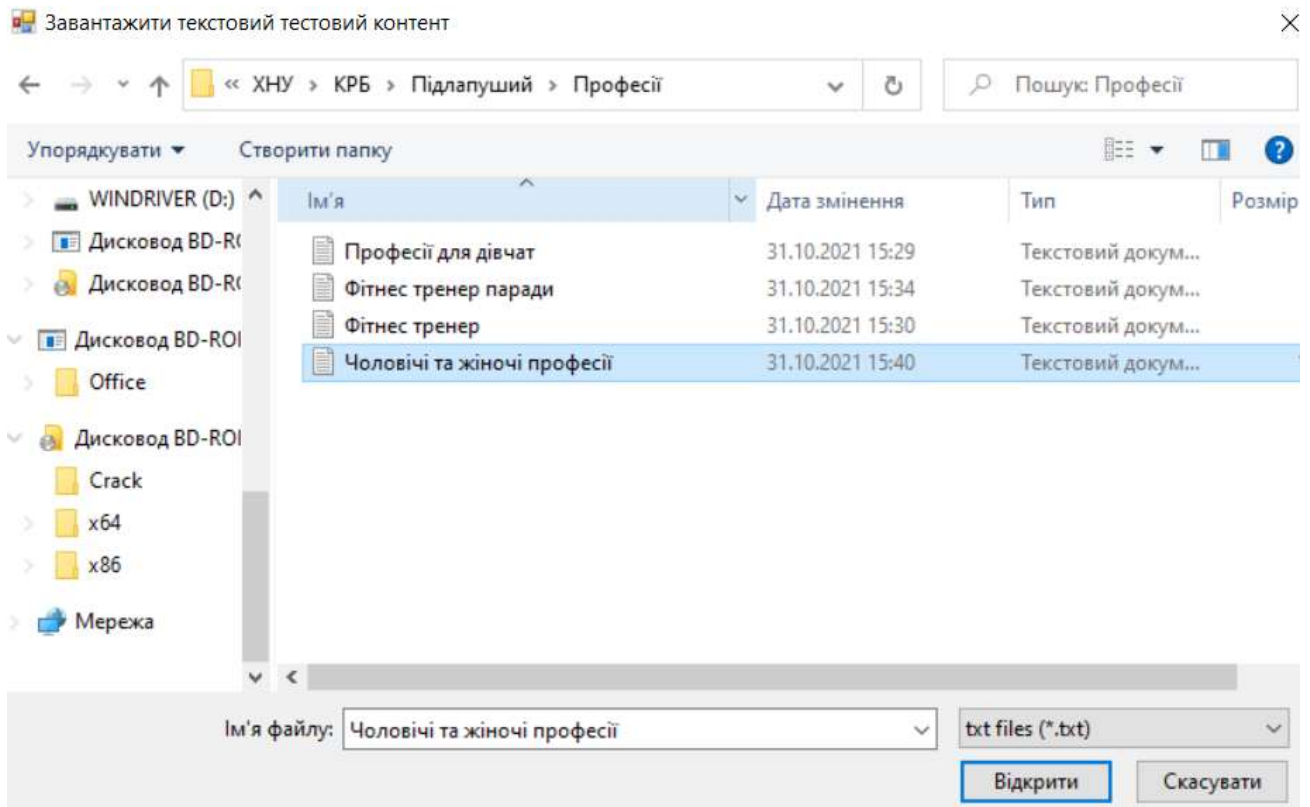


Рисунок 3.3 – Завантаження тексту з файлу

На вхід метод отримує стрінговий текст, повертає користувачу перерахування слів вже без перелічених стоп-елементів. Викликається метод при натисканні кнопки «Видалити стоп-символи» (рисунок 3.4).

Також у межах попередньої обробки можна перетворити текст у нижній регістр. Для цього необхідно натиснути кнопку «Перевести текст у нижній регістр», де спрацює метод `ToLowerCaseT()`, код якого наведено нижче:

```

public String[] ToLowerCaseT(String []s)
{
    int i = 0;
    foreach (string a in s) {
        s[i]= a.ToLower();
        i++;
    }
}

```

```

    return s;
}

```

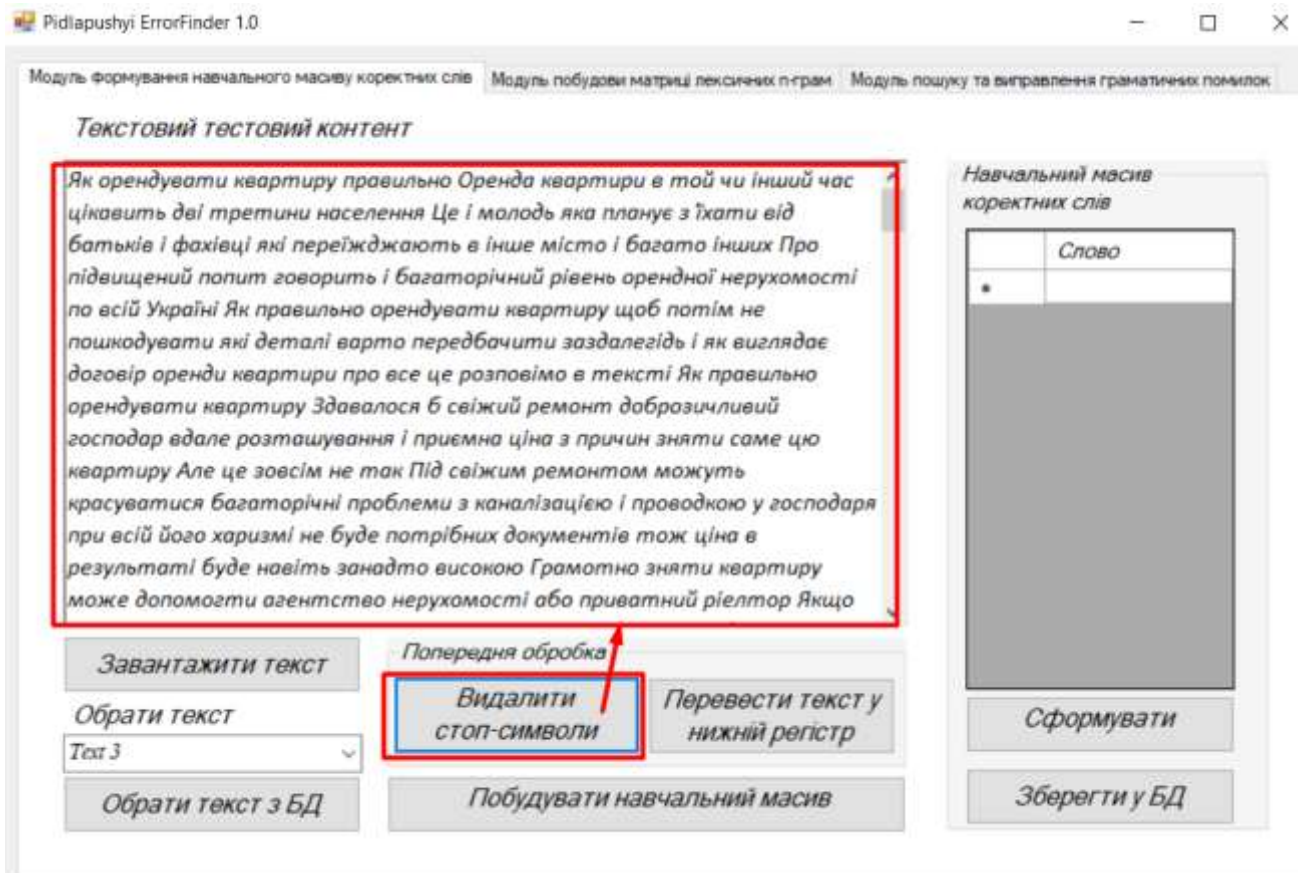


Рисунок 3.4 – Видалити стоп-символи

При цьому, текст у текстовому полі заміниться на текст з маленьких літер (рисунок 3.5).

Навчальний масив коректних слів формується з очищеного попередніми методами тексту, а в таблицю записується кодом:

```

for (int i = 0; i < W.Length; i++)
{
    dataGridView1.Rows.Add(W[i]);
}

```

Відповідно, користувач побачить відображений масив слів (рисунок 3.6).

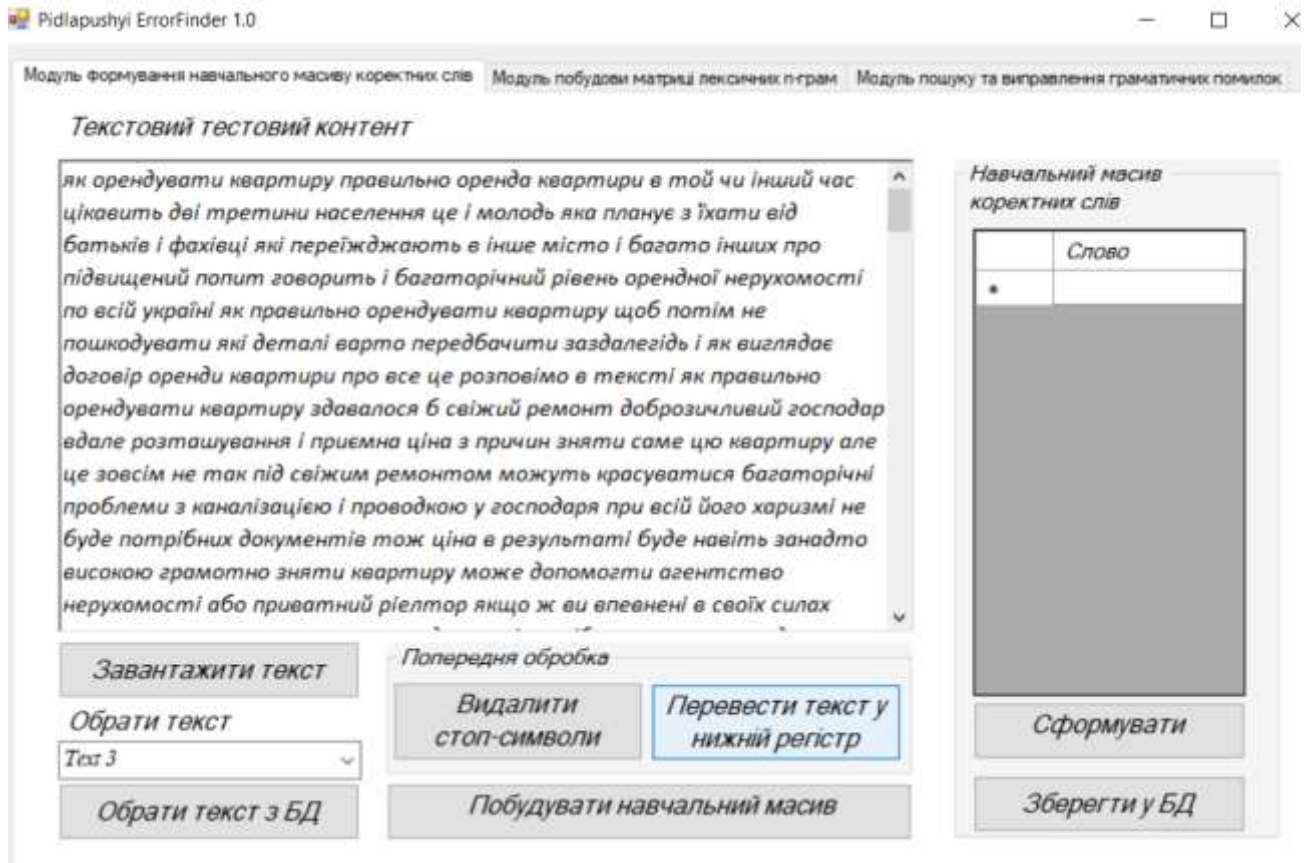


Рисунок 3.5 – Переведення тексту у нижній регістр

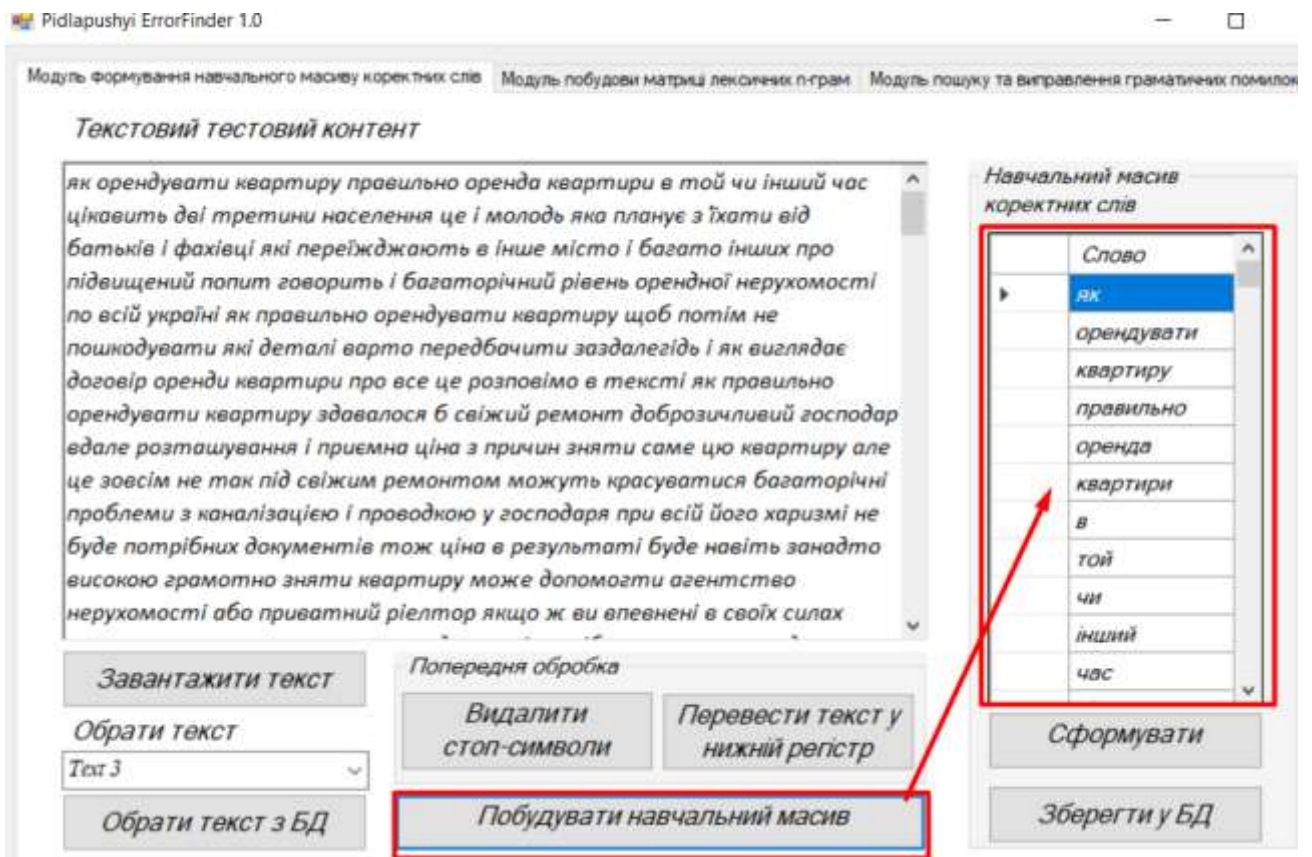


Рисунок 3.6 – Масив коректних слів

Для формування матриці лексичних n-грам використовується фрагмент коду показаний нижче:

```
Words [] w1;
w1 = new Words[W.Length];
for (int i = 0; i < W.Length-1; i++)
{
    dataGridView2.Rows.Add();
    w1[i] = new Words(W[i]);

    for (int j = 0; j < w1[i].symbols.Length; j++)
    {
        dataGridView2.Rows[i].Cells[j + 1].Value = w1[i].symbols[j];
    }
}
```

Заповнюється елементами відповідний масив об'єкта класу Words, який потім записується у відповідні комірки таблиці dataGridView2. Результат виконання коду проілюстровано на рисунку 3.7.

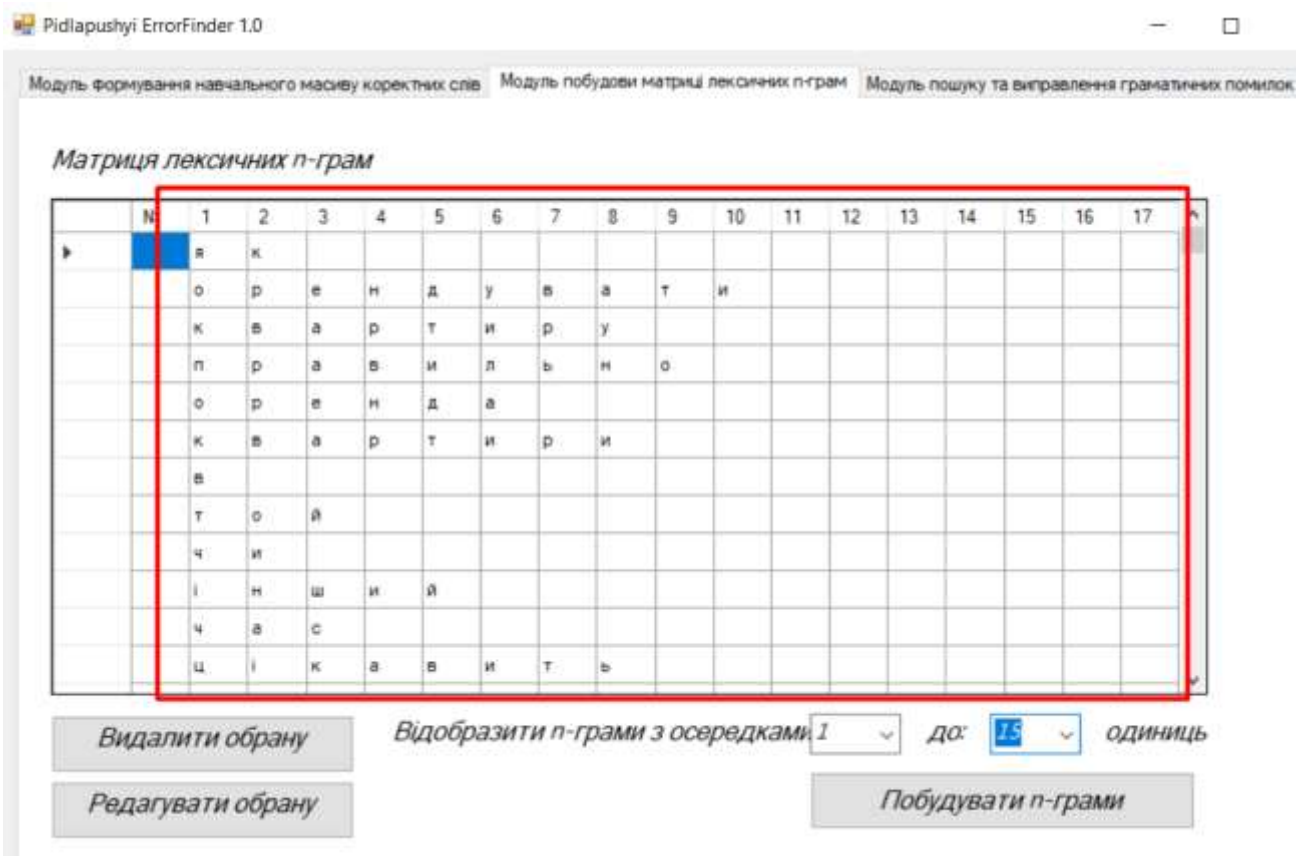


Рисунок 3.7 – Матриця лексичних n-грам

Отже, таким чином були реалізовані основні складові системи програмного застосування для автоматизованого виправлення граматичних помилок у цифрових текстах.

### 3.3 Тестування інформаційної системи

Для перевірки, чи може й наскільки ефективно може програмний продукт виконувати задані в постановці задачі функції, були створені відповідні тести для розроблених методів.

Першим тестовим випадком буде перевірка завантаження текстового тестового контенту вкладки «Модуль формування масиву коректних слів».

Таблиця 3.1 Тест-кейс VR0001

<b>Тест-кейс ID:</b> VR0001	<b>Приоритет:</b> 1	<b>Створено:</b> 17.05.2022, Підлапушний В.О.
<b>Назва:</b> Перевірка завантаження текстового тестового контенту вкладки «Модуль формування масиву коректних слів»		
<b>Кроки</b>		<b>Очікуваний результат</b>
<ol style="list-style-type: none"> <li>1. Запуск програми</li> <li>2. Обрати вкладку «Модуль формування масиву коректних слів»</li> <li>3. Натиснути кнопку «Завантажити текст»</li> <li>4. Знайти на жорсткому диску папки проекту текст із назвою «Text1» у діалоговому вікні</li> <li>5. Перевірити наявність доданого тексту у текстовому полі.</li> </ol>		Завантажений текст показано у текстовому полі
<b>Результат виконання тест-кейсу:</b> перевірку пройдено успішно.		

Після запуску програми необхідно виконати кроки, що вказані у таблиці 3.1, після яких у застосунку користувач побачить результат у вигляді тексту у текстовому полі «Текстовий тестовий контент» (рисунок 3.8).

Наступним тестовим випадком є перевірка коректності видалення стоп-символів з тексту. Послідовність дій виконання даного тесту детально описана у таблиці 3.2.

Таблиця 3.2 Тест-кейс VR0002

<b>Тест-кейс ID:</b> VR0002	<b>Приоритет:</b> 1	<b>Створено:</b> 17.05.2022, Підлапушний В.О.
<b>Назва:</b> Перевірка коректності видалення стоп-символів вкладки «Модуль формування масиву коректних слів»		
<b>Кроки</b>		<b>Очікуваний результат</b>
<ol style="list-style-type: none"> <li>1. Запуск програми</li> <li>2. Обрати вкладку «Модуль формування масиву коректних слів»</li> <li>3. Вставити в поле «Текстовий тестовий контент» речення: «Це перевірка коректності прибирання символів типу: крапка, кома, тире – та інших!»</li> <li>4. Натиснути на кнопку «Видалити стоп-символи»</li> </ol>		З речення повинні прибратись усі стоп-символи
<b>Результат виконання тест-кейсу:</b> перевірку пройдено успішно.		

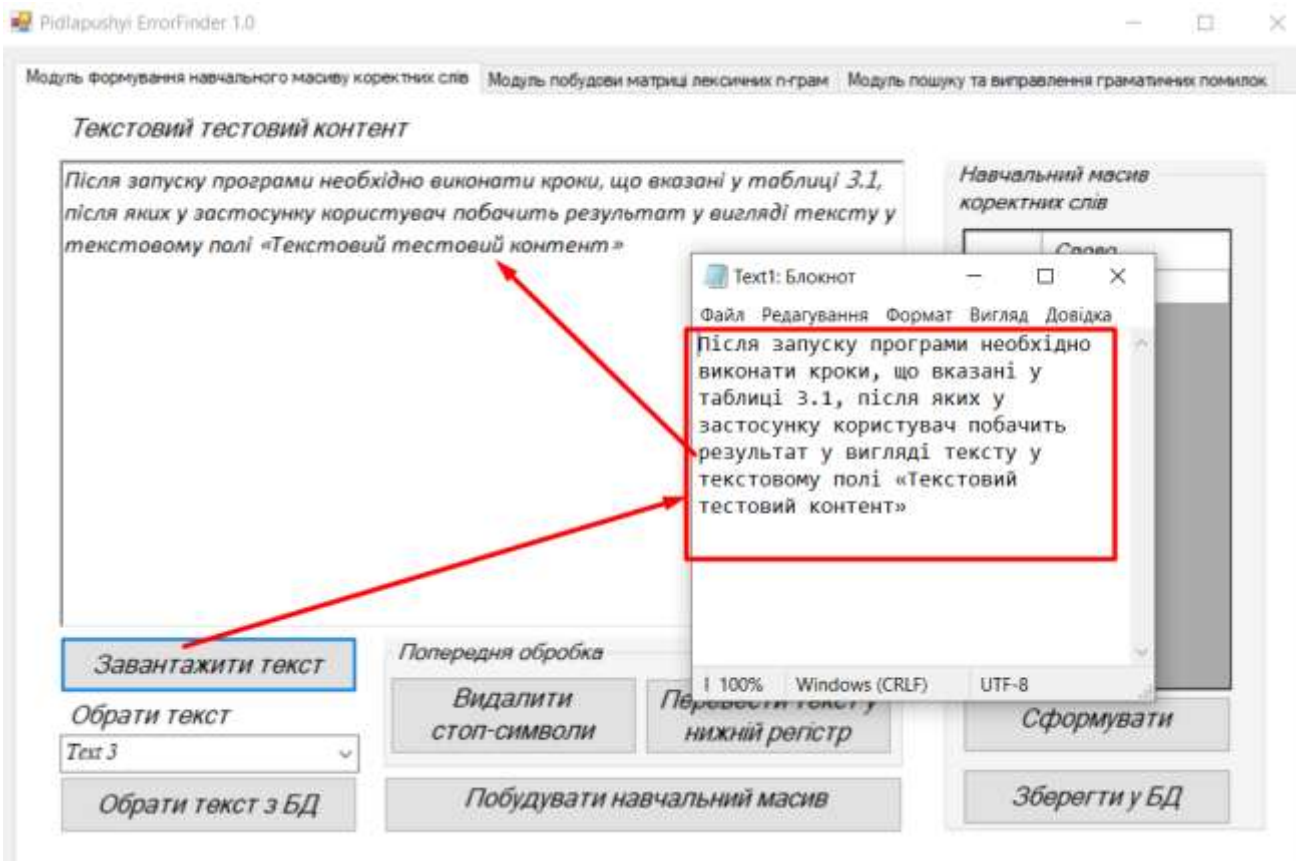


Рисунок 3.8 – Результат проходження тесту завантаження тексту з файлу

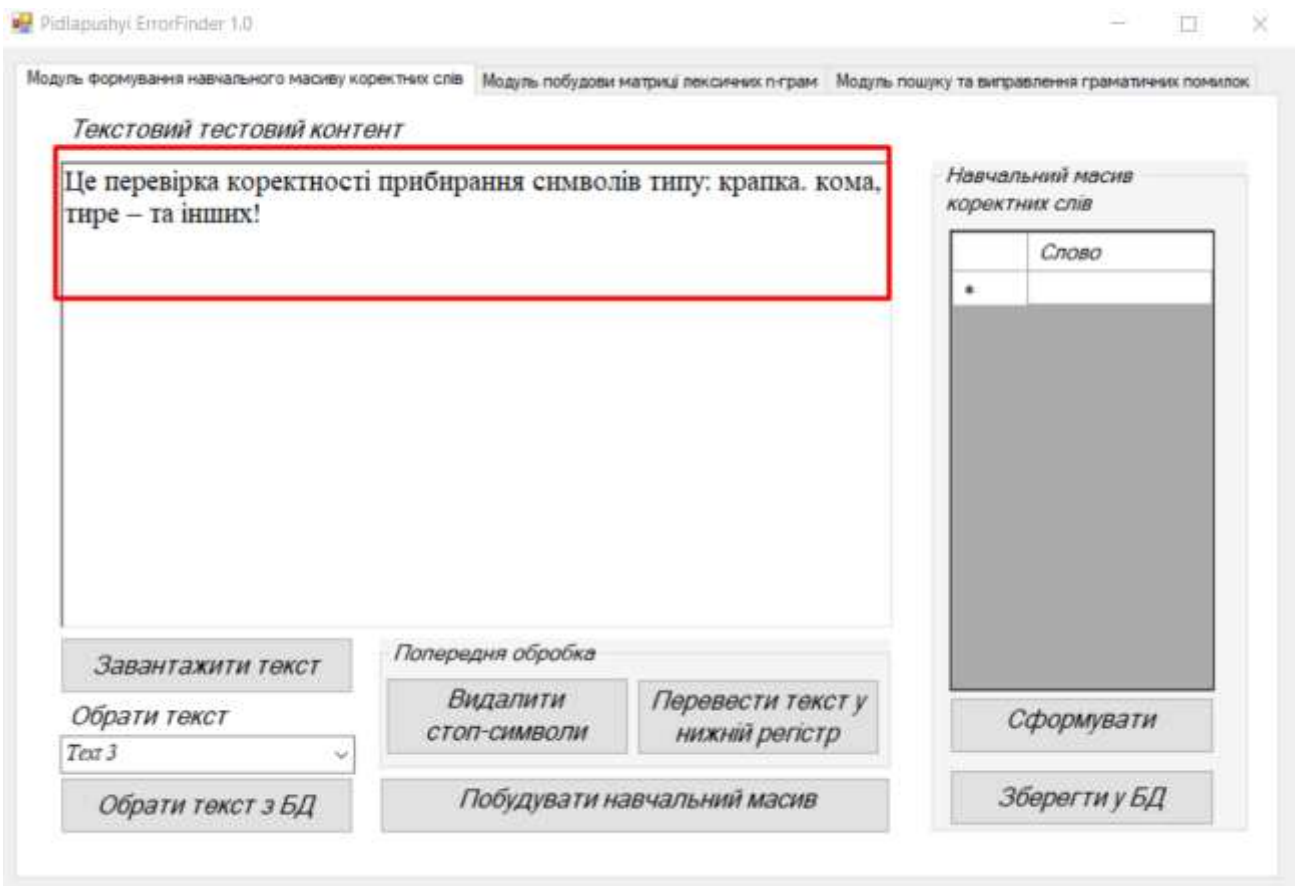


Рисунок 3.9 – Початковий текст

Після запуску програми необхідно виконати кроки, що вказані у таблиці 3.2, після яких у застосунку користувач побачить результат у вигляді очищеного тексту у текстовому полі «Текстовий тестовий контент» (рисунок 3.9 – 3.10).

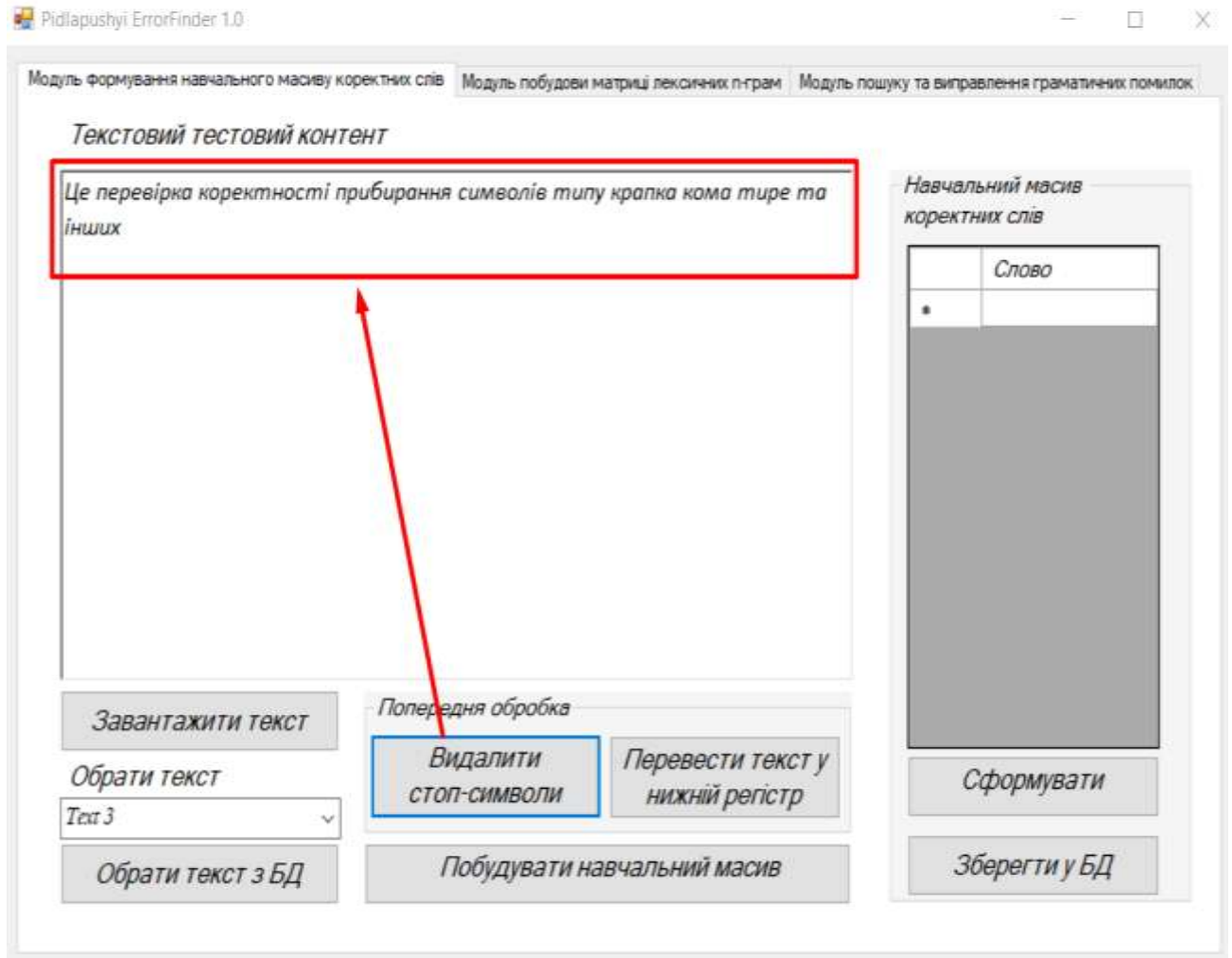


Рисунок 3.10 – Текст після натиснення на кнопку «Видалити стоп-символи»

Як видно з рисунків 3.9 та 3.10 – тест успішно пройдено. Наступним тестовим випадком буде перевірка правильності переведення тестового тексту у нижній регістр. Прогнозовано всі слова повинні стати з маленької літери. Детальніше набір кроків для перевірки даного функціоналу у таблиці 3.3.

Після запуску програми необхідно виконати кроки, що вказані у таблиці 3.3, після виконання яких у програмі користувач побачить результат у вигляді очищеного тексту в нижньому регістрі текстового поля «Текстовий тестовий контент» (рисунок 3.11).

Таблиця 3.3 Тест-кейс VR0003

<b>Тест-кейс ID:</b> VR0003	<b>Пріоритет:</b> 1	<b>Створено:</b> 17.05.2022, Підлапушний В.О.
<b>Назва:</b> Перевірка коректності переведення тексту у нижній регістр «Модуль формування масиву коректних слів»		
<b>Кроки</b>		<b>Очікуваний результат</b>
<ol style="list-style-type: none"> <li>1. Запуск програми</li> <li>2. Обрати вкладку «Модуль формування масиву коректних слів»</li> <li>3. Вставити в поле «Текстовий тестовий контент» речення: «Це перевірка коректності прибирання символів типу: крапка, кома, тире – та інших!»</li> <li>4. Натиснути на кнопку «Видалити стоп-символи»</li> <li>5. Натиснути на кнопку «Перевести текст у нижній регістр»</li> </ol>		<p>З речення повинні прибратись усі стоп-символи</p> <p>Всі слова повинні стати у нижньому регістрі</p>
<b>Результат виконання тест-кейсу:</b> перевірку пройдено успішно.		

Як видно з рисунку 3.11 – тест пройдено успішно. Наступним тестовим випадком буде перевірка коректності формування лексичних n-грам для вкладки «Модуль побудови лексичних n-грам». Для виконання тесту необхідно виконати кроки з таблиці 3.4.

Таблиця 3.4 Тест-кейс VR0004

<b>Тест-кейс ID:</b> VR0004	<b>Пріоритет:</b> 1	<b>Створено:</b> 17.05.2022, Підлапушний В.О.
<b>Назва:</b> Перевірка коректності формування лексичних n-грам для вкладки «Модуль побудови лексичних n-грам»		
<b>Кроки</b>		<b>Очікуваний результат</b>
<ol style="list-style-type: none"> <li>1. Запуск програми</li> <li>2. Обрати вкладку «Модуль формування масиву коректних слів»</li> <li>3. Вставити в поле «Текстовий тестовий контент» речення: «Це перевірка коректності прибирання символів типу: крапка, кома, тире – та інших!»</li> <li>4. Натиснути на кнопку «Видалити стоп-символи»</li> <li>5. Натиснути на кнопку «Перевести текст у нижній регістр»</li> <li>6. Натиснути кнопку «Побудувати навчальний масив коректних слів».</li> <li>7. Натиснути кнопку «Сформувати n-грами»</li> <li>8. Перейти на форму «Модуль побудови матриці лексичних n-грам» та перевірити результат</li> </ol>		<p>З речення повинні прибратись усі стоп-символи</p> <p>Всі слова повинні стати у нижньому регістрі</p> <p>У таблиці «Навчальний масив коректних слів» заповнились слова</p> <p>На вкладці «Модуль побудови лексичних n-грам» відобразились у таблиці лексичні n-грами</p>
<b>Результат виконання тест-кейсу:</b> перевірку пройдено успішно.		

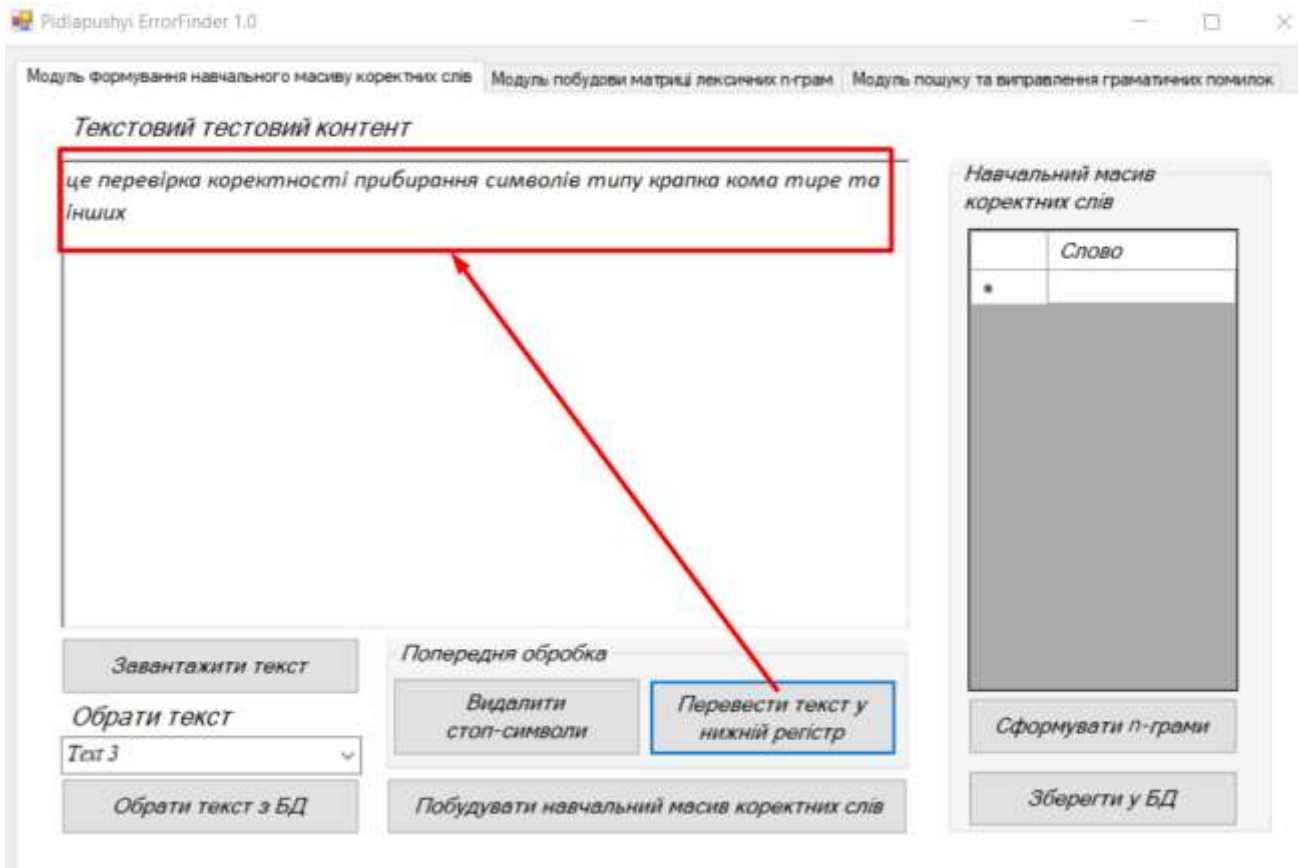


Рисунок 3.11 – Текст після натиснення на кнопку «Перевести текст у нижній регістр»

Після запуску програмного продукту користувачу потрібно виконати пункти, що вказані у таблиці 3.4, після виконання яких у програмі користувач побачить результат на першій вкладці у вигляді очищеного тексту в нижньому регістрі текстового поля «Текстовий тестовий контент» та сформованого «Навчального масиву коректних слів», а також на другій вкладці «Модуль побудови лексичних n-грам» буде сформовано лексичні n-грами для вкладки (рисунок 3.12).

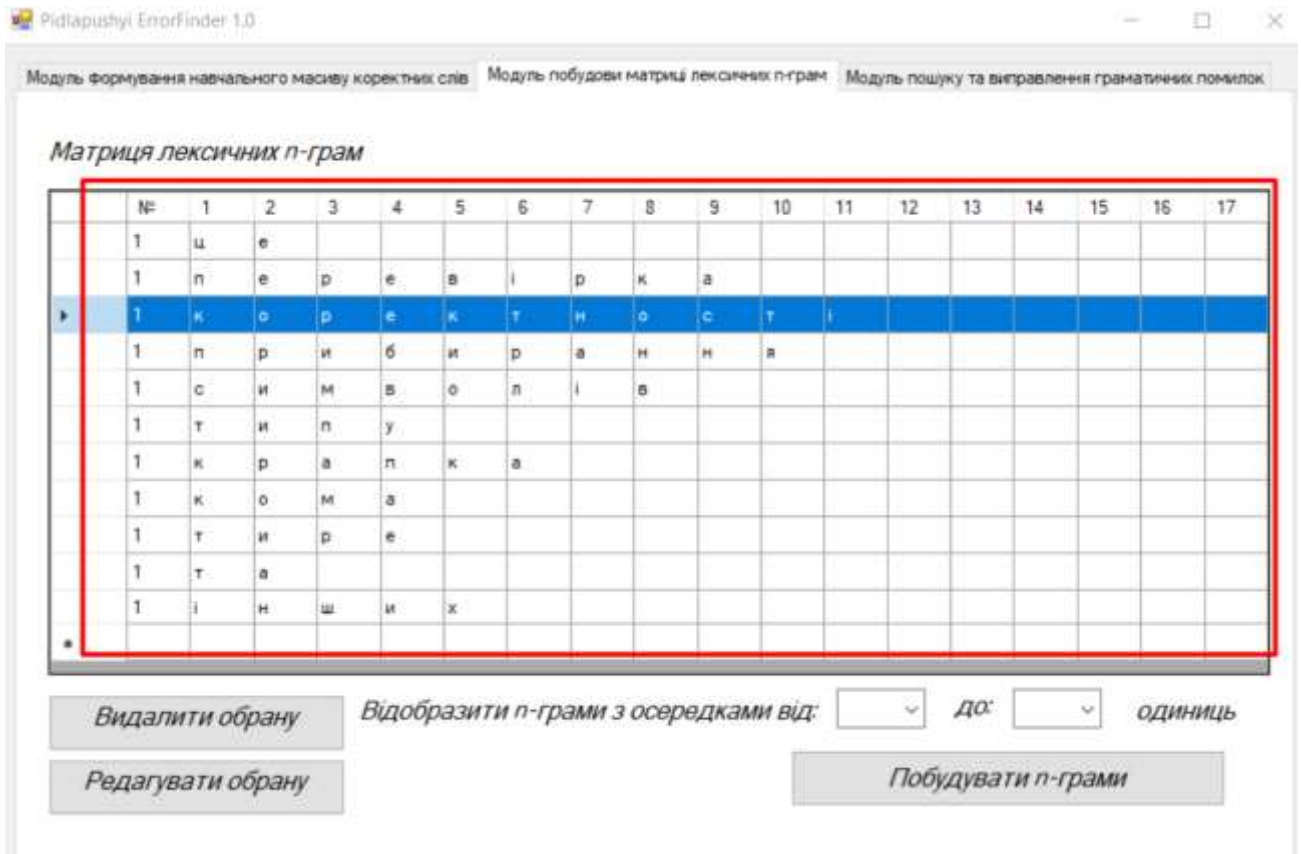


Рисунок 3.12 – Формування лексичних n-грам для вкладки «Модуль побудови лексичних n-грам»

Як видно з рисунка 3.12 – даний тест також виконано успішно. Отже, в процесі проведеного тестування програмного застосунку для автоматизованого виправлення граматичних помилок у цифрових текстах некоректно працюючих функцій виявлено не було.

### 3.4 Інструкція користувача

Після виконання користувацького тестування, потрібно створити інструкцію користувача. При запуску програмного застосунку для автоматизованого виправлення граматичних помилок у цифрових текстах користувач бачить першу вкладку – «Модуль формування масиву коректних слів» (рисунок 3.13).

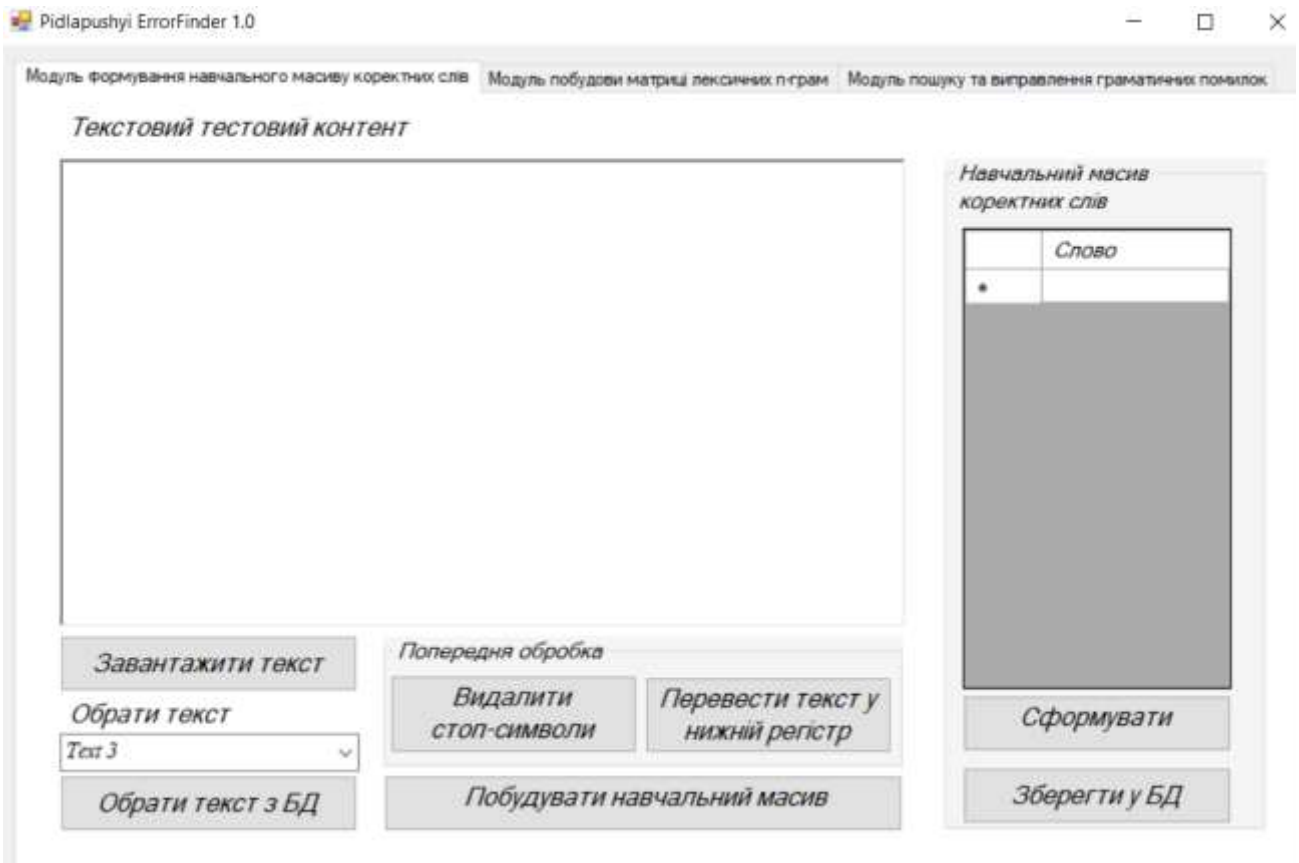


Рисунок 3.13 – Стартова вкладка застосунку

Для початку роботи користувач може завантажити текст з внесених у базу даних, для цього з переліку «Обрати текст» потрібно вибрати один із запропонованих (рисунок 3.14), або ж користувач може завантажити текст з обраного файлу, для чого йому необхідно натиснути кнопку «Завантажити текст» (рисунок 3.15).



Також користувач може просто вставити у текстове поле «Текстовий тестовий контент» власноруч написаний текст, або ж текст який його зацікавить з інтернету методом «ctrl+C» «ctrl+V». До завантаженого тексту будь-яким способом можна застосувати методи попередньої обробки. При натисненні кнопки «Видалити стоп-символи» текст стане очищеним від стоп-символів (рисунок 3.16)

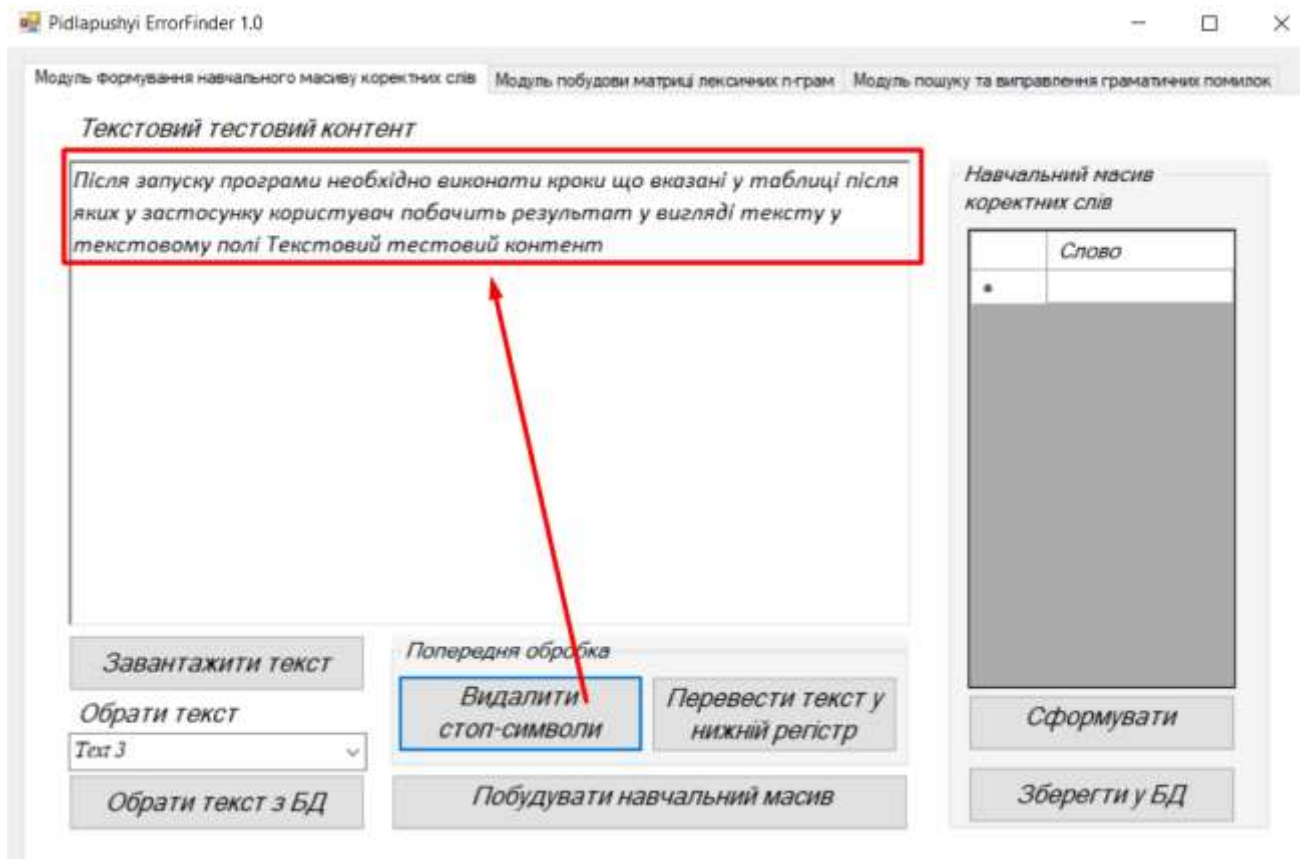


Рисунок 3.16 – Вибір тексту з ПК

Відповідно, при натисканні кнопки «Перевести текст у нижній регістр» весь текст буде приведено до нижнього регістру (рисунок 3.17). Для побудови навчального масиву слів необхідно натиснути кнопку «Побудувати навчальний масив слів», відповідний масив буде виведено у таблиці «Навчальний масив коректних слів» (рисунок 3.18).

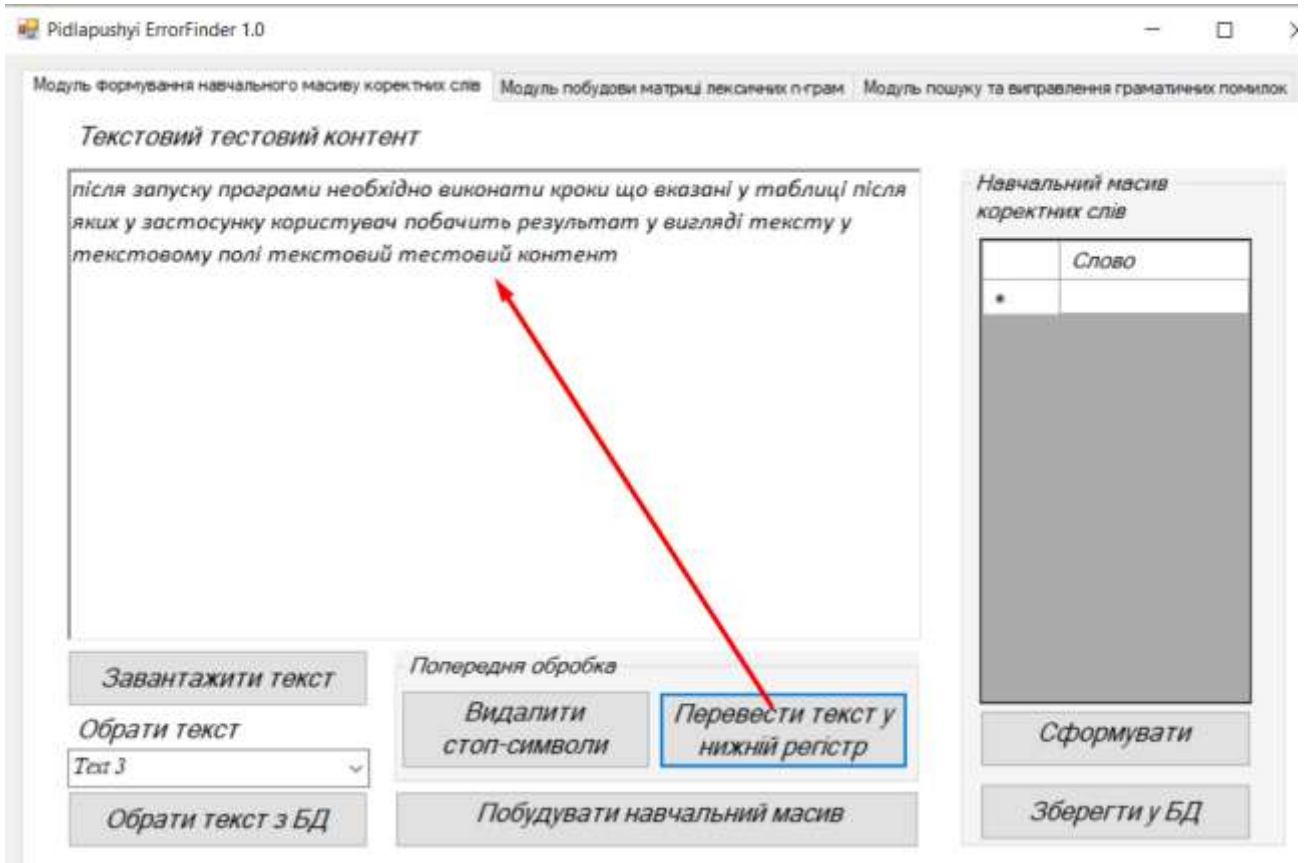


Рисунок 3.17 – Приведення тексту у нижній регістр

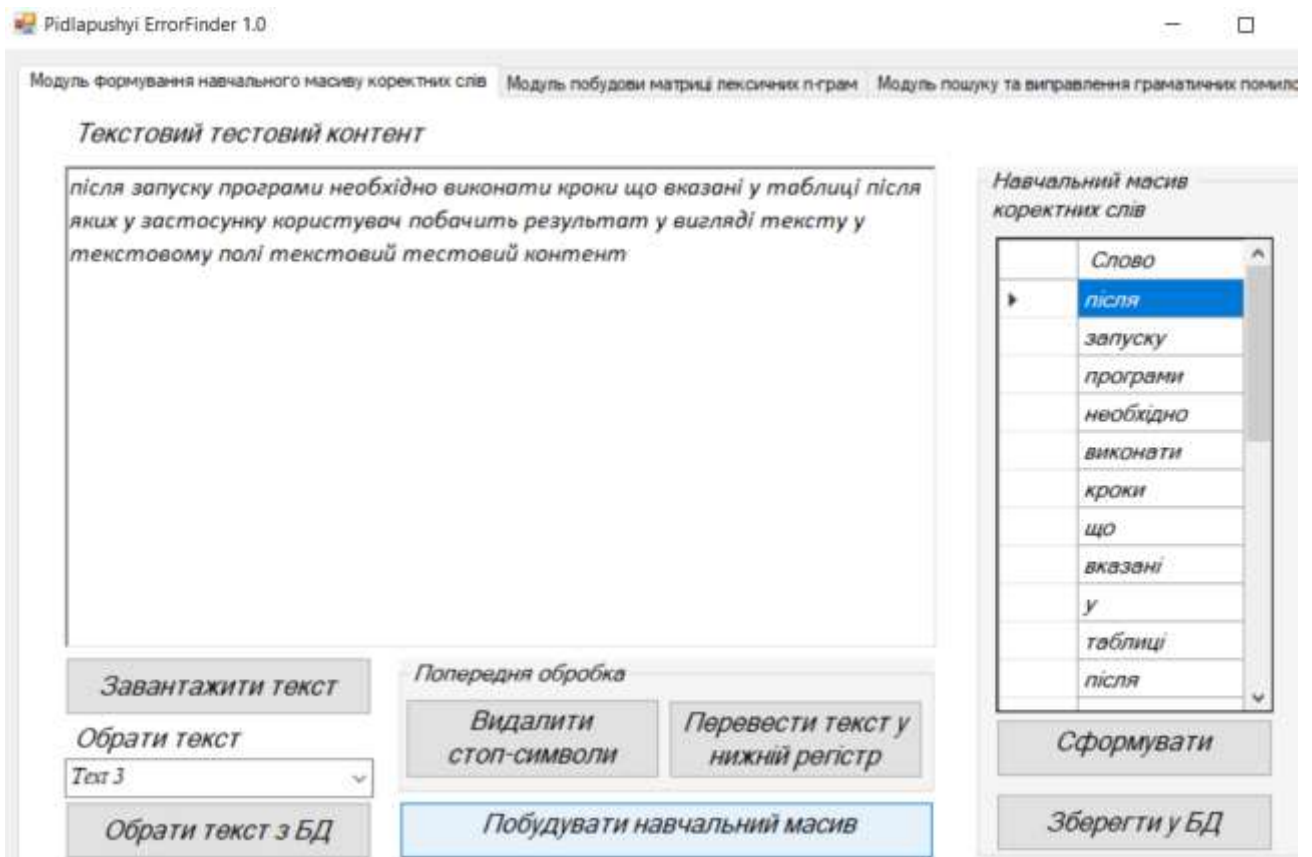


Рисунок 3.18 – Побудова навчального масиву

Масив коректних слів можна відобразити у формі n-грам. Для цього потрібно натиснути на кнопку «Сформувати n-грами», та перейти на другу вкладку системи. Відповідно, для другої вкладки є можливість переглянути всі n-грами (рисунок 3.19), або ж відфільтрувати їх за мінімальною і максимальною кількістю символів.

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	п	і	с	л	я												
1	і	з	а	п	у	с	к	у									
1	п	р	о	г	р	а	м	и									
1	н	е	о	б	х	і	д	н	о								
1	в	и	к	о	н	а	т	и									
1	к	р	о	к	и												
1	щ	о															
1	в	к	а	з	а	н	і										
4	у																

Рисунок 3.19 – Лексичні n-грами

Для фільтрування n-грам за розміром у відповідних полях «Відобразити n-грами з осередками від:» і «до» потрібно обрати потрібну розмірність та натиснути кнопку «Побудувати n-грами». Результат буде як на рисунку 3.20. Також на цій вкладці є можливість редагування n-грам, або їх видалення. Для редагування усередині таблиці потрібно внести необхідні зміни та натиснути кнопку «Редагувати обрану», для видалення необхідно курсор поставити у відповідній стрічці таблиці та натиснути кнопку «Видалити обрану».

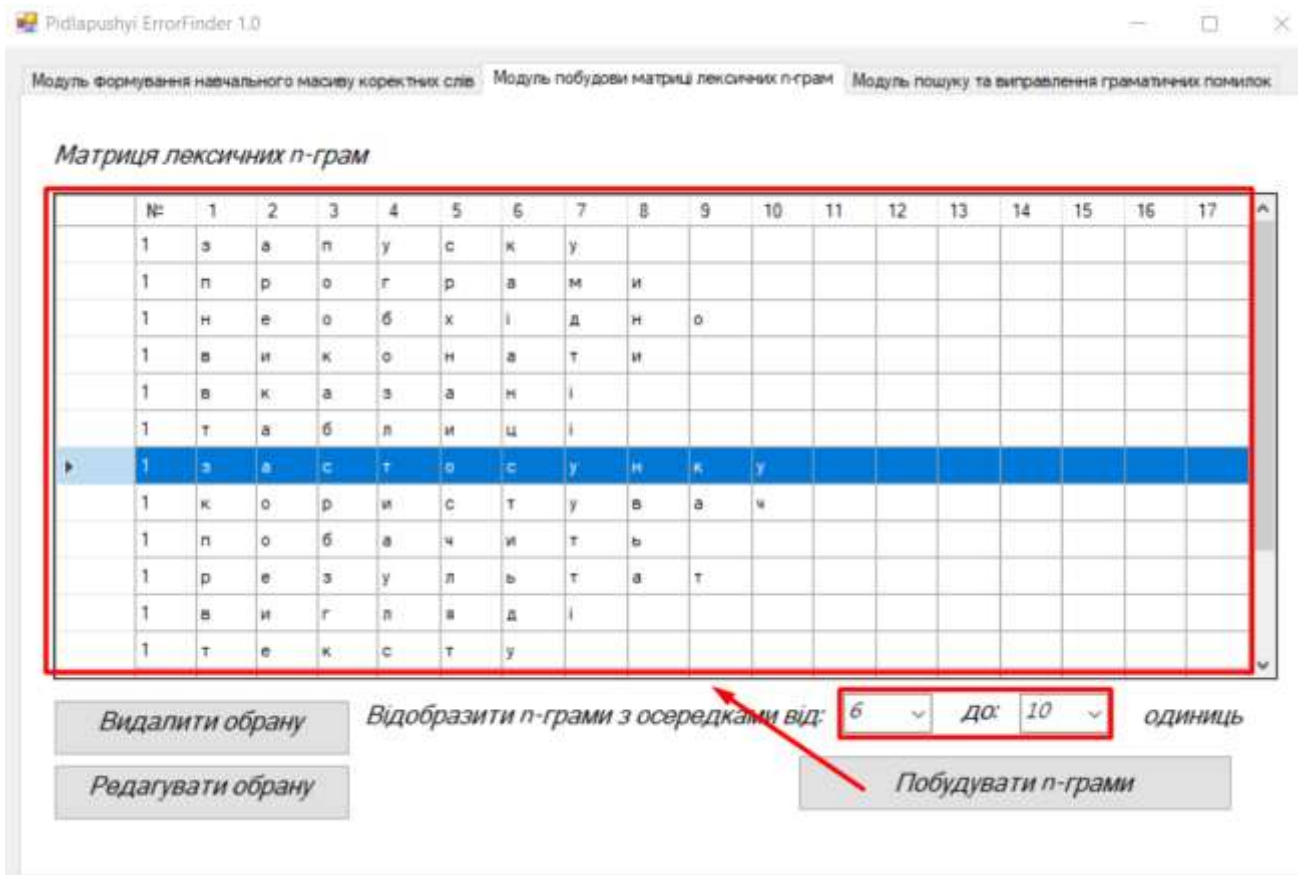


Рисунок 3.20 – Лексичні n-грами з фільтрацією

Надалі можна переходити до третьої вкладки – «Модуль пошуку та виправлення граматичних помилок». Тут за аналогією до першої вкладки можна завантажити текст з файлу або ж вставити з інтернету чи бази даних. Для перевірки потрібно натиснути кнопку «Перевірити текст», після чого у текстовому полі текст буде приведений до нижнього регістру та прибрані розділові знаки, а всі слова будуть перевірені з коректними. У випадку неспівпадіння з еталонами, будуть підібрані можливі варіанти для заміни (рисунок 3.21).

Для заміни на обране слово з «Правильних варіантів для заміни» необхідно натиснути кнопку «Замінити», і слово буде замінене у текстовому полі «Тестовий контент для перевірки».

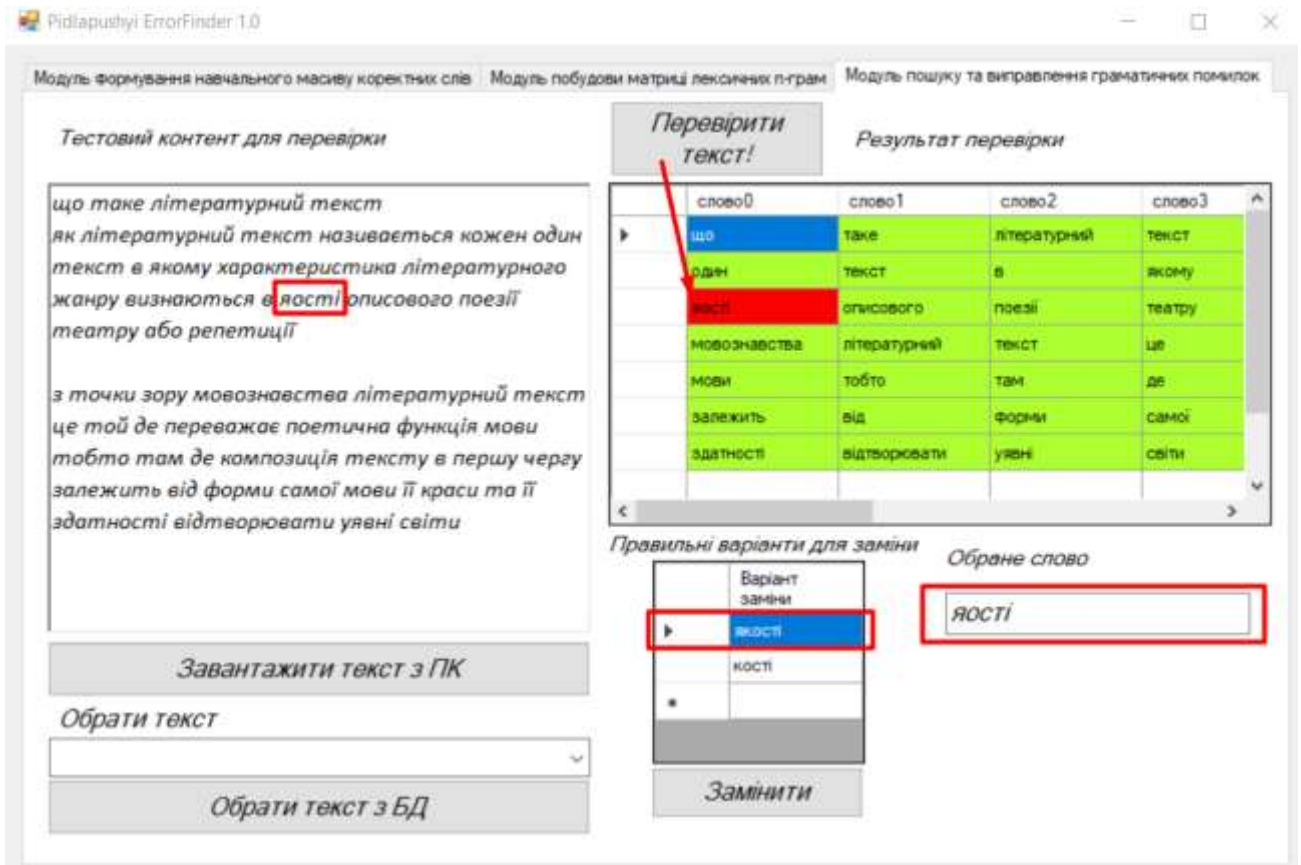


Рисунок 3.21 – Результуючий екран застосування

Отже, для більш зручного використання даного програмного продукту було створено інструкцію користувача.

## Висновки

Метою кваліфікаційної роботи бакалавра було створення, а також прикладна програмна реалізація методу автоматизованого виправлення граматичних помилок у цифрових. Для досягнення були вирішені такі задачі:

1. Проведено аналіз предметної області, зокрема огляд теоретичних підходів до розв'язання задачі виправлення граматичних помилок у цифрових текстах, а також проведено аналіз існуючих програмних рішень.

2. Розробити метод автоматизованого виправлення граматичних помилок у цифрових текстах, на базі якого створити інформаційну структуру програмного застосунку.

3. На базі проведеного аналізу існуючих засобів створення програмного забезпечення обрано комбінацію засобів для розробки програмної реалізації методу автоматизованого виправлення граматичних помилок у цифрових текстах, а саме платформу .NET та мову програмування C#.

4. Обраними засобами розробки створити та протестувати прикладну програмну реалізацію методу автоматизованого виправлення граматичних помилок у цифрових текстах.

Розроблена інформаційна система автоматизованого виправлення граматичних помилок у цифрових текстах на платформі .NET виконує такі основні функції:

- Одержання текстового навчального контенту.
- Початкова обробка тексту навчального контенту (стоп-знаки, цифри, регістр).
- Пошук меж слів (роздільники – прогалини тощо).
- Побудова навчального масиву коректних слів.
- Побудова лексичного масиву слів за навчальним масивом коректних слів.
- Обрахунок значень кількості появ кожного з слів у лексичному масиві.
- Побудова матриці лексичних n-грам за даними лексичного масиву слів.

- Відображення n-грам з обраною користувачем кількістю осередків.
- Одержання тексту для перевірки.
- Початкова обробка тексту для перевірки.
- Побудова текстового масиву для перевірки.
- Перевірка наявності кожного з слів перевірного текстового масиву у матриці лексичних n-грам.
- Візуальне виділення правильних слів, що наявні в матриці лексичних n-грам.
- Візуальне виділення неправильних слів, що відсутні в матриці лексичних n-грам.
- Формування й сортування масивів правильних слів-альтернатив до кожного з неправильних слів.
- Заміна користувачем неправильних слів на обрані слова із масивів правильних слів-альтернатив.

## Перелік посилань

1. Wikipedia. Digital poetry. URL: [https://en.wikipedia.org/wiki/Digital\\_poetry](https://en.wikipedia.org/wiki/Digital_poetry)
2. АККет. Як перевести цифровий текст у рукописний. URL: <https://akket.com/raznoe/210318-kak-perevesti-tsifrovoj-tekst-v-rukopisnyj.html>
3. Wikipedia. Слово. URL: <https://uk.wikipedia.org/wiki/Слово>
4. МійКлас. Словосполучення. Головне і залежне слово. Відмінність словосполучення від слова і речення. URL: <https://miyklas.com.ua/p/ukrainska-mova/5-klas/vidomosti-z-sintaksisu-i-punktuatciyi-14562/slovospoluchennia-39535/re-dde93220-a7e1-4d47-8f6d-68a1da6b931b>
5. Асоціація репетиторів України. Речення. URL: <https://repetitor.org.ua/rechenna-2>
6. Wikipedia. Граматика. URL: <https://uk.wikipedia.org/wiki/Граматика>
7. ZNO. Стили сучасної української літературної мови. URL: <https://znoclub.com/mova-ta-literatura/536-stili-suchasnoji-ukrajinskoji-literaturnoji-movi.html>
8. Wikipedia. Орфографія. URL: <https://uk.wikipedia.org/wiki/Орфографія>
9. Wikipedia. Орфографічний словник. URL: [https://uk.wikipedia.org/wiki/Орфографічний\\_словник](https://uk.wikipedia.org/wiki/Орфографічний_словник)
10. ВУКІ. Помилки в українській мові, які можуть все зіпсувати на ЗНО. URL: <https://buki.com.ua/news/pomylky-v-ukrayinskiy-movi-yaki-mozhut-vse-zipsuvaty-na-zno-1/>
11. Studfile. Українська орфографія. Орфографічні помилки. URL: <https://studfile.net/preview/5118416/page:25/>
12. Шляхова В. Літературне редагування у системі фахової підготовки філологів. Культура слова №74, 2011. URL: <http://dspace.nbuv.gov.ua/bitstream/handle/123456789/37162/27-Shljahova.pdf?sequence=1>

13. Домбровська Я. Особливості редакторської майстерності філологів у видавничо-поліграфічній діяльності. Теоретична і дидактична філологія. Серія «Філологія». Випуск 32. 2020. URL: <http://ephsheir.uhsp.edu.ua/bitstream/handle/8989898989/4689/5.pdf?sequence=1&isAllowed=y>
14. Wikipedia. N-грама. URL: <https://uk.wikipedia.org/wiki/N-грама>
15. Кузьма К.Т.. Інформаційна технологія оцінки рівня подібності рядків на основі методу n-грам. Вчені записки ТНУ імені В.І. Вернадського. Серія: технічні науки. Том 31(70) Ч.1 №6. 2020. URL: [https://tech.vernadskyjournals.in.ua/journals/2020/6\\_2020/part\\_1/18.pdf](https://tech.vernadskyjournals.in.ua/journals/2020/6_2020/part_1/18.pdf)
16. Багрій Р.О. Система прогнозування слів для альтернативної комунікації. Herald of Khmelnytskyi national university, Issue 4, 2017 (251). URL: <http://elar.khmnu.edu.ua/jspui/bitstream/123456789/5663/1/15.pdf>
17. Шингалов Д.В., Трищ О.В., Минайленко Р.М. Методи автоматичного аналізу настроїв в соціальних мережах. II Міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології». Центральноукраїнський національний технічний університет, м. Кропивницький, 2017. URL: <http://dspace.kntu.kr.ua/jspui/bitstream/123456789/7489/1/Methods%20of%20automatic%20analysis.pdf>
18. Лисенко С.М., Щука Р.В. Аналіз методів виявлення шкідливого програмного забезпечення в комп'ютерних системах. Вісник Хмельницького національного університету, №2, 2020 (283). URL: <http://elar.khmnu.edu.ua/jspui/bitstream/123456789/9168/1/ЩУКА.pdf>
19. LanguageTool. Покращуйте свої тексти миттєво. URL: <https://languagetool.org/uk>
20. Офіційний сайт української мови. Пошук помилок. URL: <https://ukrainskamova.com/index/wiki/0-30>
21. Мова – ДНК нації. Перевірка тексту. URL: <https://ukr-mova.in.ua/perevirka-tekstu>

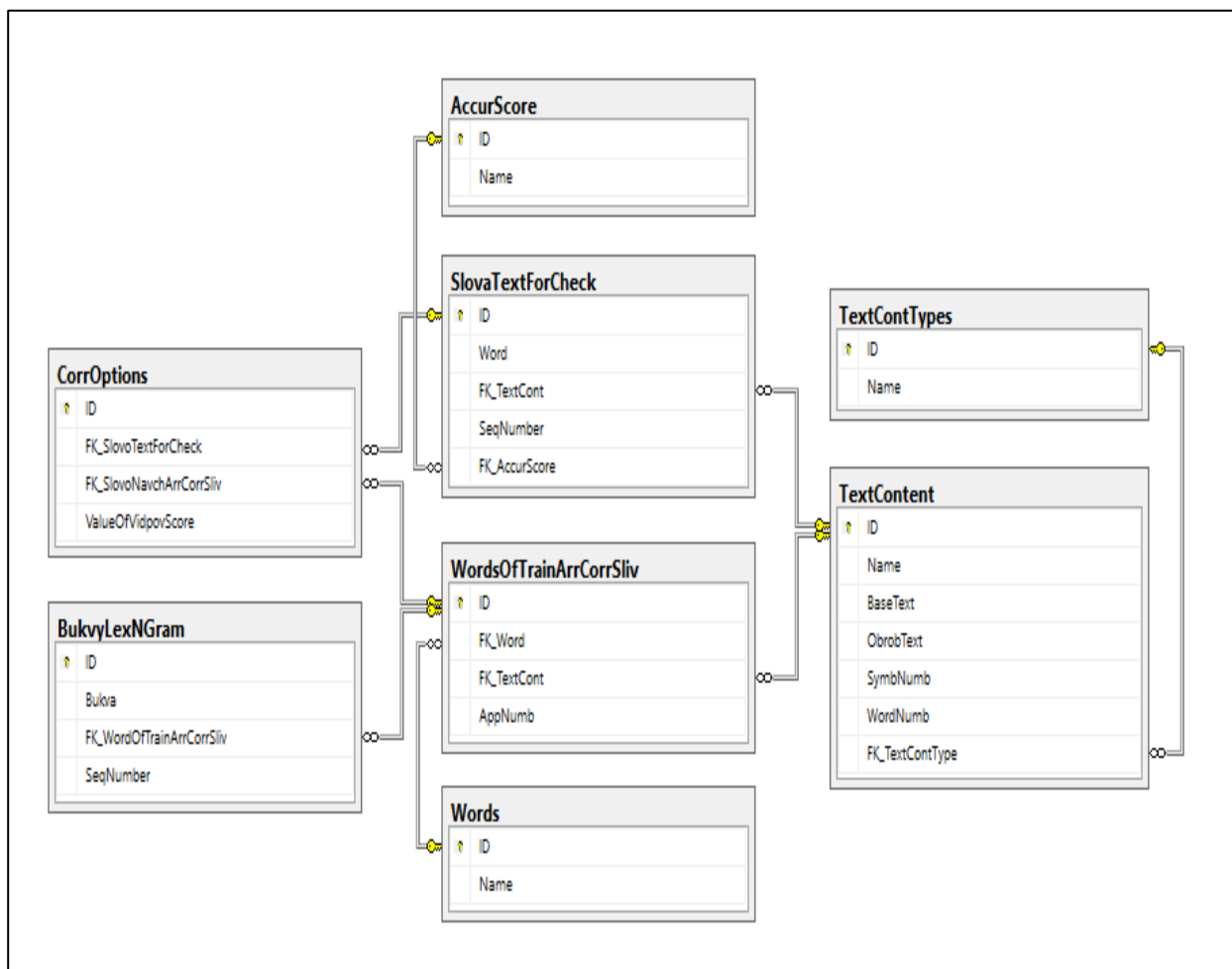
22. OnlineCorrector. Як це працює? URL:  
<https://onlinecorrector.com.ua/uk/home/>
23. Microsoft. Перевірка орфографії та граматики в Office. URL:  
<https://support.microsoft.com/uk-ua/office/перевірка-орфографії-та-граматики-в-office-5cdec7-d81d-47de-9096-efd0ee909227>
24. Grammarly. Great Writing, Simplified. URL:  
<https://www.grammarly.com/>
25. Microsoft. Grammarly: Grammar Checker and Writing App. URL:  
<https://microsoftedge.microsoft.com/addons/detail/grammarly-grammar-checker/cnlefmmeadmmdciolhbnfeacpfbkd?hl=ru-RU&gl=RU%3Fhl%3Dru-RU%3Fhl%3Dru-RU>
26. WebsitePlanet. Огляд сервісу GRAMMARLY. URL:  
<https://www.websiteplanet.com/uk/blog/огляд-сервісу-grammarly-для-перевірки-правопи>
27. Code Control. Mobile application against mobile website - advantages and disadvantages for your business. URL: <https://codecontrol.io/de/blog/mobile-app-vs-mobile-website-die-vor-und-nachteile-f%C3%BCr-dein-business>
28. Doctor Web Agency. I vantaggi delle applicazioni web app. URL:  
<https://www.doctor-web.it/i-vantaggi-delle-applicazioni-web-app/>
29. S.C.F. GROUP. Applicazioni web: cosa sono, vantaggi, svantaggi. URL:  
<https://www.scfgroup.it/applicazioni-web-vantaggi/>
30. YUHIRO. Werden Webanwendungen bald Desktop Applikationen komplett ersetzen? URL: <https://www.yuhiro.de/werden-webanwendungen-bald-desktop-applikationen-komplett-ersetzen/>
31. IDEAMOTIVE. C# vs Java: Which Is Better For Building Your Product? URL: <https://www.ideamotive.co/blog/c-sharp-vs-java-which-is-better-for-building-your-product>
32. Ncube. Pros and Cons of .NET Framework. URL:  
<https://ncube.com/blog/pros-and-cons-of-net-framework>

33. Productive C#. 15 reasons why you should learn C# in 2022. URL: <https://www.productivecsharp.com/why-you-should-learn-csharp/>
34. Wikipedia. Інтегроване середовище розробки [https://uk.wikipedia.org/wiki/Інтегроване\\_середовище\\_розробки](https://uk.wikipedia.org/wiki/Інтегроване_середовище_розробки)
35. Incredibuild Software. What is Visual Studio? URL: <https://www.incredibuild.com/integrations/visual-studio>
36. Microsoft. What's new in Visual Studio 2019. URL: <https://docs.microsoft.com/en-au/visualstudio/ide/whats-new-visual-studio-2019?view=vs-2022>
37. Vertabelo SA. MS SQL Server vs. PostgreSQL: Which Should You Choose for a New Project? URL: <https://learnsql.com/blog/ms-sql-vs-postgresql/>

# ДОДАТКИ

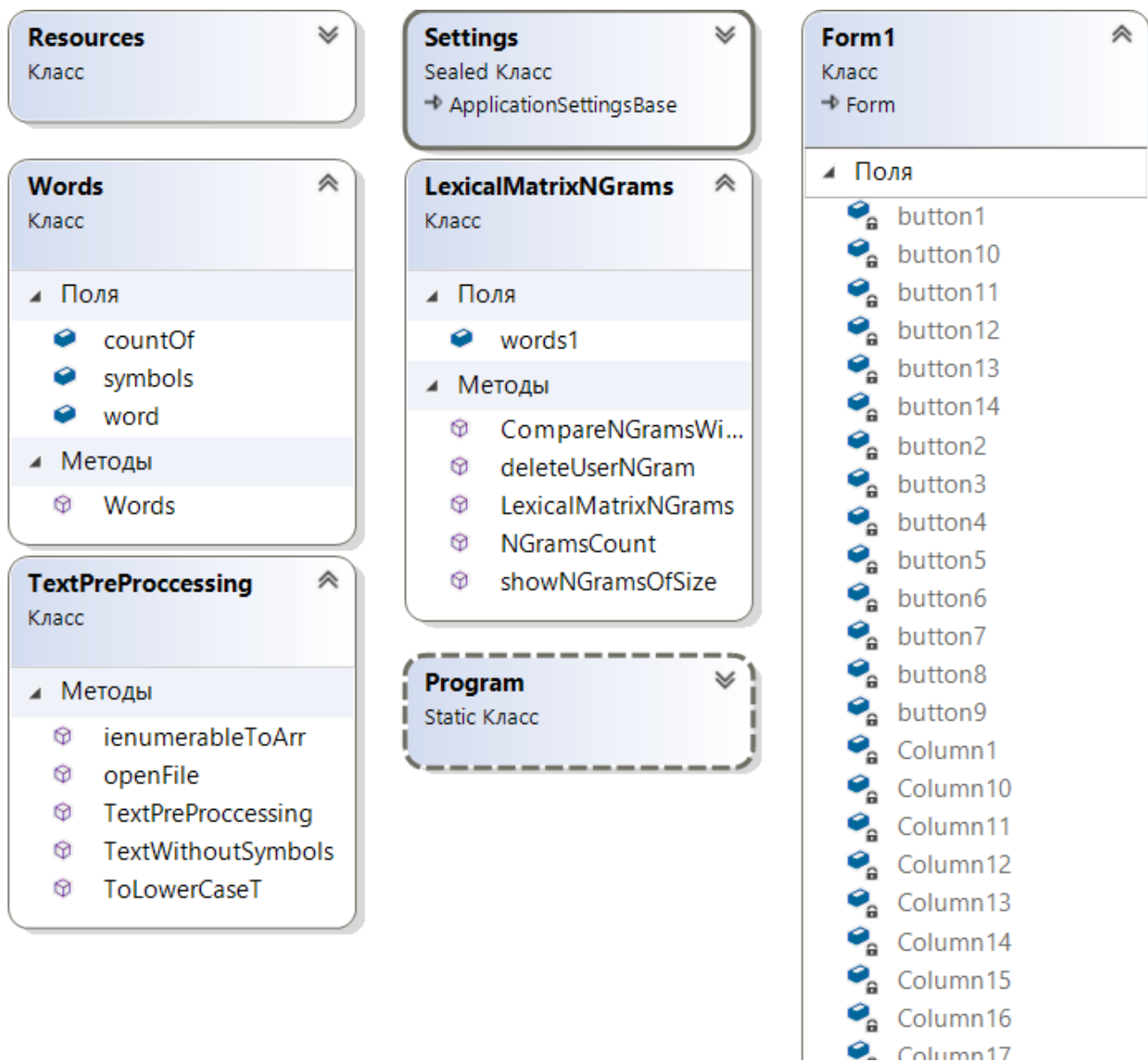
## Додаток А

## Структура бази даних інформаційної системи автоматизованого виправлення граматичних помилок у цифрових текстах



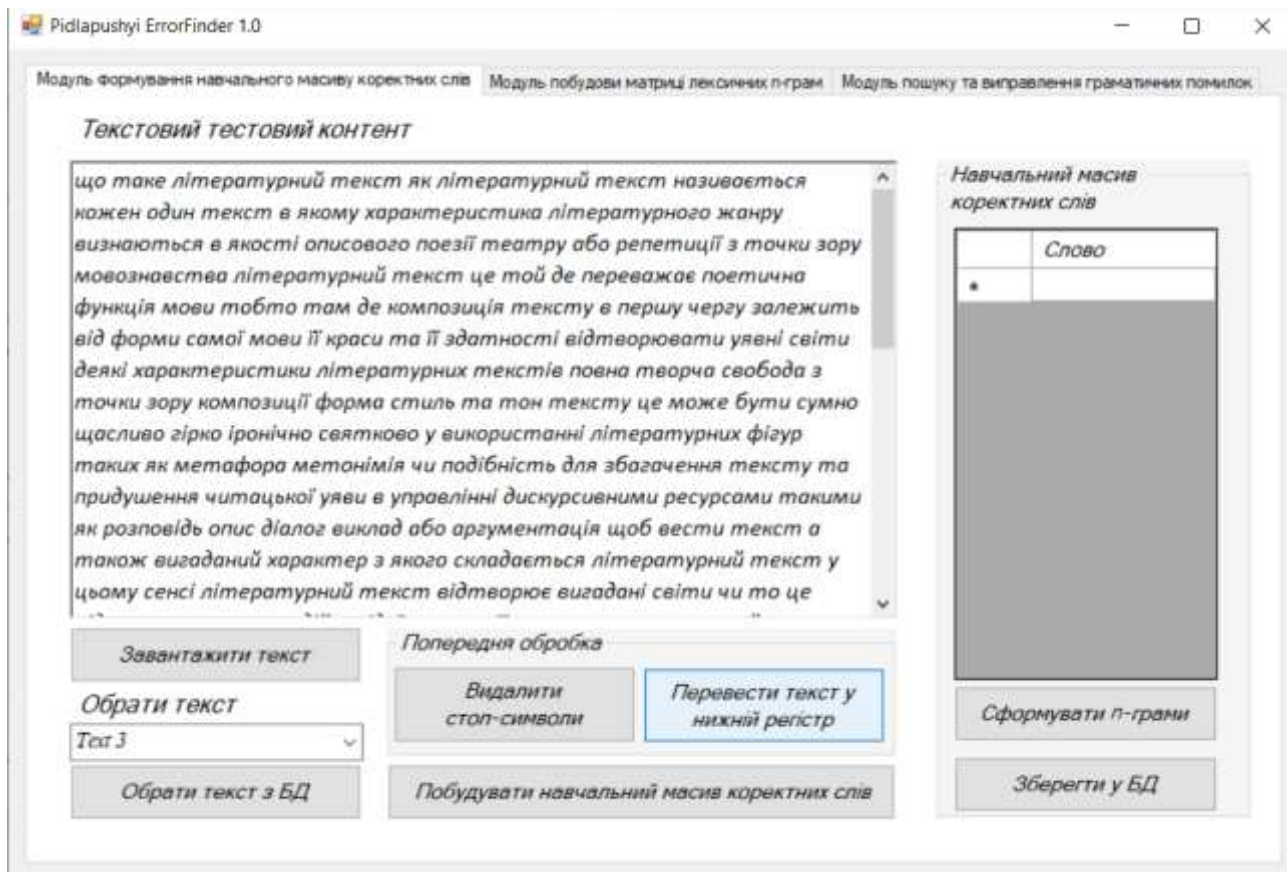
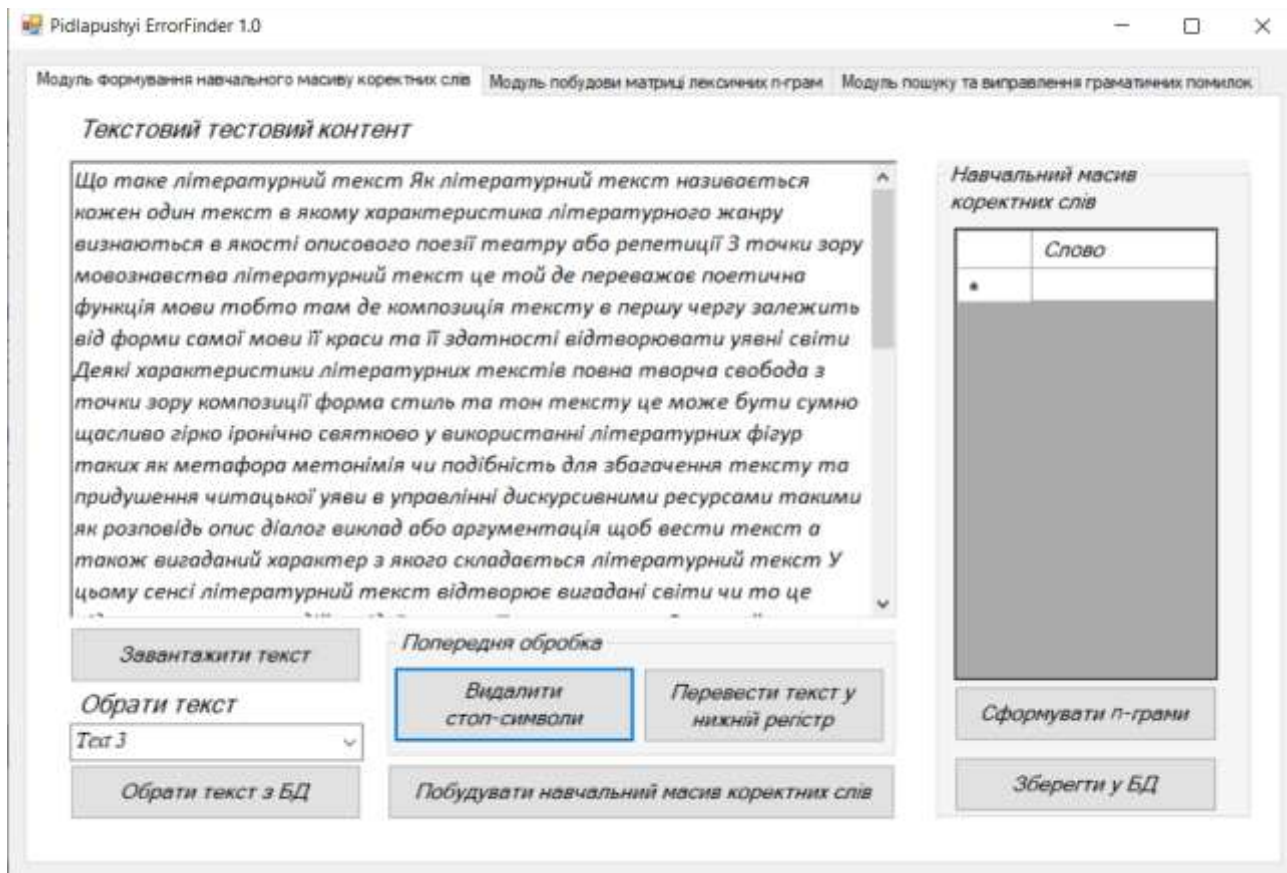
## Додаток Б

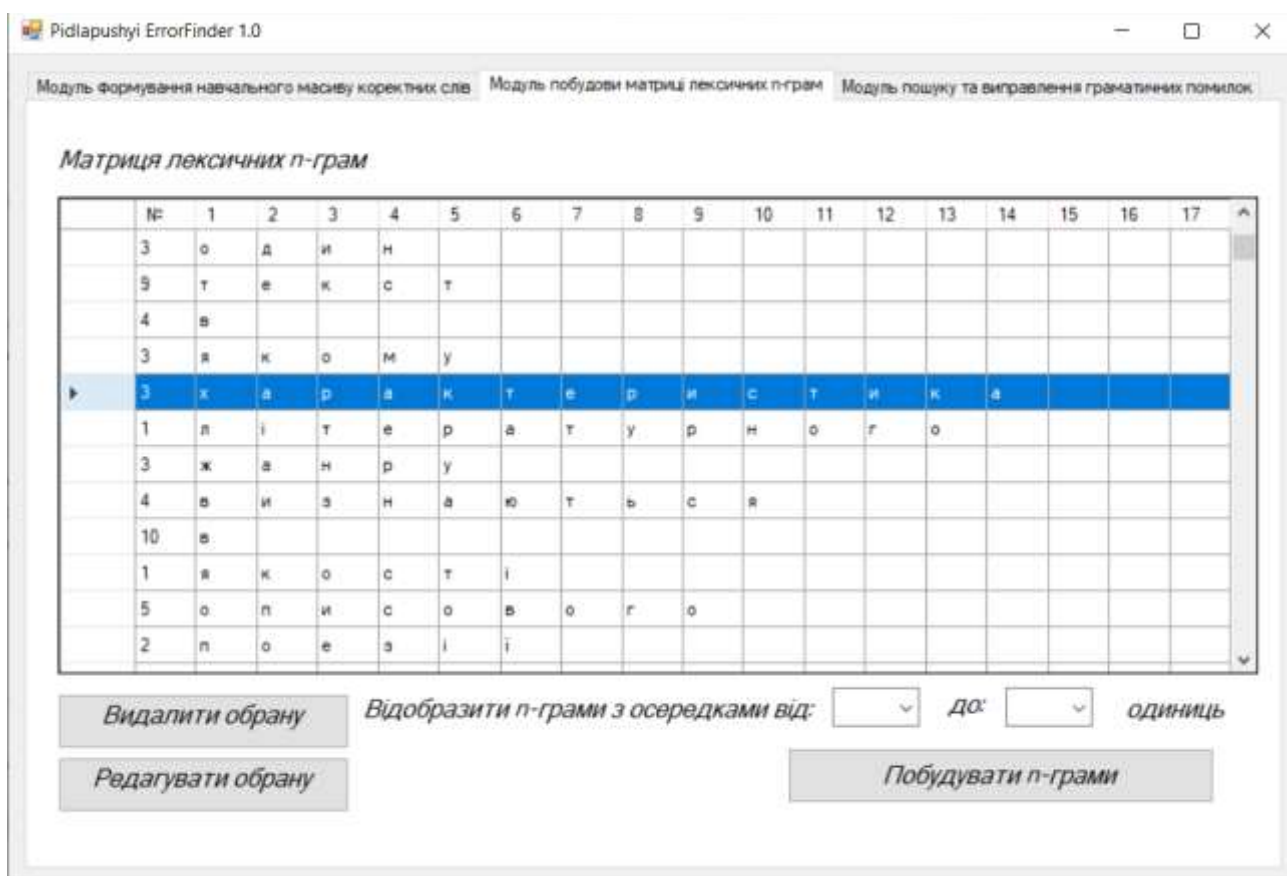
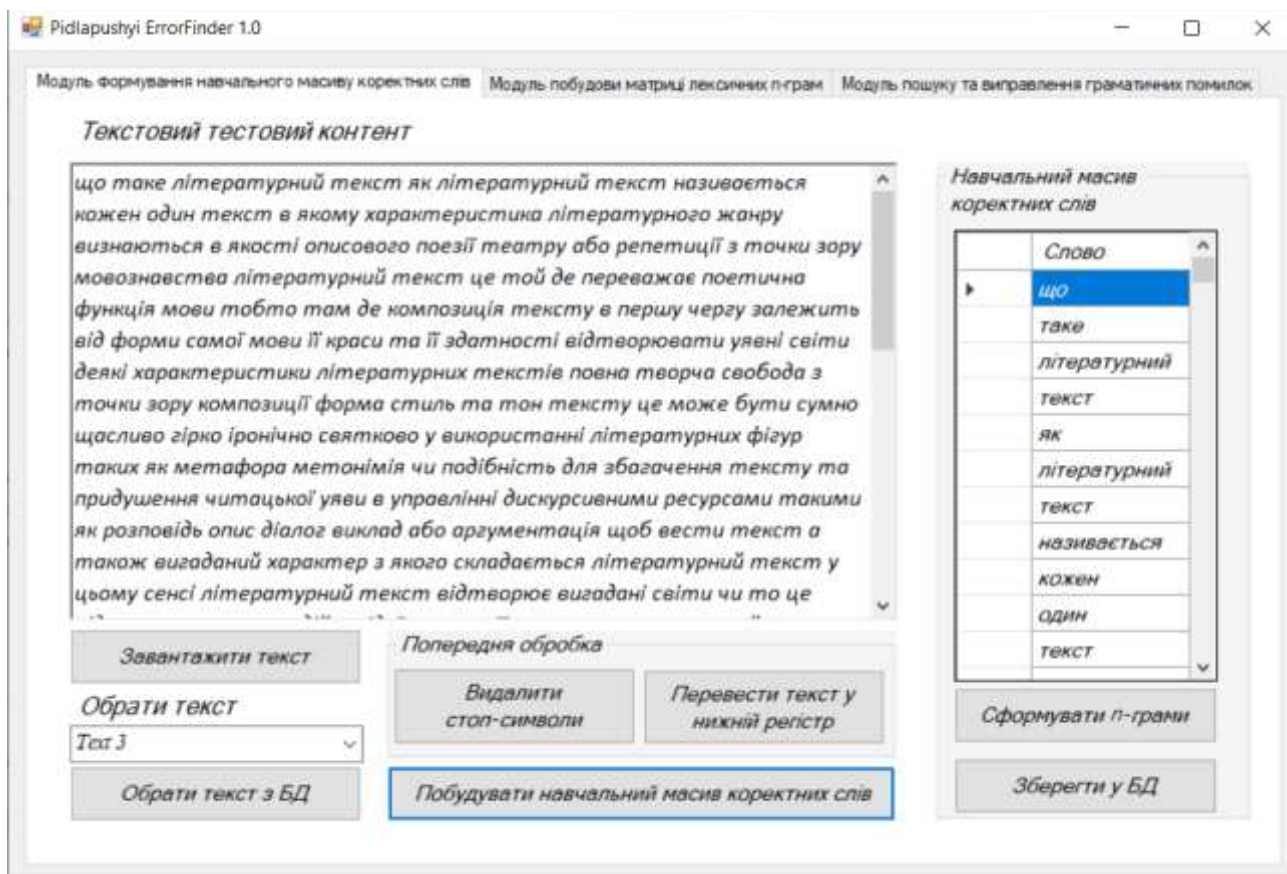
**Розгорнута структура класів інформаційної системи автоматизованого  
виправлення граматичних помилок у цифрових текстах**

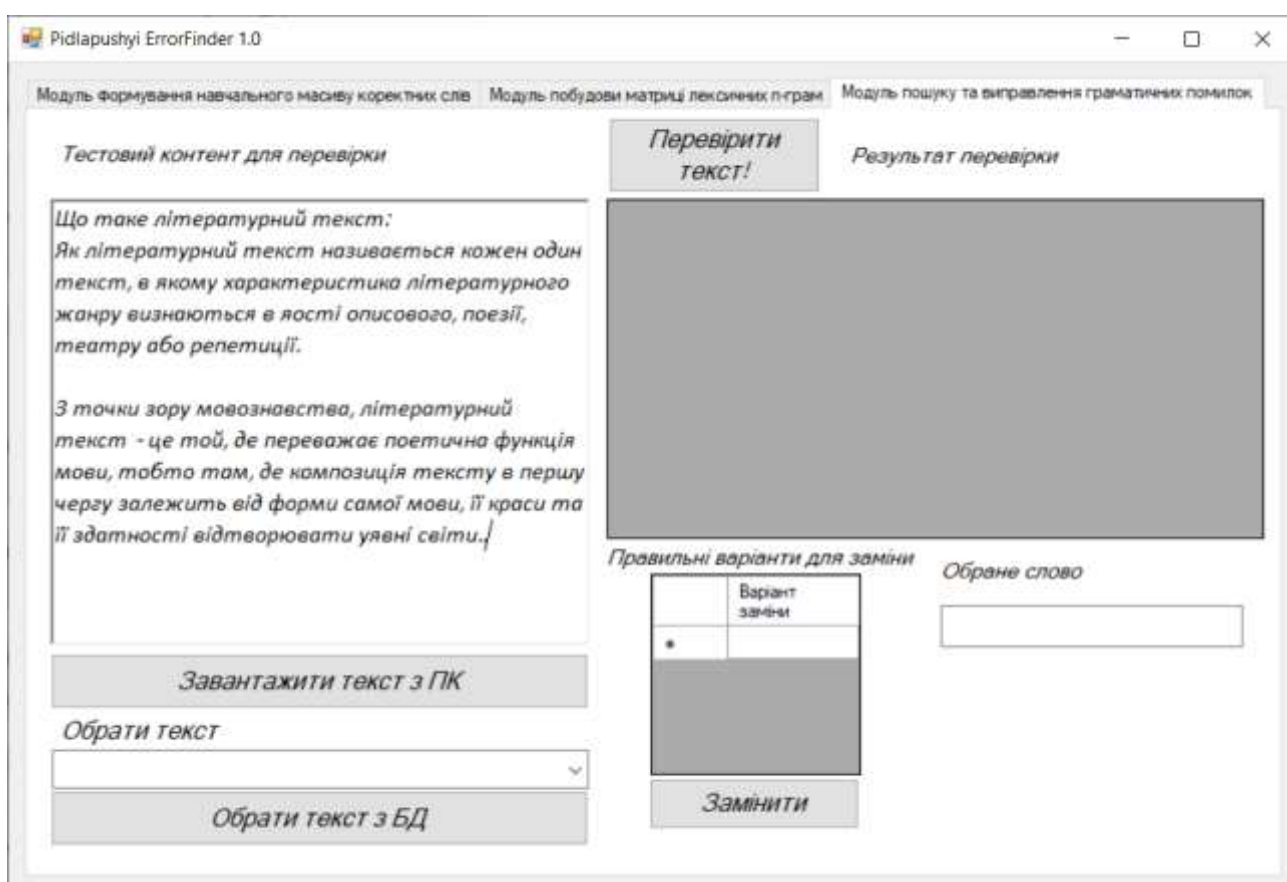


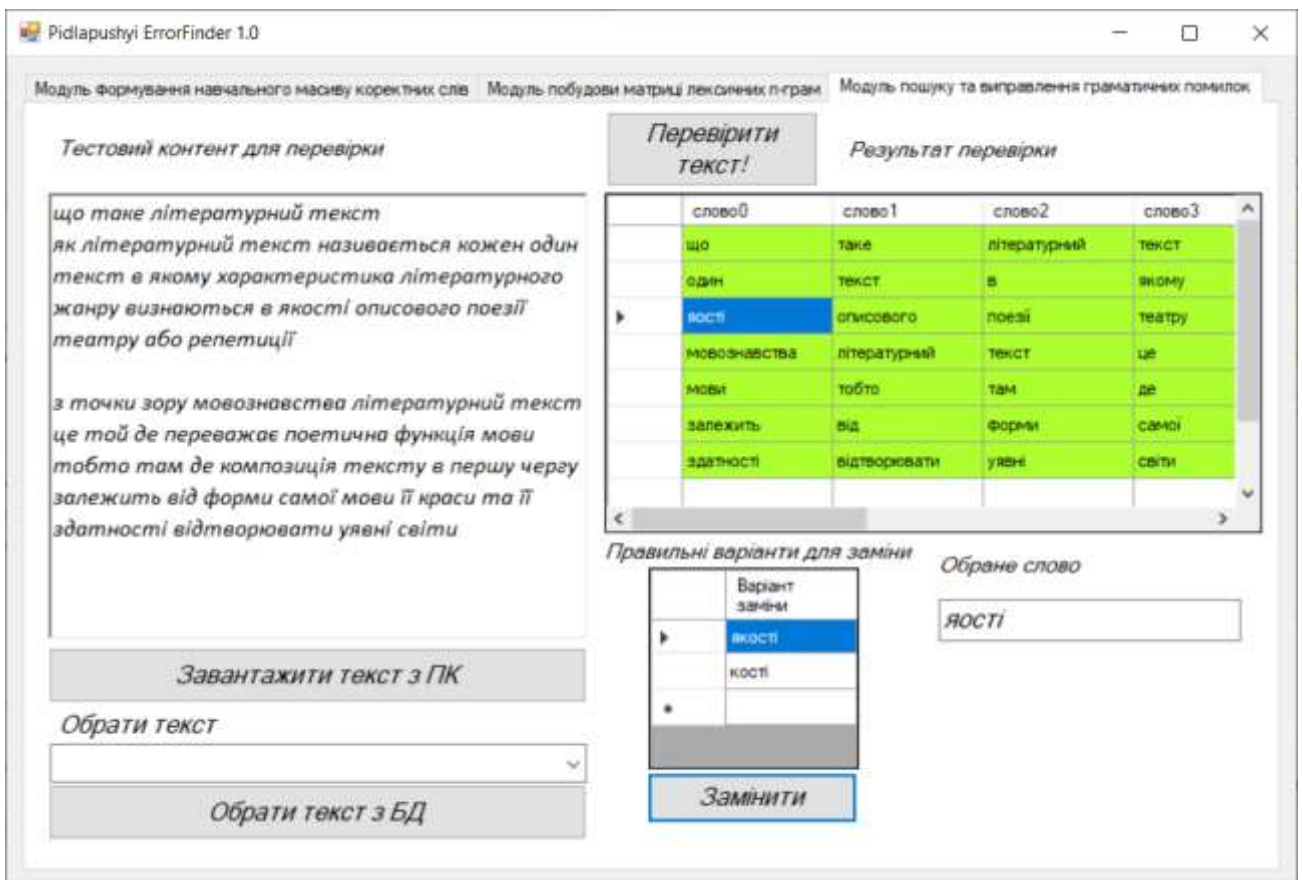
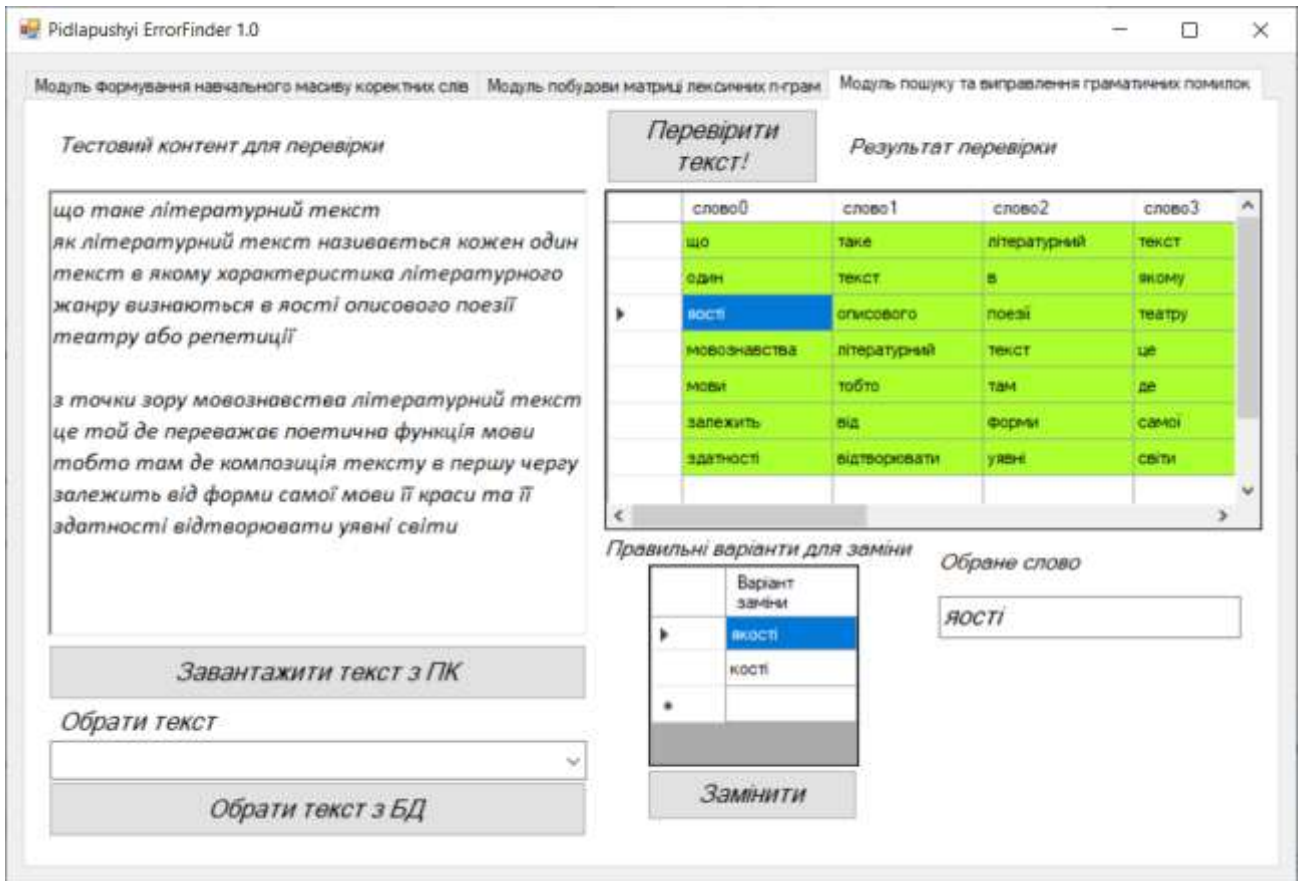
## Додаток В

### Світлини роботи програми









## Додаток Г

### Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

# МЕТОД АВТОМАТИЗОВАНОГО ВИПРАВЛЕННЯ ГРАМАТИЧНИХ ПОМИЛОК У ЦИФРОВИХ ТЕКСТАХ



**Виконав:**  
*студент 4 курсу, група КН-18-1*  
**Підлапушний Владислав Олегович**



**Керівник:**  
*викладач кафедри КН*  
**Собко Олена Віталіївна**

## Актуальність

З широким поширенням текстової інформації у вигляді цифрових документів особливо актуальним є перевірка та виправлення граматичних помилок з метою створення якісного контенту. Адже текст, в якому є помилки здатен відштовхувати читача та знижувати загальну затребуваність автора таких текстів.

Для створення цифрових документів, що не містять граматичних помилок, автор повинен володіти відповідними знаннями – правилами написання слів, їх зв'язку у реченнях, доцільного використання термінів, що належать до тих чи інших стилів мовлення.

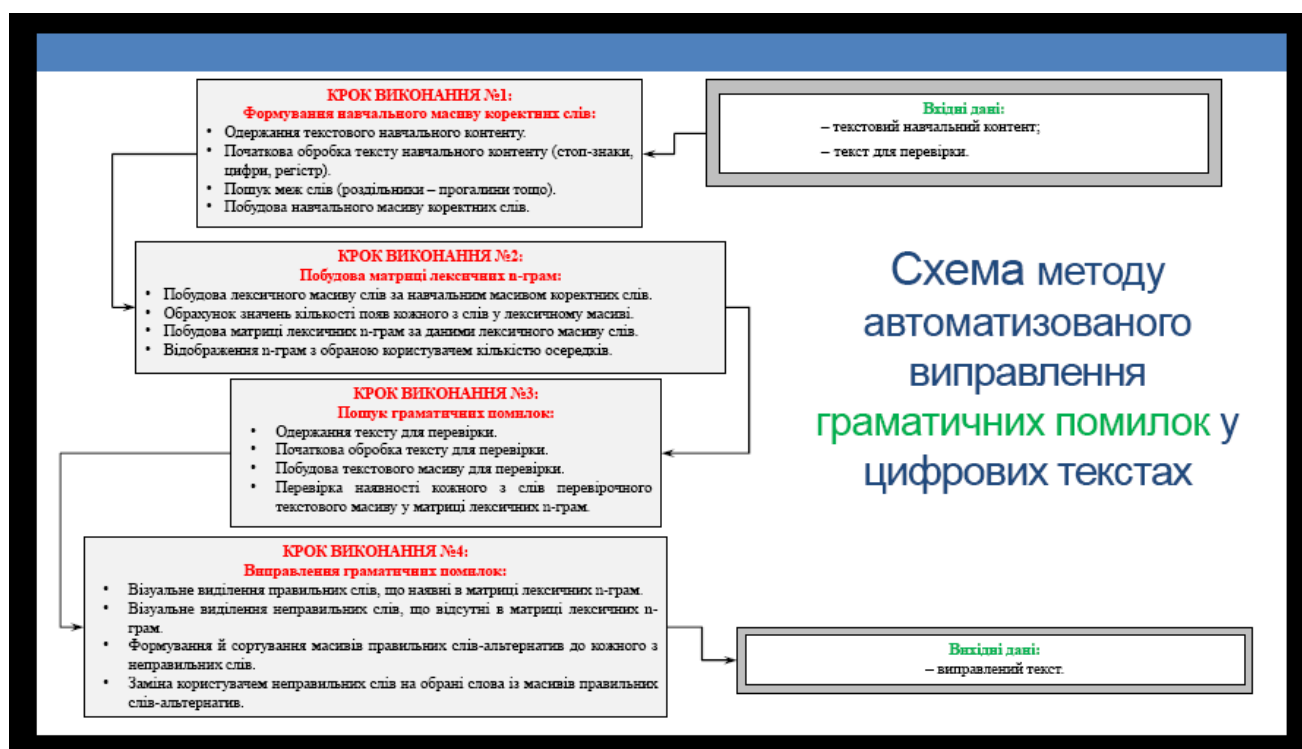
При створенні цифрових текстів автори можуть допускати не тільки граматичні помилки, а й стилістичні, лексичні, пунктуаційні. Допомогають знаходити та виправляти такі помилки редактори та коректори. Вони можуть проводити різного рівня редагування – від пошуку граматичних помилок, до стилістичних. Дана робота є доволі трудомістка, та вимагає великої бази знань для коректного виправлення помилок у цифрових текстах. Для проведення даної роботи існує ряд програмних застосунків, які дозволяють робити це автоматизовано. Кожен з них має свої переваги та недоліки, тому повністю покладатися на правильність роботи цих застосунків не варто.

Таким чином має сенс проведення дослідження щодо існуючих методів та засобів для автоматизованого виправлення граматичних помилок у цифрових текстах, а також розробка власного методу, що вирішує дану задачу.

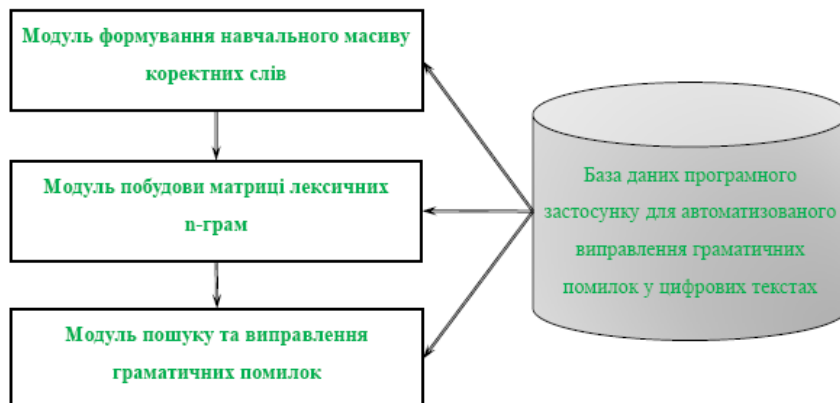
## Мета і задачі роботи

Метою кваліфікаційної роботи бакалавра є створення та прикладна програмна реалізація методу автоматизованого виправлення граматичних помилок у цифрових текстах. Для досягнення мети потрібно вирішити наступні задачі:

1. Провести аналіз предметної області, зокрема виконати огляд теоретичних підходів до розв'язання задачі виправлення граматичних помилок у цифрових текстах, а також провести аналіз існуючих програмних рішень.
2. Створити метод автоматизованого виправлення граматичних помилок у цифрових текстах, на базі якого створити інформаційну структуру програмного застосунку.
3. На базі проведеного аналізу існуючих засобів створення програмного забезпечення обрати комбінацію засобів для розробки програмної реалізації методу автоматизованого виправлення граматичних помилок у цифрових текстах
4. Обраними засобами розробки створити та протестувати прикладну програмну реалізацію методу автоматизованого виправлення граматичних помилок у цифрових текстах



## Схема програмного застосунку для автоматизованого виправлення граматичних помилок у цифрових текстах



## Схема модулю формування навчального масиву коректних слів



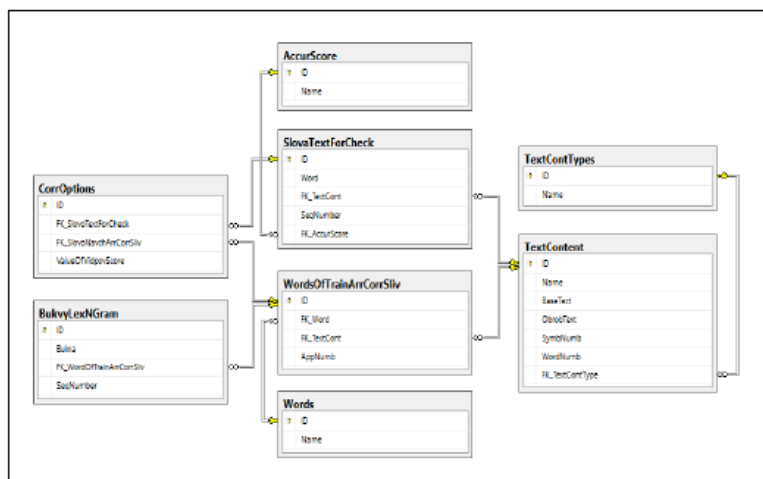
## Схема модулю побудови матриці лексичних n-грам



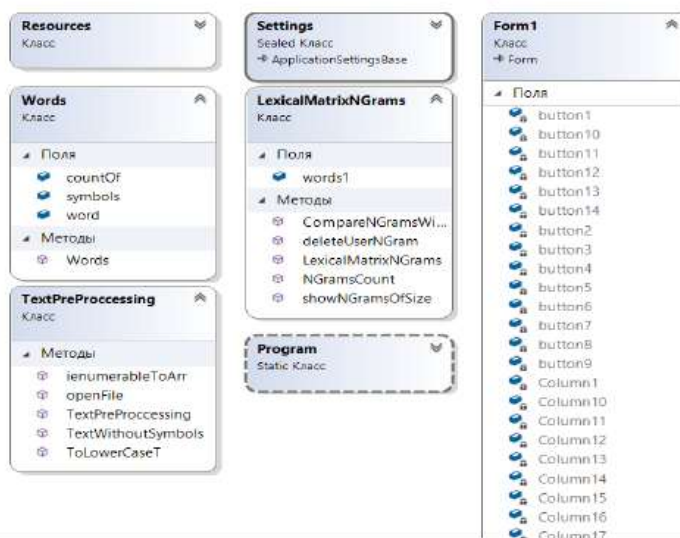
## Схема модулю пошуку та виправлення граматичних помилок

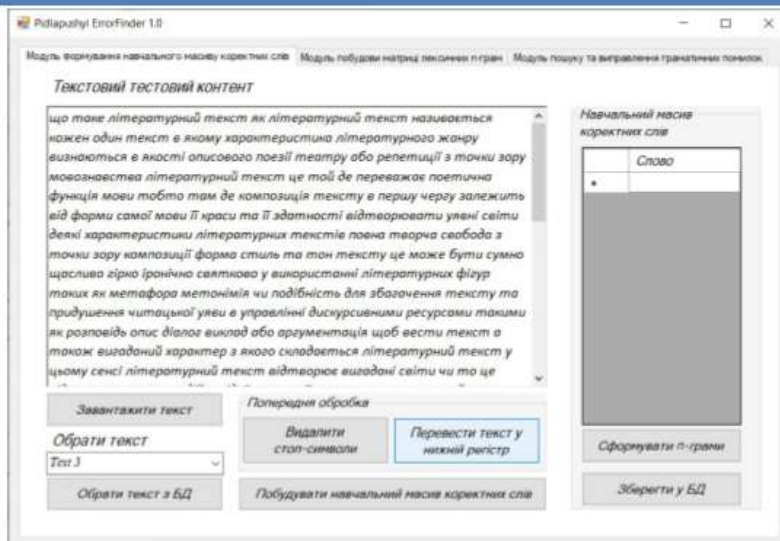


## Схема бази даних програмного застосунку для автоматизованого виправлення граматичних помилок у цифрових текстах

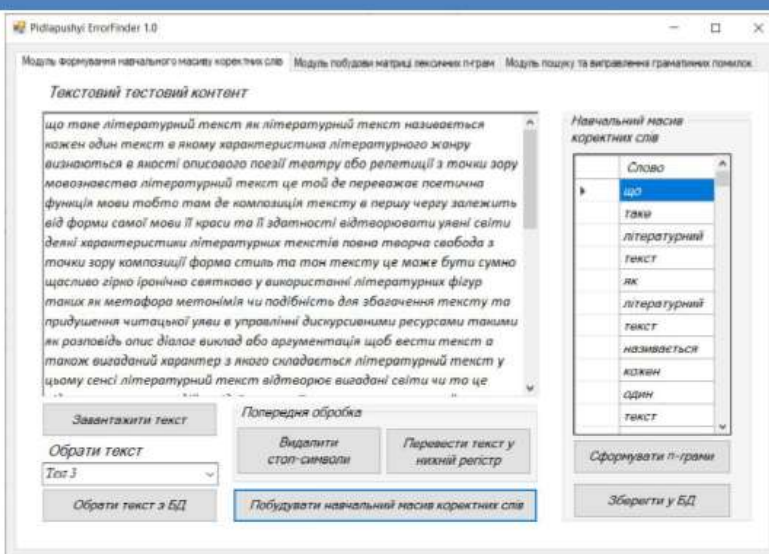


## Діаграма класів програмного застосунку для автоматизованого виправлення граматичних помилок у цифрових текстах





Вибір тексту



Побудова навчального масиву

Pidlapushnyi ErrorFinder 1.0

Модуль формування навчального масиву коректних слів Модуль побудови матриці лексичних n-грам Модуль пошуку та видалення граничних помилок

**Матриця лексичних n-грам**

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
5	о	п	и	с	о	в	о	г	о								
5	р	е	п	е	т	и	ц	і	і								
5	м	о	в	о	з	н	е	в	с	т	в	а					
4	л	і	т	е	р	а	т	у	р	н	и	я					
5	п	е	р	е	в	а	ж	а	с								
1	п	о	е	т	и	ч	н	а									
5	к	о	м	п	о	з	и	ц	і	я							
2	з	а	л	е	ж	и	т	ь									
2	з	а	д	а	т	н	о	с	т	і							
2	в	і	д	а	т	в	о	р	ю	в	а	т	и				
2	х	а	р	а	к	т	е	р	и	с	т	и	к	и			
2	л	і	т	е	р	а	т	у	р	н	и	я					

Видалити обрану Вибразити n-грами з осередками від: 8 До: 15 одиниць

Редагувати обрану Побудувати n-грами

## Лексичні n-грами

Pidlapushnyi ErrorFinder 1.0

Модуль формування навчального масиву коректних слів Модуль побудови матриці лексичних n-грам Модуль пошуку та видалення граничних помилок

**Тестовий контент для перевірки**

що таке літературний текст  
як літературний текст називається кожен один текст в якому характеристика літературного жанру визнаються євості описового поезії театру або репетиції

з точки зору мовознавства літературний текст це той де переважає поетична функція мови тобто там де композиція тексту в першу чергу залежить від форми самої мови її краси та її здатності відтворювати уявні світи

Завантажити текст з ПК

Обрати текст

Обрати текст з БД

**Перевірити текст!**

**Результат перевірки**

слово0	слово1	слово2	слово3
що	таке	літературний	текст
євості	текст	в	якою
описового	поезії	театру	
мовознавства	літературний	текст	це
мови	тобто	жан	де
залежить	від	форми	самої
здатності	відтворювати	уявні	світи

Правильні варіанти для заміни

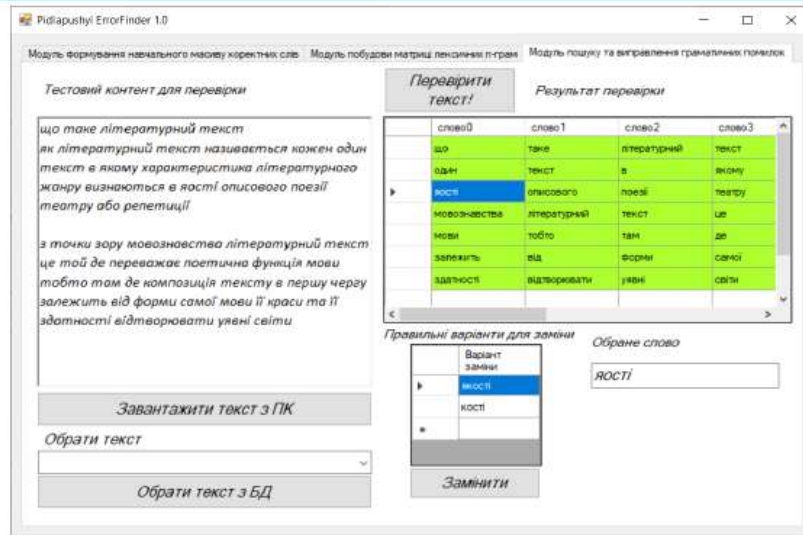
Варіант запису
явості
явості

Обране слово

Явості

Замінити

## Знайдена помилка



Виправлення помилки

## Висновки

Мету кваліфікаційної роботи бакалавра було успішно досягнуто, а саме – створення й прикладна програмна реалізація методу автоматизованого виправлення граматичних помилок у цифрових.

За досягнення мети були вирішені такі задачі:

1. Проведено аналіз предметної області, зокрема огляд теоретичних підходів до розв'язання задачі виправлення граматичних помилок у цифрових текстах, а також проведено аналіз існуючих програмних рішень.
2. Розробити метод автоматизованого виправлення граматичних помилок у цифрових текстах, на базі якого створити інформаційну структуру програмного застосунку.
3. На базі проведеного аналізу існуючих засобів створення програмного забезпечення обрано комбінацію засобів для розробки програмної реалізації методу автоматизованого виправлення граматичних помилок у цифрових текстах, а саме платформу .NET та мову програмування C#.
4. Обраними засобами розробки створити та протестувати прикладну програмну реалізацію методу автоматизованого виправлення граматичних помилок у цифрових текстах.

Ім'я користувача:  
Кафедра КН

ID перевірки:  
1011548879

Дата перевірки:  
12.06.2022 08:57:43 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
12.06.2022 08:59:59 EEST

ID користувача:  
100005671

Назва документа: Підлапушний\_ЗАПИСКА\_short

Кількість сторінок: 65 Кількість слів: 9639 Кількість символів: 72301 Розмір файлу: 3.93 MB ID файлу: 1011420917

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

## 5.18% Схожість

Найбільша схожість: 2.7% з джерелом з Бібліотеки (ID файлу: 1009540880)

1.6% Джерела з Інтернету

161

Сторінка 67

4.12% Джерела з Бібліотеки

128

Сторінка 68

## 0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

## 0% Вилучень

Немає вилучених джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

7

Підозріле форматування

20  
сторінок

# Anti-Plagiarism v-15.257

**Максимальное совпадение с одним документом 3.0%**

Словари проверки: en\_US, ru\_RU, ua\_UA. **Ошибок в документах: 10%**

ID: 105044 Название: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод автоматизованого виправлення граматичних помилок у цифрових текстах Добавлено в БД: 2022-06-12 Авторы: В.О. Підлапушний Руководители: О.В. Собко Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	55043	823	3052 (6%)	52 (6%)

## Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

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

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

Назва: Метод автоматизованого виправлення граматичних помилок у цифрових текстах

Автор: студент групи КН-18-1 Підлапушиний Владислав Олегович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: викладач кафедри КН Собко Олена Віталіївна

Після аналізу звіту подібності зроблено такий висновок:

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

*Підтвердження:*

*Запозичення, виявлені в роботі В.О. Підлапушиного, є законними і не є плагіатом, оскільки:*

*1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;*

*2) усі запозичення фрагментарні;*

*3) до запозичень входять фрагменти програмного коду, що не мають авторства і містять поширені конструкції;*

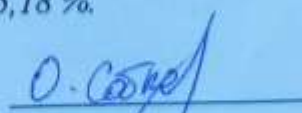
*4) серед запозичень знаходяться загальновідомі терміни, скорочення та визначення.*

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

*- за системою Anti-Plagiarism: 6%;*

*- за системою Unichек: 5,18 %.*

Керівник роботи



Олена СОБКО

Гарант ОП



Олександр МАЗУРЕЦЬ

Завідувач кафедри КН



Олександр БАРМАК



## РЕЦЕНЗІЯ

### на кваліфікаційну роботу бакалавра

студента р. КТ-18-1 Підлапушиний Владислав Олегович

за темою: Метод автоматизованого виправлення граматичних помилок у цифрових текстах

#### 1. Актуальність обраної теми

При створенні цифрових текстів автори можуть допускати не тільки граматичні помилки, а й стилістичні, лексичні, пунктуаційні. Допомагають знаходити та виправляти такі помилки редактори та коректори. Для проведення даної роботи існує ряд програмних застосунків, які дозволяють робити це автоматизовано. Кожен з них має свої переваги та недоліки, тому повністю покладатися на правильність роботи цих застосунків не варто. Тому є актуальною задачею комп'ютерних наук проведення дослідження щодо існуючих методів та засобів для автоматизованого виправлення граматичних помилок у цифрових текстах, а також розробка нового методу, що вирішує дану задачу.

#### 2. Повнота розкриття мети та завдань дослідження

Кваліфікаційна робота бакалавра студента Підлапушиного В.О. виконана в повному обсязі, було проаналізовано предметну область, створено метод автоматизованого виправлення граматичних помилок у цифрових текстах та реалізовано програмний застосунок на базу вищезазначеного методу.

#### 3. Зміст кожного розділу роботи

Перший розділ присвячений проведенню аналізу предметної області та визначенню основних параметрів для розв'язку поставленої задачі. Другий розділ присвячений проєктуванню функціональної структури інформаційної системи автоматизованого виправлення граматичних помилок у цифрових текстах. Третій розділ присвячений програмній реалізації спроектованої функціональної структури інформаційної системи. Також сформовано висновок по роботі.

#### 4. Оцінка розробленої інформаційної системи, її практична цінність

Напрямами практичного використання системи є: обробка текстів та визначення в них граматичних помилок, візуальне відображення місць в тексті, що містять помилки, підбір пропозицій щодо виправлень та відсоткове значення помилок в тексті.

#### 5. Якість оформлення кваліфікаційної роботи бакалавра

Робота виконана на належному науково-методичному рівні та відповідає встановленим вимогам щодо оформлення такого роду праць.

#### 6. Недоліки кваліфікаційної роботи бакалавра

В роботі варто було використовувати не тільки повну назву створеного методу, а й ввести скорочену назву, для зменшення обсягу контенту.

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

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

Рецензент

Мартишок В.В.



## ВІДГУК НАУКОВОГО КЕРІВНИКА на кваліфікаційну роботу бакалавра

студента *гр. КН-18-1 Підлапушиний Владислав Олегович*

за темою Метод автоматизованого виправлення граматичних помилок у цифрових текстах

### 1. Актуальність теми

Процес виявлення та виправлення помилок доволі складний, адже робота над текстом може бути різної глибини – від перевірки граматичних помилок, і до перевірки стилістичних та лексичних помилок. Це потребує від перевіряючого великого обсягу знань – як базових правил написання слів, так і поглиблених. Тому особливо актуальним є автоматизація процесу виявлення помилок у цифрових текстах для спрощення роботи людини. З іншої сторони можна стверджувати, що цей процес не можна цілком довірити машині, він повинен контролюватися людиною, адже машина не завжди може враховувати особливості правопису слів певної мови. Відповідно, автоматизація процесів виправлення граматичних помилок у цифрових текстах є актуальною задачею комп'ютерних наук.

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

Поставлена у кваліфікаційній роботі бакалавра мета стосується розробки методів і технологій отримання, зберігання, обробки, передачі та використання інформації, інтелектуального аналізу даних і прийняття рішень, а саме розробки методу виправлення граматичних помилок у цифрових текстах. При цьому при вирішенні поставлених задач використовуються математичні моделі, методи та алгоритми розв'язання теоретичних і прикладних задач, що виникають при розробці інформаційних технологій. Тому результати виконання кваліфікаційної роботи бакалавра відповідають стандарту бакалавра спеціальності 122 – Комп'ютерні науки.

### 3. Професійні та особистісні якості бакалавра

При роботі над кваліфікаційною роботою бакалавра Підлапушиний Владислав Олегович вчасно виконував поставлені задачі дослідження й проявив себе кваліфікованим фахівцем. Як в процесі написання пояснювальної записки, так і при розробці прикладного програмного забезпечення проявив цілком достатні для одержання успішного результату компетентності.

**4. Ступінь самостійності під час виконання кваліфікаційної роботи**

Одержані в роботі результати є наслідком особистої діяльності студента, який самостійно виконував всі поставлені задачі.

**5. Ступінь оволодіння методами дослідження**

В роботі при розробці та прикладній реалізації методу автоматизованого виправлення граматичних помилок у цифрових текстах виявлено достатній ступінь оволодіння студентом необхідними інструментами та обладнанням, методами, методиками та технологіями предметної області комп'ютерних наук.

**6. Повнота та якість розкриття теми роботи**

Тема роботи в повній мірі обґрунтована й розкрита, проведено аналіз актуальності та відомих досліджень в межах обраної теми, поставлені завдання, які у роботі виконані, та розроблено програмне забезпечення для автоматизованого виправлення граматичних помилок у цифрових текстах.

**7. Логічність, послідовність, аргументованість, літературна грамотність викладу матеріалу**

Структура роботи та послідовність викладення логічні та відповідні поставленій меті. Викладення матеріалу грамотне та виявляє високий ступінь відповідності стилю.

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

Запропонований метод автоматизованого виправлення граматичних помилок у цифрових текстах може мати практичне використання, зокрема програмно реалізовано систему автоматизованого виправлення граматичних помилок у цифрових текстах. Розроблена система призначена для користувачів, що мають необхідність швидкого виправлення граматичних помилок в текстах, реалізований програмний продукт може бути особливо корисним для осіб, що створюють тексти для поширення в соціальних мережах, статті, рецензії, тощо. Інструментами для практичного використання системи є: обробка текстів та визначення в них граматичних помилок, візуальне відображення місць в тексті, що містять помилки, підбір пропозицій щодо виправлень та відсоткове значення помилок в тексті.

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

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

Керівник



викладач каф. КН Олена СОБКО