Хмельницький національний університет

Факультет інформаційних технологій

Кафедра комп'ютерних наук

Версія шаблону © CHONG\_2023\_КП\_МСШІ\_AstraKoza\_7.02

**КУРСОВИЙ ПРОЄКТ**

з дисципліни «Методи та системи штучного інтелекту»

на тему *Метод застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм*

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 групи КНс-22-1 Вадим МАЛАЙДАХ*

Курс, група виконавця Підпис  Ініціали, прізвище

Керівник: *к.т.н., доц. каф. КН Олександр МАЗУРЕЦЬ*

Науковий ступінь, посада Підпис  Ініціали, прізвище

Нормоконтроль: *к.т.н., доц. каф. КН Олександр МАЗУРЕЦЬ*

Науковий ступінь, посада Підпис  Ініціали, прізвище

До захисту допускаю:

*Зав. кафедри КН, д.т.н., професор*  *Олександр БАРМАК*

Підпис Ініціали, прізвище \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 р.

Хмельницький 2023

**Анотація**

Тема курсового проєкту: *«Метод застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм»*

Виконавець курсового проєкту: *студент групи КНс-22-1 Вадим Малайдах*

Керівник курсового проєкту: *к.т.н., доцент кафедри КН Олександр*

*Мазурець*

Курсовий проєкт містить:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Пояснювальна записка | |  | Кількість додатків |
| Сторінок | Рисунків | Таблиць | Джерел інформації |
| 40 | 16 | 0 | 24 | 3 |

Метою цього курсового проекту із дисципліни «Методи та системи штучного інтелекту» є розробка методу, який використовує методи інтелектуального аналізу для вдосконалення процесів введення тексту за допомогою біграм, та відповідної програмної системи.

Проект спрямований на дослідження потенціалу використання шаблонів біграм, щоб допомогти користувачам більш ефективно складати текст. Мета полягає в тому, щоб розкрити розуміння того, як біграми можна використовувати для прогнозування та пропозиції слів, таким чином забезпечуючи більш інтуїтивно зрозумілий і розумний досвід створення тексту для користувачів. Для розробки програмної системи було використано мову програмування С# в середовищі Visual Studio Community 2022.

Ключові слова: біграм, інтелектуальний аналіз, введення тексту, обробка природної мови.

Виконавець: *студент групи КНс-22-1 Вадим МАЛАЙДАХ*

Курс, група виконавця Підпис Ініціали, прізвище

**Зміст**

Додатки

# Перелік скорочень

|  |  |
| --- | --- |
| **Скорочення, термін,** |  |
| **позначення** | **Пояснення** |
| IT | Інформаційні технології |
| ПК | Персональний комп’ютер |
| КН | Комп’ютерні науки |
| PDF | Portable Document Format |
| DOCX | Document and XML(Extensible Markup Language) |
| IDE | Integrated development environment |
| XML | Extensible Markup Language |
| HTML | Hypertext Markup Language |
| MS | Microsoft |
| VS | Visual Studio |
| NLP | Natural Language Processing |
| Т9 | Text on 9 keys |
| HMS | Hasselgren, Montnémery, Svensson - імена авторів |

# Вступ

За останні роки сфера інформатики стала свідком значного прогресу, що призвело до значних трансформацій у різних областях. Однією з таких областей, яка значно виграла від цих технологічних досягнень, є інтелектуальний аналіз, застосований до введення тексту [16] [17]. Здатність ефективно аналізувати та інтерпретувати текстові дані стала вирішальною в багатьох програмах, починаючи від обробки природної мови до систем пошуку інформації. Цей курсовий проект має на меті вивчити новий метод застосування інтелектуального аналізу для сприяння введенню тексту за допомогою біграм.

Введення тексту [18] відіграє фундаментальну роль у взаємодії людини з комп’ютером, дозволяючи користувачам передавати свої думки, запити та команди обчислювальним системам. Процес введення тексту традиційно передбачає введення символів, слів або фраз вручну за допомогою клавіатури чи інших пристроїв введення. Однак цей звичайний підхід часто займає багато часу та може викликати помилки. Інтеграція методів інтелектуального аналізу в системи введення тексту має потенціал для підвищення ефективності, точності та взаємодії з користувачем.

Запропонований метод у цьому курсовому проекті зосереджений на використанні біграм [19] [20], які є парами суміжних слів, як основи для інтелектуального аналізу. Біграми фіксують контекстну інформацію та залежності, наявні в природній мові, забезпечуючи глибше розуміння введення тексту. Використовуючи біграми, можна розробити інтелектуальні алгоритми, які можуть передбачати та пропонувати відповідні слова на основі вхідного контексту, що призводить до покращених можливостей завершення тексту, автоматичного виправлення та передбачення.

Автоматизація процесів введення тексту за допомогою застосування інформаційних технологій приносить кілька помітних переваг. По-перше, це підвищує продуктивність за рахунок скорочення часу та зусиль, необхідних для введення тексту вручну, дозволяючи користувачам зосередитися на створенні значущого вмісту, а не боротися з громіздкими завданнями набору тексту. Подруге, це покращує точність введення тексту шляхом мінімізації орфографічних і граматичних помилок завдяки пропозиціям у реальному часі та автоматичним виправленням. Крім того, автоматизація процесів введення тексту може підвищити доступність для людей з фізичними або когнітивними обмеженнями, надаючи їм альтернативні засоби спілкування та взаємодії.

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

## 1.1 Аналіз інформаційних моделей

Робота з текстами [9] на ПК може включати в себе створення, редагування, форматування та зберігання текстових документів. Одним з найпопулярніших програмних засобів для роботи з текстами є Microsoft Word. Це програмне забезпечення дозволяє покращувати якість текстів, створювати ефектно оформлені документи та співпрацювати з іншими будь-де й будь-коли. За допомогою Редактора Microsoft можна перевіряти орфографію, граматику, написання з великої букви й розділові знаки в усіх документах Word. Також можна ділитися документами з колегами й редагувати їх разом, а також додавати коментарі та приймати запропоновані зміни в реальному часі. Документи можна зберегти у OneDrive і отримувати до них доступ із будь-якого пристрою.

Цифровий текст [4] [5] то текст, який готовий до читання на різних цифрових платформах, таких як комп’ютери, мобільні телефони, планшети тощо. Цифрові тексти можуть бути створені в різних форматах, таких як PDF, DOCX, HTML та інших. Цифрові тексти можуть бути збережені на комп’ютері або в хмарному сховищі та доступні для читання та редагування з будь-якого пристрою з підключенням до Інтернету. Цифрові тексти можуть бути використані для роботи з документами, електронними книгами, блогами, вебсторінками та іншими видами контенту.

Введення тексту на ПК може здійснюватися за допомогою текстових редакторів, таких як Microsoft Word, LibreOffice Writer, Google Docs [12, 13, 14] тощо. Ці програми дозволяють створювати, редагувати та форматувати текстові документи. Вони також мають функції для перевірки орфографії та граматики, а також для вставки таблиць, зображень та інших об’єктів.

Таблиця символів є інструментом, який дозволяє вставляти спеціальні символи у текстовий документ. Це може бути корисно, коли потрібно вставити символ, який не доступний на клавіатурі. Таблиця символів містить різні символи, включаючи математичні символи, символи валют, літери з діакритичними знаками тощо.

Файли шрифтів [7] [8] то файли, які містять інформацію про шрифти, що використовуються у текстових документах. Це дозволяє забезпечити правильне відображення тексту на різних пристроях. Файли шрифтів можуть бути встановлені на комп’ютер для розширення набору доступних шрифтів.

Параметри тексту [7] [8] включають різні налаштування, такі як кегль, тип шрифту, стилі та формати. Наприклад, у програмі Microsoft Word можна налаштувати або створити нові стилі для швидкого форматування всього документа. Можна змінити форматування (наприклад, розмір і колір шрифту та відступ тексту) у стилях, застосованих до назв, заголовків, абзаців, списків тощо.

Допомога при введенні тексту [14] [15] може включати різні інструменти та функції, які полегшують введення тексту. Один з таких інструментів є функція автоматичне заповнення, яка пропонує користувачеві слова та фрази під час набору тексту. Це може бути корисним для швидкого введення часто використовуваних слів та фраз.

Іншим корисним інструментом є програми для перевірки граматики та орфографії, такі як Grammarly або Microsoft Editor. Ці програми допомагають знаходити та виправляти помилки у тексті, а також покращують стиль та тон написання.

Крім того, багато текстових редакторів мають вбудовані функції для полегшення введення тексту, такі як шаблони, автозаміна та гарячі клавіші. Використання цих функцій може допомогти швидко форматувати текст та вводити часто використовувані елементи.

Словники правопису - це словники, які містять правила написання слів у певній мові. Наприклад, український правописний словник містить правила написання слів українською мовою.

Щодо технологій, Т9 [3] то предиктивна система набору текстів для мобільних телефонів. Назва Т9 виникла від англ. Text on 9 keys, тобто набір тексту на 9 кнопках. Т9 розроблена компанією Tegic Communications і використовується в мобільних телефонах багатьох великих виробників. При наборі тексту система Т9 намагається передбачити, яке слово ви намагаєтесь набрати, використовуючи словник. Найуживаніші слова підставляються першими.

Google Keyboard [1] [2], також відома як Gboard, то клавіатура для мобільних пристроїв, розроблена компанією Google. Вона має все, що ви любите у клавіатурі Google – швидкість та надійність, Glide Typing, голосовий ввід, рукописний ввід та багато іншого.

Допомога при введенні тексту може мати значний ефект на зручність та продуктивність користувача. Наприклад, функції автозаповнення та автозаміни можуть допомогти користувачам швидше вводити текст, зменшуючи кількість помилок та зберігаючи час. Тому засоби та методи допомоги при введенні тексту на сучасному етапі є актуальним напрямком ІТ.

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

Інтелектуальний аналіз тексту [16] [17] є напрямом інтелектуального аналізу даних та штучного інтелекту, метою якого є отримання інформації з колекцій текстових документів, ґрунтуючись на застосуванні ефективних, у практичному плані, методів машинного навчання та обробки природної мови. Інтелектуальний аналіз тексту використовує всі ті ж підходи до перероблювання інформації, що й інтелектуальний аналіз даних, однак різниця між цими напрямками проявляється лише в кінцевих методах, а також у тому, що інтелектуальний аналіз даних має справу зі сховищами та базами даних, а не електронними бібліотеками та корпусами текстів.

Введення тексту [18] є фундаментальним аспектом взаємодії людини та комп’ютера, який дозволяє користувачам передавати свої думки, повідомлення, команди чи запити обчислювальним системам за допомогою письмової мови. Традиційні методи введення тексту включають ручне введення символів, слів або фраз за допомогою фізичної або віртуальної клавіатури. Користувачі вводять потрібний текст, натискаючи клавіші, що відповідають певним символам, а введення потім передається та обробляється системою.

Біграмна мовна модель [19] [20] то тип статистичної мовної моделі, яка прогнозує ймовірність появи слова в послідовності на основі попереднього слова. Вона розглядає пари послідовних слів (біграм) і оцінює ймовірність появи певного слова, враховуючи попереднє слово в тексті або реченні.

Інтелектуальні алгоритми [21] є практичними альтернативними методами для вирішення різноманітних складних інженерних проблем. Наприклад, методи нечіткого керування можуть бути використані для побудови нелінійних регуляторів за допомогою евристичної інформації, коли інформація про фізичну систему є обмеженою. Ці алгоритми можна використовувати в різних галузях, таких як штучний інтелект, машинне навчання та оптимізація.

Обробка природної мови (Natural Language Processing, NLP) [20] [22] – це

міждисциплінарна підгалузь лінгвістики, комп'ютерних наук та штучного інтелекту, що займається взаємодією між комп'ютером і людською мовою. Вона передбачає надання комп'ютерам здатності розуміти текст і розмовні слова так само, як це роблять люди. NLP поєднує обчислювальну лінгвістику - моделювання людської мови на основі правил – зі статистичними моделями, моделями машинного навчання та моделями глибокого навчання. Це дозволяє комп'ютерам обробляти людську мову у вигляді тексту або голосових даних і "розуміти" її повне значення, разом з намірами та почуттями того, хто говорить або пише.

Одним із способів застосування інтелектуального аналізу для допомоги у введенні тексту за допомогою біграм є використання алгоритмів прогнозування тексту. Ці алгоритми можуть аналізувати текст, який користувач ввів до цього часу, і використовувати біграмні мовні моделі для прогнозування найбільш ймовірного наступного слова або фрази. Це може допомогти прискорити введення тексту, пропонуючи слова або фрази, які користувач, ймовірно, введе наступними. Крім того, методи машинного навчання можна використовувати для підвищення точності цих прогнозів з часом, навчаючись на основі шаблонів друку користувача.

Існує кілька програм для застосування інтелектуального аналізу для допомоги у введенні тексту з використанням біграм. Однією з таких програм є HMS [23], яка передбачає слова з послідовності натискань клавіш,

використовуючи словник і функцію, що поєднує частоти біграм і довжину слова ¹. Інший підхід полягає у використанні марковських моделей для обчислення ймовірностей послідовних подій для передбачення тексту.

HMS то метод предиктивного введення тексту, який використовує біграми для допомоги у введенні тексту. Він передбачає слова з послідовності натискань клавіш, використовуючи словник і функцію, що поєднує частоту біграм і довжину слова. HMS було реалізовано на програмній імітації клавіатури мобільного телефону і порівняно з широко доступною комерційною системою. Мовна модель була навчена на корпусі шведських новин і оцінена. Результати показали зменшення на 7-13 відсотків кількості натискань клавіш, необхідних для введення тексту. Метод введення тексту HMS можна легко перенести на інші мови.

Марковські моделі [24] можуть бути потужним інструментом для

обчислення ймовірностей послідовних подій для прогнозування тексту. Однак, як і будь-яка статистична модель, вони мають свої обмеження. Одна з потенційних проблем полягає в тому, що марковські моделі не мають пам'яті, тобто вони роблять прогнози на основі лише поточного стану системи і не беруть до уваги минулі стани. Це може обмежити їхню здатність вловлювати довгострокові залежності в даних. Крім того, якість прогнозів, зроблених марковською моделлю, залежить від якості та кількості навчальних даних, які використовуються для побудови моделі.

Отже, засоби та методи інтелектуального аналізу текстів здатні допомогти в вирішенні задачі допомоги при введенні цифрових текстів. Зокрема, визначено перспективним використання марковських моделей, зокрема біграм.

## 1.3 Аналіз існуючих програмних рішень

Одним з прикладів програми для введення тексту є T9 [3] і Google Keyboard [1, 2], які є предиктивними системами набору текстів для мобільних телефонів. Назва T9 виникла від англ. «Text on 9 keys», тобто набір тексту на 9 кнопках. T9 розроблена компанією Tegic Communications і використовується в мобільних телефонах багатьох великих виробників. При наборі тексту система T9 намагається передбачити, яке слово ви намагаєтесь набрати, використовуючи словник. Найуживаніші слова підставляються першими.

Вона дозволяє формувати слова за допомогою одного натискання клавіші для кожної літери, що є значним поліпшенням порівняно з підходом мультитап, який використовується у звичайному введенні тексту на мобільних телефонах, де кілька літер пов’язано з кожною клавішею, а вибір однієї літери часто вимагає кількох натискань клавіш (рисунок 1.1).

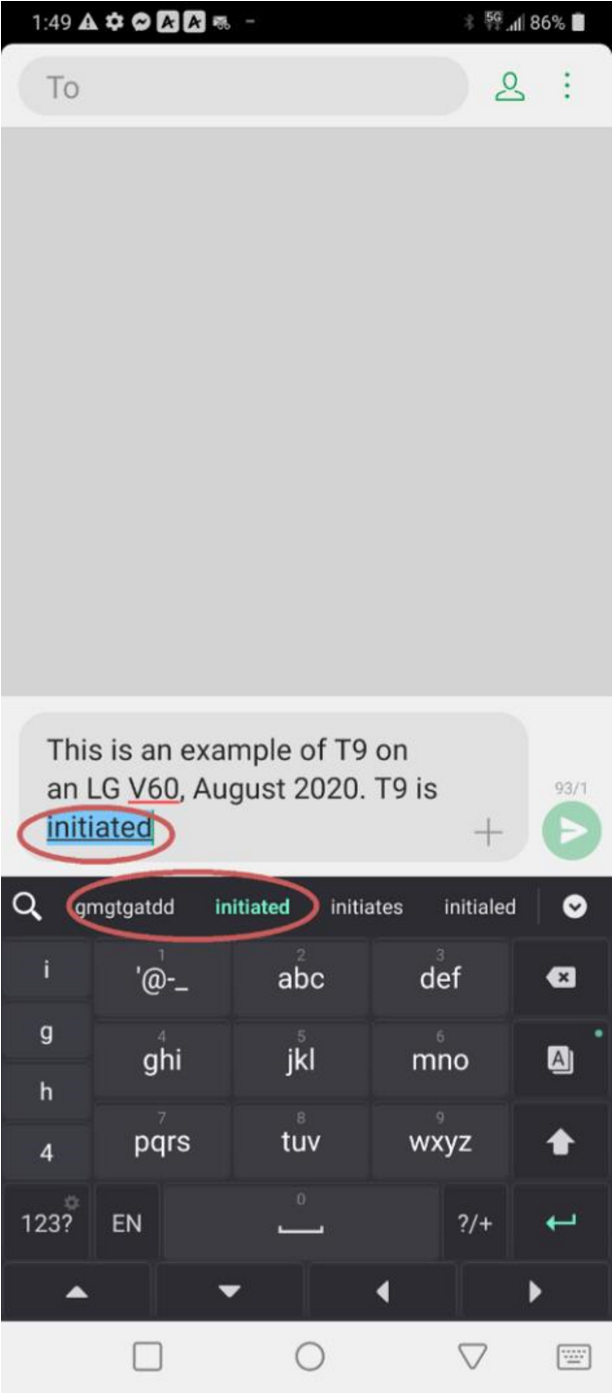


Рисунок 1.1 – Інтерфейс Т9 [131]

Однак у T9 також є деякі недоліки. Наприклад, система може не завжди правильно передбачити слово, яке ви намагаєтесь набрати, особливо якщо це рідко вживане слово або ім’я. У таких випадках користувачеві доводиться вручну вибирати правильне слово зі списку пропозицій або вводити його за допомогою мультитапу.

Іншим прикладом є Google Keyboard, також відома як Gboard, - це клавіатура для мобільних пристроїв, розроблена компанією Google. Вона має все, що ви любите у клавіатурі Google - швидкість та надійність, Glide Typing,

голосовий ввід, рукописний ввід та багато іншого (рисунок 1.2).

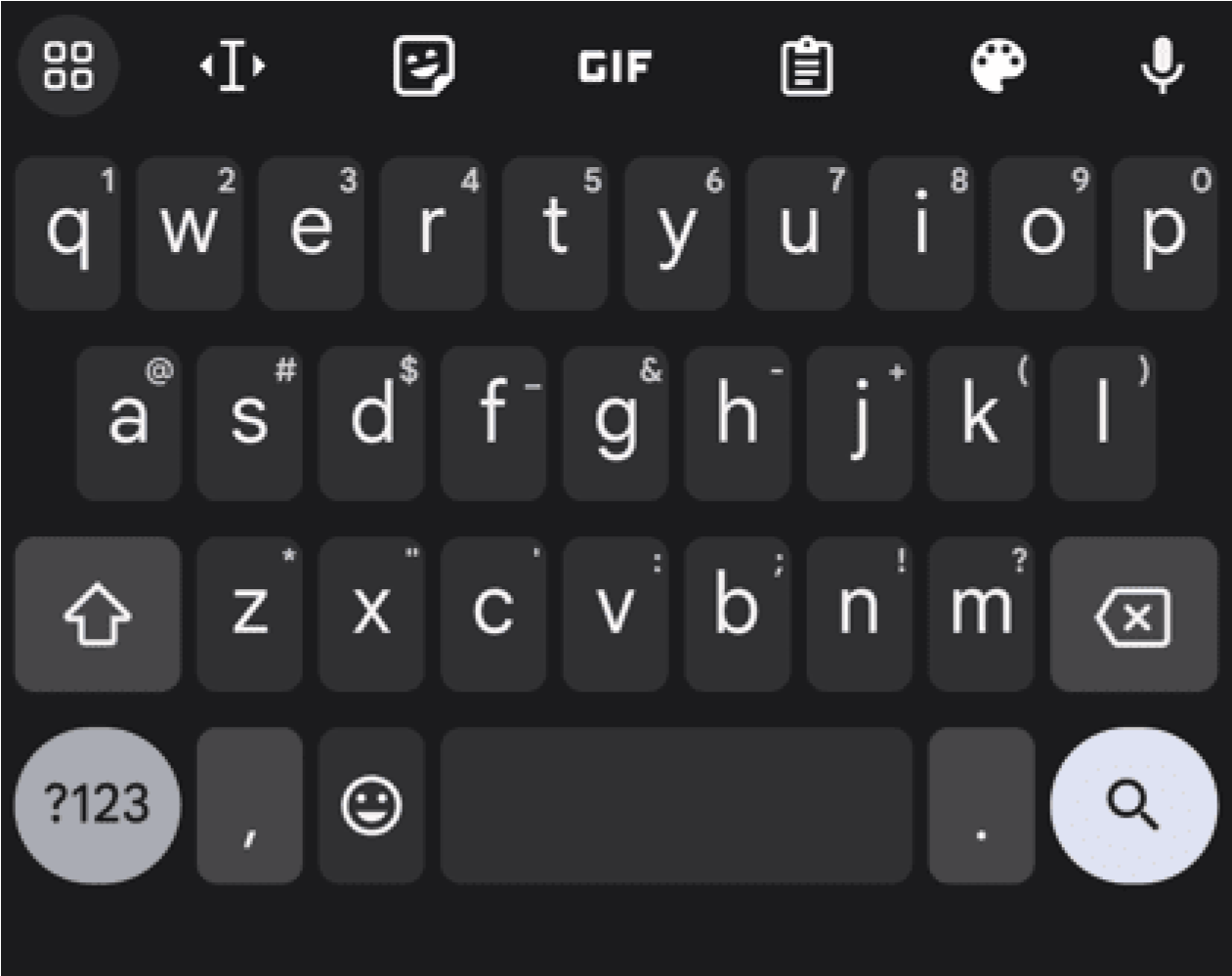


Рисунок 1.2 – Google Keyboard

Gboard пропонує ряд функцій, таких як Glide Typing, який дозволяє швидше набирати текст, перетягуючи палець від літери до літери. Голосовий ввід дозволяє легко диктувати текст замість набору на клавіатурі. Рукописний ввід дозволяє писати курсивом та друкованими літерами. Пошук емодзі допомагає знайти потрібне емодзі швидше. GIF-файли - пошук та обмін GIFфайлами для ідеальної реакції. Багатомовний ввід - не потребує перемикання мов ручками. Gboard буде автокоректувати та пропонувати з будь-якої з увімкнених мов. Google Translate - перекладайте поки друкуєте на клавіатурі.

Ось деякі переваги Gboard:

* Glide Typing: дозволяє швидше набирати текст, перетягуючи палець від літери до літери.
* Голосовий ввід: дозволяє легко диктувати текст замість набору на клавіатурі.
* Рукописний ввід: дозволяє писати курсивом та друкованими літерами.
* Пошук емодзі: допомагає знайти потрібне емодзі швидше.
* GIF-файли: пошук та обмін GIF-файлами для ідеальної реакції.
* Багатомовний ввід: не потребує перемикання мов ручками. Gboard буде автокоректувати та пропонувати з будь-якої з увімкнених мов.
* Google Translate: перекладайте поки друкуєте на клавіатурі1.

Однак у Gboard також є деякі недоліки. Наприклад, у порівнянні з іншими клавіатурами, такими як SwiftKey або Swype, Gboard може мати менше функцій та опцій налаштування. Також можуть бути проблеми з автокорекцією та передбаченням слів.

Отже, аналіз існуючих програмних рішень в напрямку застосування засобів та методів інтелектуального аналізу текстів для допомоги при введенні цифрових текстів показав, що на сучасному етапі є ряд ефективних реалізацій, проте в умовах актуальності даного питання задача допомоги при введенні цифрових текстів залишається актуальною, зокрема визначено перспективним використання біграм

## 1.4 Мета, задачі та вимоги до реалізації інформаційної системи

Метою буде підвищення ефективності та точності введення тексту шляхом прогнозування найбільш ймовірного наступного слова або фрази на основі попереднього слова. Цілі можуть включати зменшення кількості натискань клавіш, необхідних для введення тексту, підвищення швидкості введення тексту і підвищення задоволеності користувачів процесом введення тексту. Вимоги до такої системи включатимуть великий масив текстових даних для навчання алгоритмів прогнозування, а також метод поєднання частот біграм і довжини слів для точного прогнозування. Крім того, система повинна мати можливість адаптуватися до індивідуальних особливостей набору тексту користувачами, щоб з часом покращувати свої прогнози.

Метою курсового проєкту є розробка методу застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм та відповідної інформаційної системи передбачення та пропозицій введення слів на платформі .NET, що виконує наступні основні функції:

* введення навчального тексту для автоматизованого визначення закономірностей чергування слів;
* очищення навчального тексту від спеціальних символів;
* переведення тексту до нижнього регістру;
* формування послідовного масиву слів навчального тексту;
* формування масиву біграм (пар оригінальних слів) навчального тексту;
* обрахунок кількості появ біграм у навчальному тексті й формування статистичної біграмної таблиці;
* введення користувачем робочого тексту;
* пропозиція введення варіантів наступного слова за допомогою статистичної біграмної таблиці;
* вибір користувачем наступного слова із числа запропонованих варіантів або введення нового слова;
* збереження нового користувацького тексту.

# Розділ 2 Проєктування інформаційної системи передбачення та пропозицій введення слів

## 2.1 Метод застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм

Метод застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм призначений для перетворення вхідних даних у вигляді навчального тексту для формування статистичної біграмної таблиці в вихідні дані у вигляді сформованого користувацького тексту з використанням пропозицій введення варіантів наступного слова за допомогою статистичної біграмної таблиці.

Вхідними даними методу застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм є навчальний текст для формування статистичної біграмної таблиці та введений робочий текст користувачем.

Далі є два етапи а між ними проміжні дані. У першому етапі формується статистича біграмна таблиця діляться на такі процеси:

1. Введення навчального тексту для автоматизованого визначення закономірностей чергування слів;
2. Очищення навчального тексту від спеціальних символів;
3. Переведення тексту до нижнього регістру;
4. Формування послідовного масиву слів навчального тексту;
5. Формування масиву біграм (пар оригінальних слів) навчального тексту;
6. Обрахунок кількості появ біграм у навчальному тексті й формування статистичної біграмної таблиці.

Далі проміжні дані якому є статистична біграмна таблиця. Наступне це другий етап а саме допомога при введенні тексту з використанням біграм в ньому є:

1. Введення користувачем робочого тексту;
2. Пропозиція введення варіантів наступного слова за допомогою статистичної біграмної таблиці;
3. Вибір користувачем наступного слова із числа запропонованих варіантів або введення нового слова;
4. Збереження нового користувацького тексту.

А на останнє вихідні дані, а саме сформований користувацький текст. Схема методу застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм наведена у рисунку 2.1.

Рисунок

2

.

1

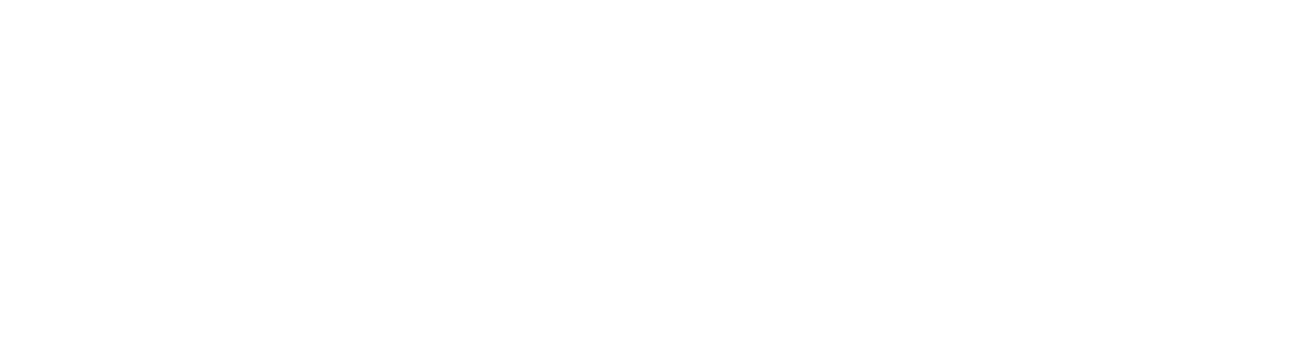
–

Схема м

етод

у

застосування інтелектуального аналізу для



**Етап 1**

**–**

**Ф**

**ормування статистичної біграмної таблиці**

**:**

1)

Введення навчального тексту для автоматизованого визначення закономірностей

чергування слів;

2)

Очищення навчального тексту від

спеціальних символів;

3)

Переведення тексту до нижнього регістру;

4)

Формування послідовного масиву слів навчального тексту;

5)

Формування масиву біграм (пар оригінальних слів) навчального тексту;

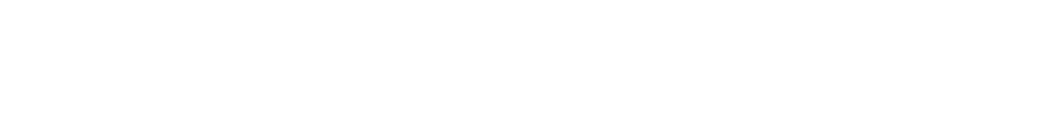
6)

Обрахунок кількості появ біграм у навчальному тексті й формування ст

атистичної

біграмної таблиці

.



–

Навчальний текст для

формування статистичної біграмної

таблиці

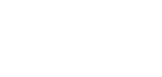
;

–

Процес в

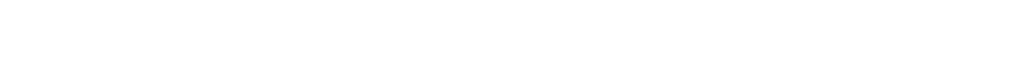
ведення користувачем робочого тексту

.



**Вхідні**

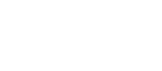
**дані**



Сформований

користувацьк

ий текст.

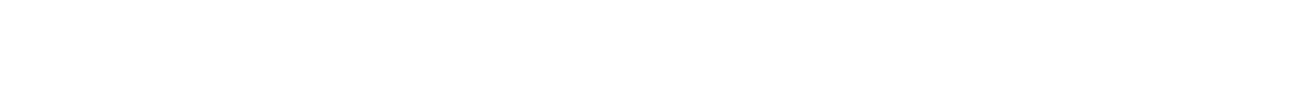


**В**

**и**

**хідні**

**дані**



С

татистичн

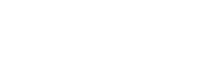
а

біграмн

а

таблиц

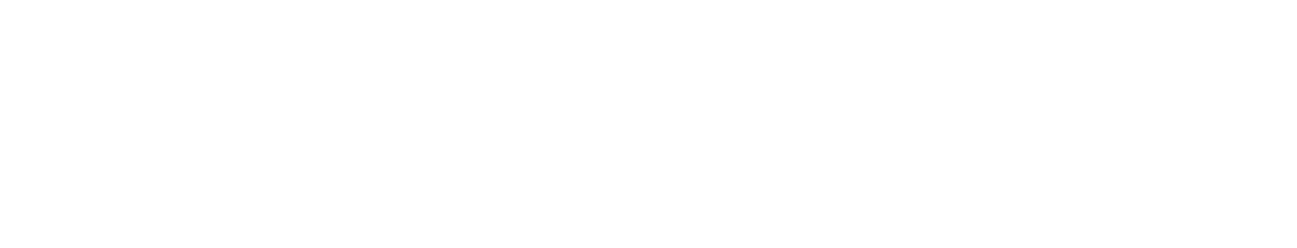
я.



**Проміж**

**ні**

**дані**



**Етап 2**

**–**

**Д**

**опомог**

**а**

**при введенні тексту з використанням біграм**

**:**

1)

Введення користувачем робочого тексту;

2)

Пропозиція введення варіантів наступного слова за допомогою статистичної біграмної

таблиці;

3)

Вибір користувачем наступного слова із числа

запропонованих варіантів або введення

нового слова;

4)

Збереження нового користувацького тексту.

допомоги при введенні тексту з використанням біграм

Також наведено рисунок діаграми активності методу застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм (рис. 2.2).



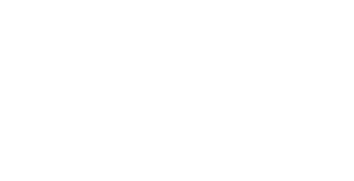
Введення навчального тексту



Початок діяльності



Формування статистичної біграмної таблиці

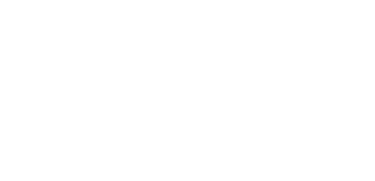


Вивід у дисплей

статистичну

біграмну

таблицю



Допомога при введенні

тексту з використанням

біграм



Зберегти користувацький текст



Кінець

діяльності

Рисунок 2.2 – Діаграма активності методу застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм

Аналогічно наведено діаграму послідовності у рисунку 2.3.

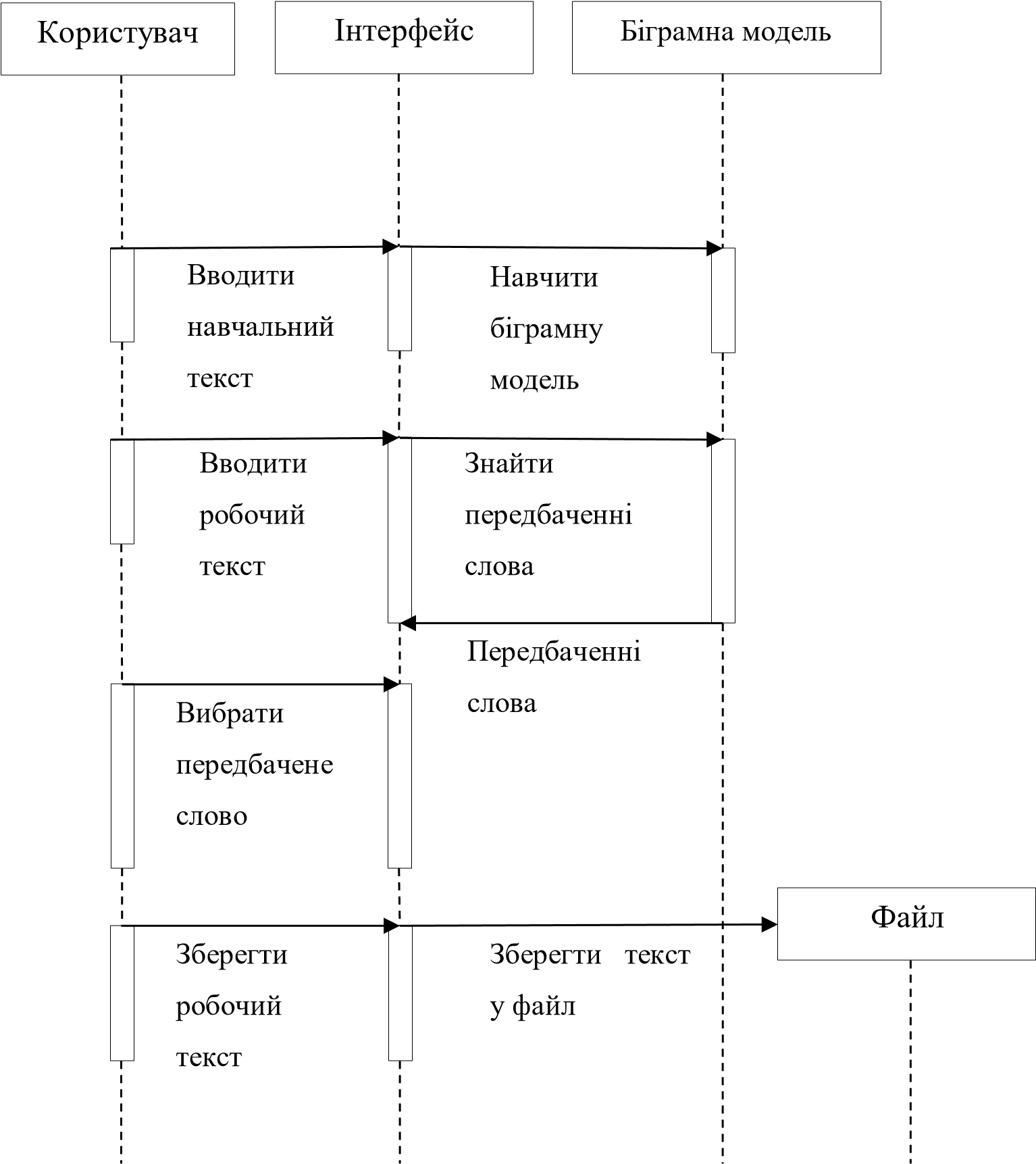


Рисунок 2.3 – Діаграма послідовності методу застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм

Таким чином, було розроблено метод застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм, який призначений для перетворення вхідних даних у вигляді навчального тексту для формування статистичної біграмної таблиці в вихідні дані у вигляді сформованого користувацького тексту з використанням пропозицій введення варіантів наступного слова за допомогою статистичної біграмної таблиці.

## 2.2 Інформаційна структура системи передбачення та пропозицій введення слів

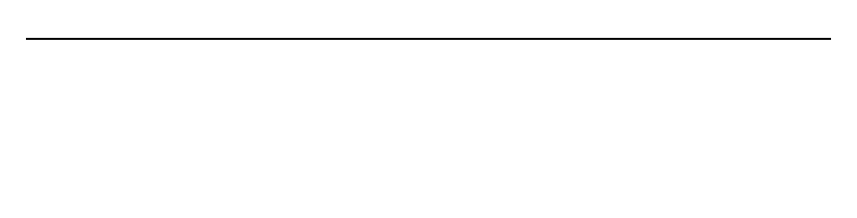
Інформаційна структура системи передбачення та пропозицій введення слів використовує метод застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм для перетворення вхідних даних у вигляді навчального тексту для формування статистичної біграмної таблиці в вихідні дані у вигляді сформованого користувацького тексту з використанням пропозицій введення варіантів наступного слова.

Система передбачення та пропозицій введення слів складається з двох модулів – підсистеми формування статистичної біграмної таблиці та підсистеми введенні тексту з використанням біграм.

Підсистема формування статистичної біграмної таблиці виконує функції, а саме очищення навчального тексту від спеціальних символів, переведення тексту до нижнього регістру, формування послідовного масиву слів навчального тексту, формування масиву біграм (пар оригінальних слів) навчального тексту, обрахунок кількості появ біграм у навчальному тексті й формування статистичної біграмної таблиці.

Підсистема введенні тексту з використанням біграм виконує функції, а саме введення користувачем робочого тексту, пропозиція введення варіантів наступного слова за допомогою статистичної біграмної таблиці, вибір користувачем наступного слова із числа запропонованих варіантів або введення нового слова.

Відповідно до методу застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм (див. п.2.1), було спроєктовано відповідну структуру інформаційної системи, зображену на Рисунку 2.2. Інформаційна система включає в себе дві підсистеми перша формує статистичну біграмну таблицю з введеного навчального тексту, а друга приймає таблицю для передбачення слова з введеного робочого тексту користувачем також отримати сформований текст.



*Підсистема введенні тексту з використанням біграм*

1)

Введення користувачем робочого тексту;

2)

Пропозиція введення варіантів наступного слова за

допомогою статистичної біграмної таблиці;

3)

Вибір користувачем наступного слова із числа

запропонованих варі

антів або введення нового слова.



*Підсистема формування статистичної біграмної таблиці*

1)

Очищення нав

чального тексту від спеціальних

символів;

2)

Переведення тексту до нижнього регістру;

3)

Формування послідовного масиву слів навчального

тексту;

4)

Формування масиву біграм (пар оригінальних слів)

навчального тексту;

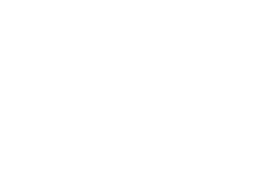
5)

Обрахунок кількості появ біграм у навчальному те

ксті й

формування статистичної біграмної таблиці

.



В

ведення

користувачем

робочого

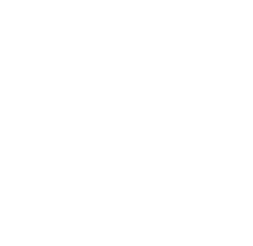
тексту



Сформований

користувацьк

ий текст



Навчальний

текст для

формування

статистичної

біграмної

таблиці



С

татистичн

а

біграмн

а

таблиц

я

Рисунок 2.2 – Схема інформаційної системи інтелектуального аналізу для допомоги при введенні тексту з використанням біграм

Таким чином, було сформовано інформаційну структуру системи передбачення та пропозицій введення слів, яка використовує метод застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм для перетворення вхідних даних у вигляді навчального тексту для формування статистичної біграмної таблиці в вихідні дані у вигляді сформованого користувацького тексту з використанням пропозицій введення варіантів наступного слова.

## 2.3 Вибір засобів розробки інформаційної системи

### 2.3.1 Вибір типу застосунку

Було вибрано віконний застосунок тому що Windows є однією з найпопулярніших операційних систем у світі, тому розробка додатка для цієї платформи може забезпечити широке охоплення аудиторії. Крім того, Microsoft надає розробникам ряд інструментів та ресурсів для спрощення процесу розробки, таких як Visual Studio та .NET Framework.

Також варто враховувати, що віконні додатки зазвичай мають більше можливостей для налаштування інтерфейсу користувача та інтеграції з операційною системою, ніж веб-додатки. Це може дозволити створити більш потужний та зручний для користувача продукт.

### 2.3.2 Вибір платформи

Вибрано .NET платформу, оскільки .NET також має хорошу сумісність з різними платформами, що дозволяє розробникам використовувати свої навички та код на всіх з них у знайомому середовищі. Це означає, що розробники можуть створювати застосунки швидше та з меншими витратами.

Крім того, .NET має велику спільноту розробників та багатий екосистему, що дозволяє легко інтегрувати бібліотеки з менеджера пакетів NuGet та ринку Visual Studio. Це дозволяє швидко знаходити відповіді на технічні проблеми та отримувати підтримку від спільноти, MVP та нашої великої організації підтримки.

### 2.3.3 Вибір середовища програмування

Обрано середовище програмування MS Visual Studio, що дає можливість робити віконний застосунок і програмувати у платформі .NET. MS Visual Studio є потужним інструментом розробки, який дозволяє виконувати весь цикл розробки в одному місці. Це комплексне інтегроване середовище розробки (IDE), яке можна використовувати для написання, редагування, налагодження та збирання коду, а потім розгортання вашого додатка.

MS Visual Studio має багато функцій та можливостей для кодування, що дозволяє розробникам ефективно та спільно розробляти код високої якості. Встановлювач на основі навантаження дозволяє встановлювати лише те, що вам потрібно, а потужні інструменти та функції кодування забезпечують все необхідне для створення вашого додатка в одному місці.

MS Visual Studio також підтримує кросплатформену розробку, що дозволяє створювати додатки для будь-якої платформи. Інтеграція з системою контролю версій дозволяє співпрацювати з кодом з членами команди.

### 2.3.4 Вибір фреймворка

Вибір впав на .NET Framework. Однією з переваг .NET Framework є його продуктивність. Фреймворк пропонує передові мовні функції, такі як узагальнення, LINQ та асинхронне програмування, а також широкий спектр класових бібліотек та підтримку багатьох мов для розробників. Це дозволяє розробникам ефективно створювати код високої якості.

.NET Framework також має велику спільноту розробників та багатий екосистему, що дозволяє легко інтегрувати бібліотеки з менеджера пакетів NuGet та ринку Visual Studio. Це дозволяє швидко знаходити відповіді на технічні проблеми та отримувати підтримку від спільноти, MVP та нашої великої організації підтримки.

### 2.3.5 Вибір мови програмування

Було зроблено вибір мови програмування C# для розробки застосунку, що може бути пояснено кількома причинами. C# є потужною мовою програмування, яка легко навчається та використовується для розробки широкого спектра програмного забезпечення, включаючи веб-застосунки, додатки для Windows та відеоігри для консолей та ПК.

Однією з переваг C# є його продуктивність. Це статично типізована мова, яка легко читається та зрозуміла, що полегшує пошук помилок у коді (вашому або іншого розробника) та розуміння коду, який ви читаєте.

C# також є кросплатформеною мовою. Це означає, що якщо ви створите застосунок на C#, він може працювати на будь-якій операційній системі або платформі, включаючи Apple, iOS, Windows, Android або у хмарі.

Крім того, C# має велику спільноту розробників та багатий екосистему. Це дозволяє швидко знаходити відповіді на технічні проблеми та отримувати підтримку від спільноти.

### 2.3.6 Вибір редактора програмного коду

Обрано Visual Studio Community 2022 тому що Visual Studio Community

2022 безкоштовна версія Visual Studio, повноцінного інтегрованого середовища розробки (IDE) для Windows та Mac. Це потужний інструмент для написання, редагування, налагодження та запуску коду, а також розгортання вашого додатка. Visual Studio Community 2022 має багато функцій та можливостей для кодування, а також підтримку кросплатформеної розробки.

Таким чином, вибрано: тип віконного застосунку, платформу .NET, середовище програмування MS Visual Studio, фреймворк .NET Framework, мову програмування C#, а також редактор програмного коду Visual Studio Community 2022.

# Розділ 3 Програмна реалізація інформаційної системи передбачення та пропозицій введення слів

## 3.1 Структура та функціональне призначення програмних складових системи

Для програмної реалізації методу застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм було створено відповідну структуру інформаційної системи для передбачення та пропозицій уведення слів.

Програмна складова для передбачення та пропозицій уведення слів складається із двох модулів – модуля формування статистичної біграмної таблиці та модуля уведенні тексту з використанням біграм.

Програмна складова формування статистичної біграмної таблиці призначена для виконання ряду функцій, а саме: очищення навчального тексту від спеціальних символів, функції переведення тексту до нижнього регістру, функції формування послідовного масиву слів навчального тексту, формування масиву біграм (пар оригінальних слів) навчального тексту, функції обрахунку кількості появ біграм у навчальному тексті й формуванні статистичної біграмної таблиці.

Програмна складова уведення тексту з використанням біграм призначена для виконання функцій уведення користувачем робочого тексту, виведенню пропозиції введення варіантів наступного слова за допомогою статистичної біграмної таблиці, та для вибору користувачем наступного слова із числа запропонованих варіантів чи введення нового слова.

У класі Form1 розроблено поля і методи: \_bigram, \_bigramTable, \_predictable, Form1\_Load, TabControl1\_SelectedIndexChanged, richTextBox1

TextChanged, richTextBox2\_TextChanged, Insert\_Click, SaveText\_Click,

IntermediateData, PredictableWords. Також є клас BigramLM розроблено методи і поля: setOfSymbol, Train, Predict, GetLast, ToTokens. Класи Form1 і BigramLM наведено у рисунку 3.1.

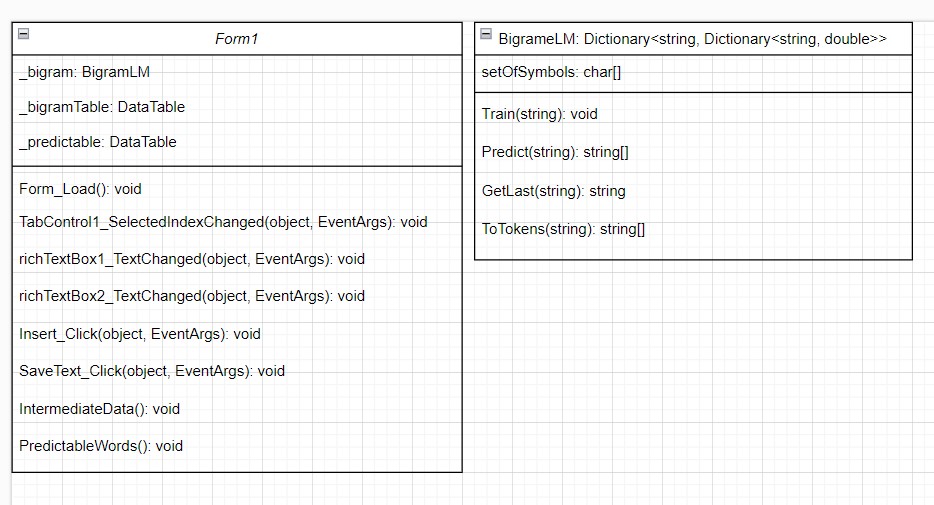


Рисунок 3.1 – Діаграма модулів та відповідних класів Form1 і BigramLM

Отже, таким чином було описано труктуру та функціональне призначення програмних складових системи інтелектуального аналізу для допомоги при введенні тексту з використанням біграм.

## 3.2 Особливості реалізації програмних складових системи передбачення та пропозицій введення слів

За вищеописаною структурою та функціональним призначенням складових було здійснено їх програмну реалізацію.

У програмі розроблено клас форми застосунку Form1 який унаслідується від класу Form та клас біграмної мовної моделі BigramLM який також він наслідує від класу Dictionary. Форму (Form1) застосунку сформовано так: об’єкт класу TabControl що містить вкладки класу TabPage, вкладка навчального тексту tabPage1, вкладка проміжних даних tabPage2, вкладка робота зі текстом tabPage3. У вкладці навчального тексу сформовано поле richTextBox1 класу RichTextBox для вводу навчального тексту та позначка label1 типу Label пояснення що треба ввести навчальний текст. У вкладці проміжних даних є таблиця біграм та їх частотами появи dataGridView1 і типу DataGridView. І останнє це вкладка роботи зі текстом в ній є: поле для праці зі текстом richTextBox2 класу RichTextBox, табличка якому можна вибрати передбаченні слова dataGridView2 класу DataGridView, кнопка Insert класу Button вставки вибраного передбаченого слова і кнопка SaveText класу Button збереження опрацьованого тексту

Реалізація методу GetBigrams, що призначений для виведення біграм за прийнятим рядком тексту та повертає список біграмів наведена нижче:

private List<string> GetBigrams(string text)

{

List<string> bigrams = new List<string>(); string[] words = text.Split(' ');

for (int i = 0; i < words.Length - 1; i++)

{

string bigram = words[i] + " " + words[i + 1]; bigrams.Add(bigram);

}

return bigrams;

}

Цей метод працює шляхом розділення вхідного тексту на слова, а потім послідовного формування біграм за допомогою з'єднання двох сусідніх слів.

Програмний код, що реалізовує запис знайдених біграм у таблицю наведено нижче:

string text = txtInput.Text;

List<string> bigrams = GetBigrams(text);

dataGridView.Rows.Clear(); foreach (string bigram in bigrams)

{

dataGridView.Rows.Add(bigram);

}

Ілюстрація виконаня наведених вище програмних кодів на рисунку 3.2.

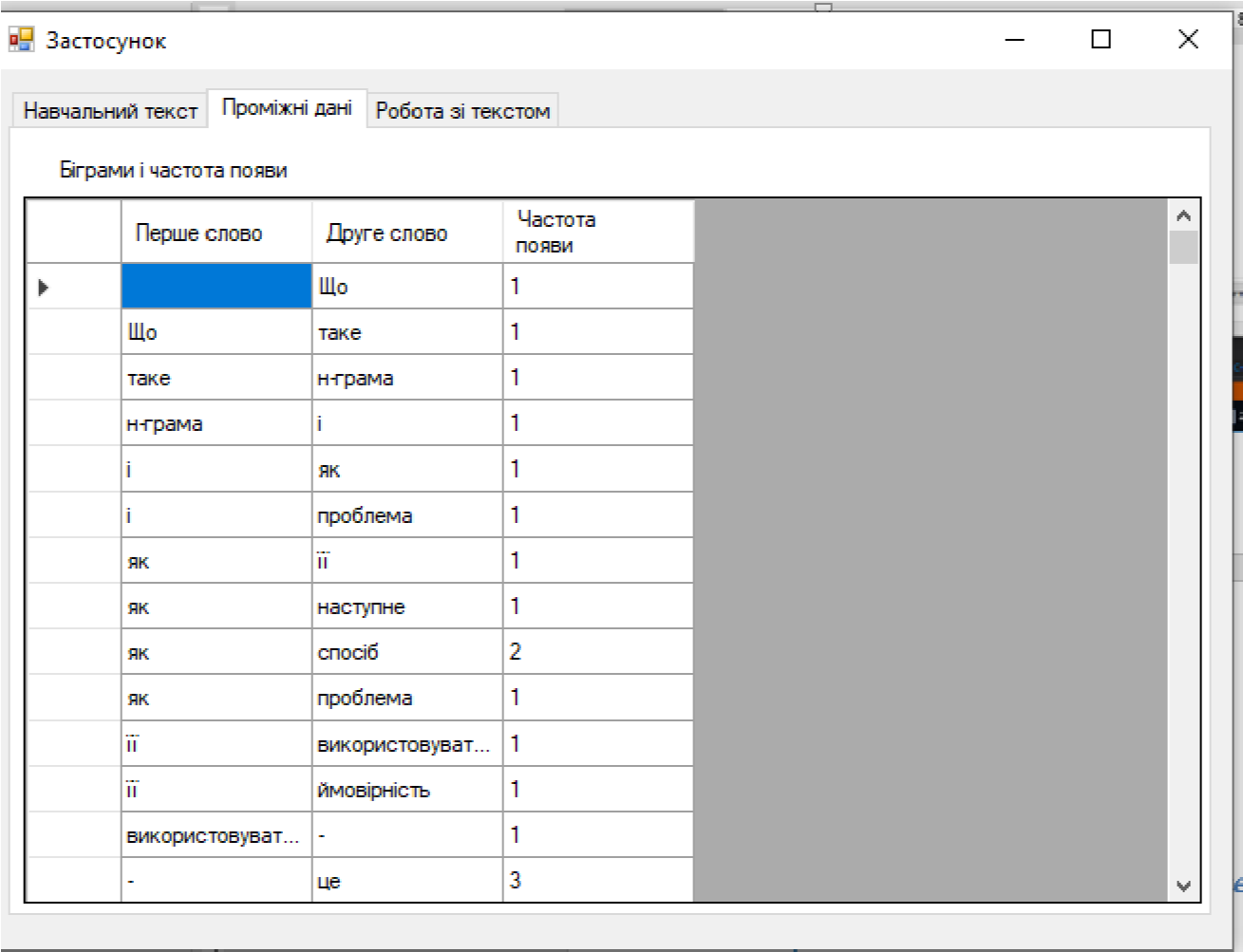
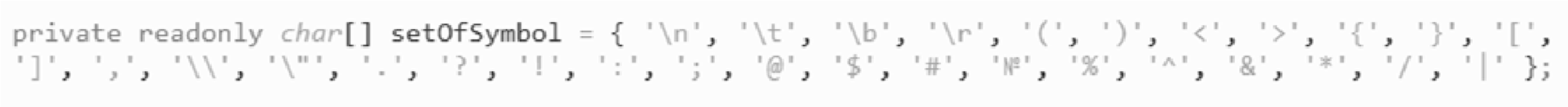


Рисунок 3.2 – Виведення біграм

Найбільш особливим є клас BigramLM, а також в ньому реалізовано такі компоненти: setOfSymbol, Train, Predict, GetLast, ToTokens. Реалізація setOfSymbol відповдає реалізації фільтра в якому конвертується певні символи в пробіл та виглядить так:



У реалізації коду класу Form1 є методи, поля і подій:

* поле \_bigram то об’єкт класу біграмної мовної моделі BigramLM;
* поле \_bigramTable це таблиця проміжних даних;
* поле \_predictable то таблиця передбачених слів;
* подія Form1\_Load виконується при запуску форми для формування dataGridView1 і dataGridView2;
* подія TabControl1\_SelectedIndexChanged виконується при переходу вкладок виконує відображення таблиці проміжних даних;
* подія richTextBox1\_TextChanged виконується при зміні тексту та виконує тренування біграмної мовної моделі;
* подія richTextBox2\_TextChanged виконується при зміні тексту та виконує передбачення слів від біграмної мовної моделі;
* подія Insert\_Click виконує вставку вибраного передбаченого слова у поле richTextBox2;
* подія SaveText\_Click виконує збереження робочого тексту;  метод IntermediateData відображує таблицю проміжних даних;
* метод PredictableWords відображує таблицю передбачених слів.

Отже, таким чином було виконано програмну реалізацію

запропонованого методу інтелектуального аналізу для допомоги при введенні тексту з використанням біграм, що призначається для перетворення вхідних даних у вигляді навчального тексту для формування статистичної біграмної таблиці у вихідні дані у вигляді сформованого користувацького тексту з використанням пропозицій введення варіантів наступного слова за допомогою статистичної біграмної таблиці.

## 3.3 Тестування інформаційної системи

Для перевірки коректності роботи створеного програмного застосунку було проведено функціональне тестування.

Першим тестовим випадком була перевірка функції обробки навчального тексту. Результат успішного проходження перевірки показано на рисунку 3.3.

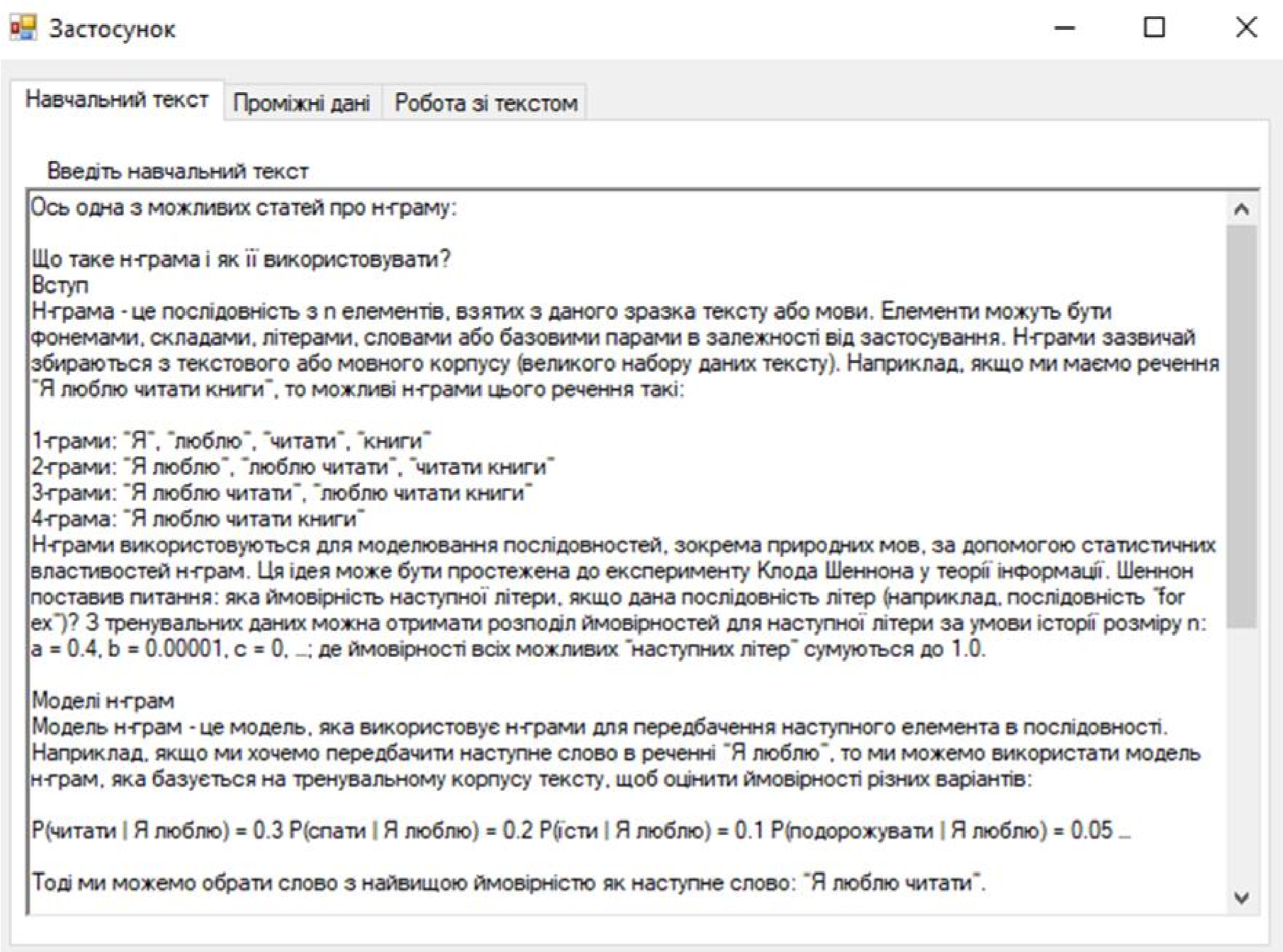


Рисунок 3.3 – Перевірка коректності функції обробки навчального тексту

Наступним тестовим випадком було перевірка формування проміжних даних. Результат коректного виконання перевірки проміжних даних наведено на рисунку 3.4.

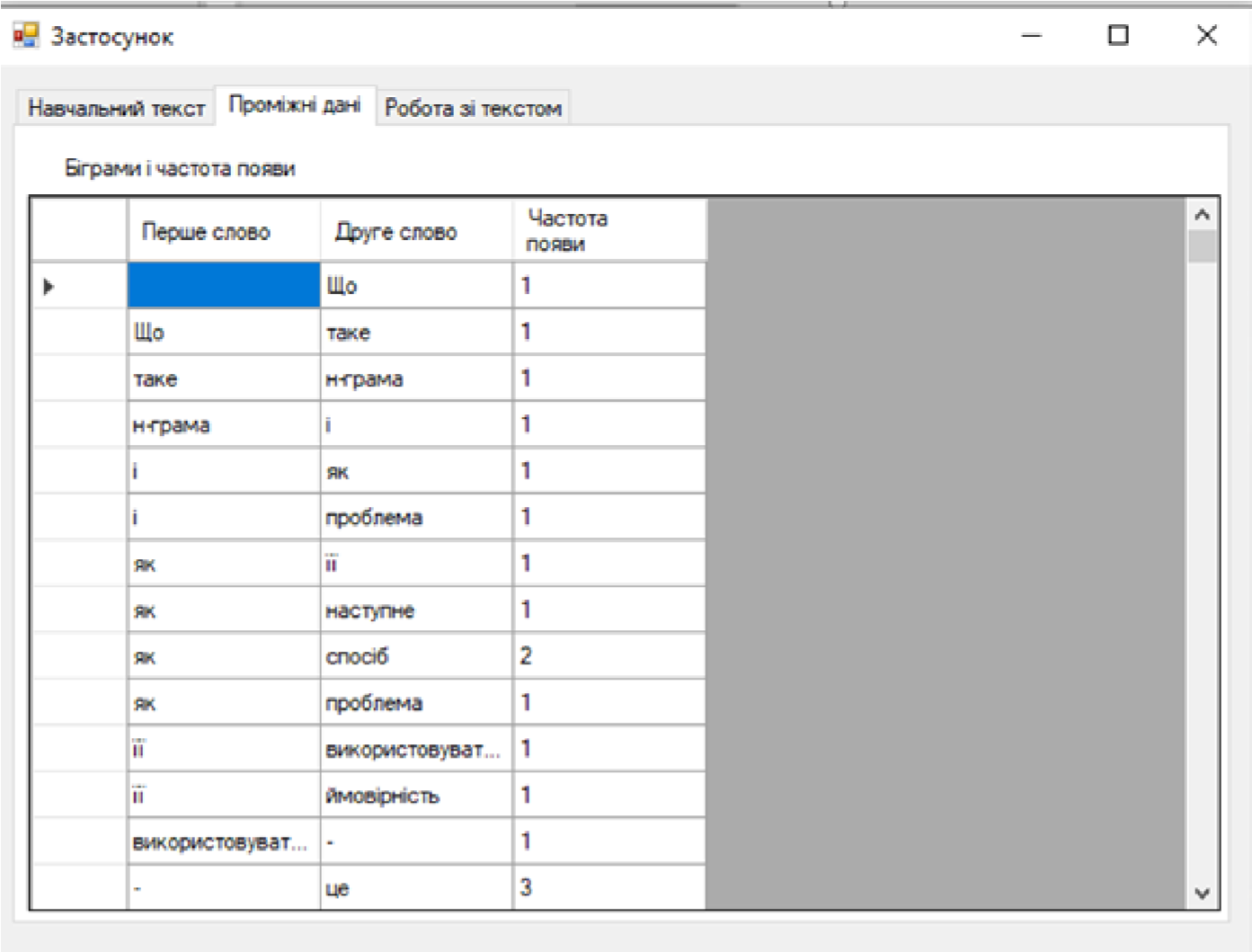


Рисунок 3.4 – Перевірка коректності формування проміжних даних

Наступним тестовим випадком було перевірка функції роботи з текстом. Результат коректного виконання перевірки функції роботи з текстом наведено на рисунку 3.5.

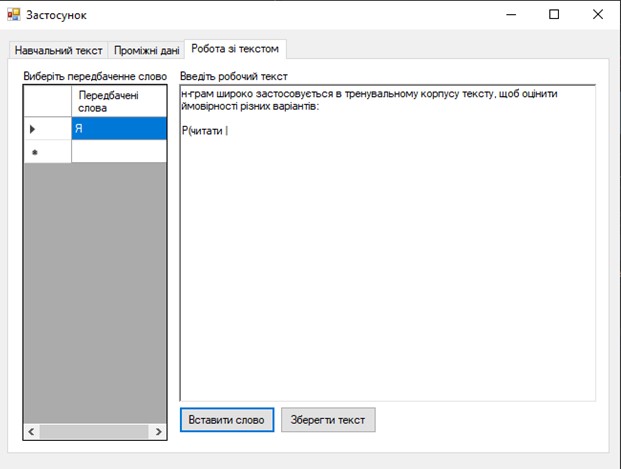


Рисунок 3.5 – Перевірка коректності роботи з тектсом

Отже, після виконаного тестування доведено, що некоректно працюючих функцій немає.

## 3.4 Інструкція користувача

Для користування створеним програмним забезпеченням необхідно запустити програму , та перейти до вкладки «Навчальний текст» (рисунок 3.6).



Рисунок 3.6 – Вкладка «Навчальний текст»

Після заповнення даних можна перейти до вкладки «Проміжні дані», де можна переглянути утворені з навчального тексту біграми (рисунок 3.7).



Рисунок 3.7 – Вкладка «Проміжні дані»

Після перегляду проміжних даних можна перейти до вкладки «Робота з текстом», де користувач зможе згенерувати новий текст на основі статистики зустрічання біграм. Вікно взаємодії зображено на рисунку 3.8.

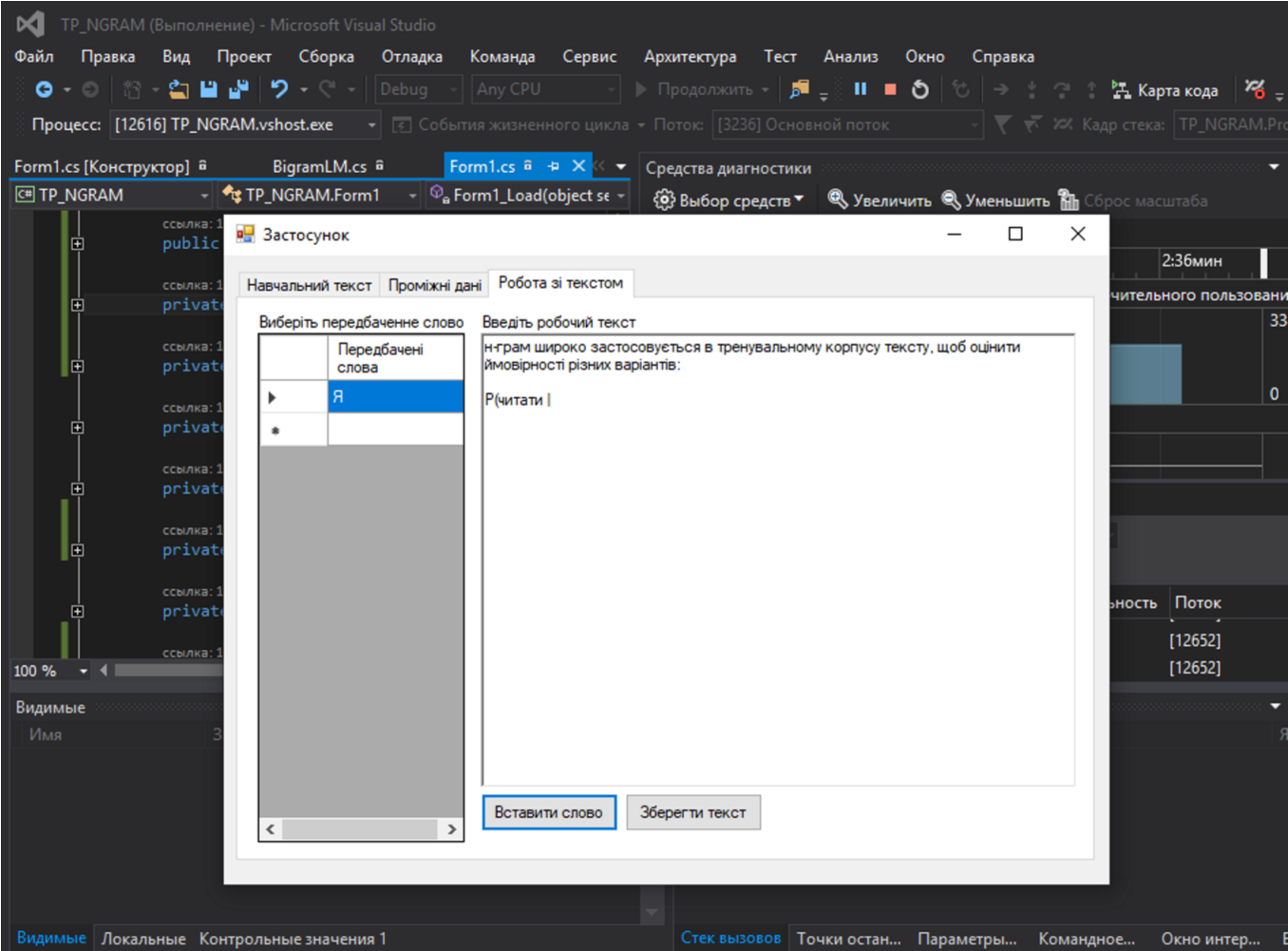


Рисунок 3.8 – Вкладка «Робота з текстом»

Для додавання нового імовірного слова необхідно натиснути кнопку

«Вставити слово» (рисунок 3.9).

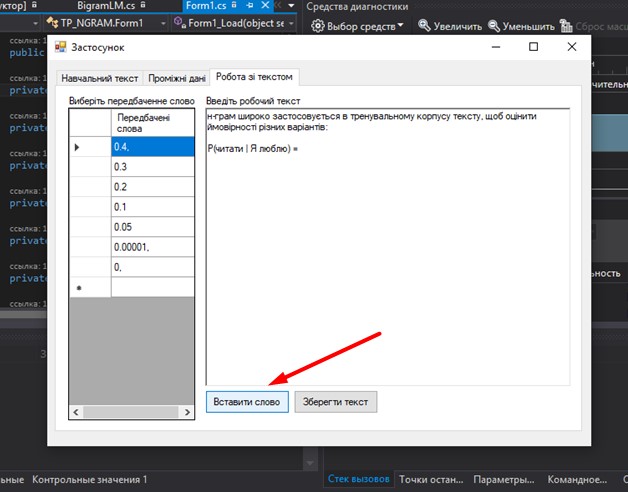


Рисунок 3.9 – Додавання нового слова

Також є можливість зберегти згенерований текст. Для цього необхідно натиснути кнопку «Зберегти текст». Буде виведено діалогове вікно із можливістю збереження тексту (рисунок 3.10).

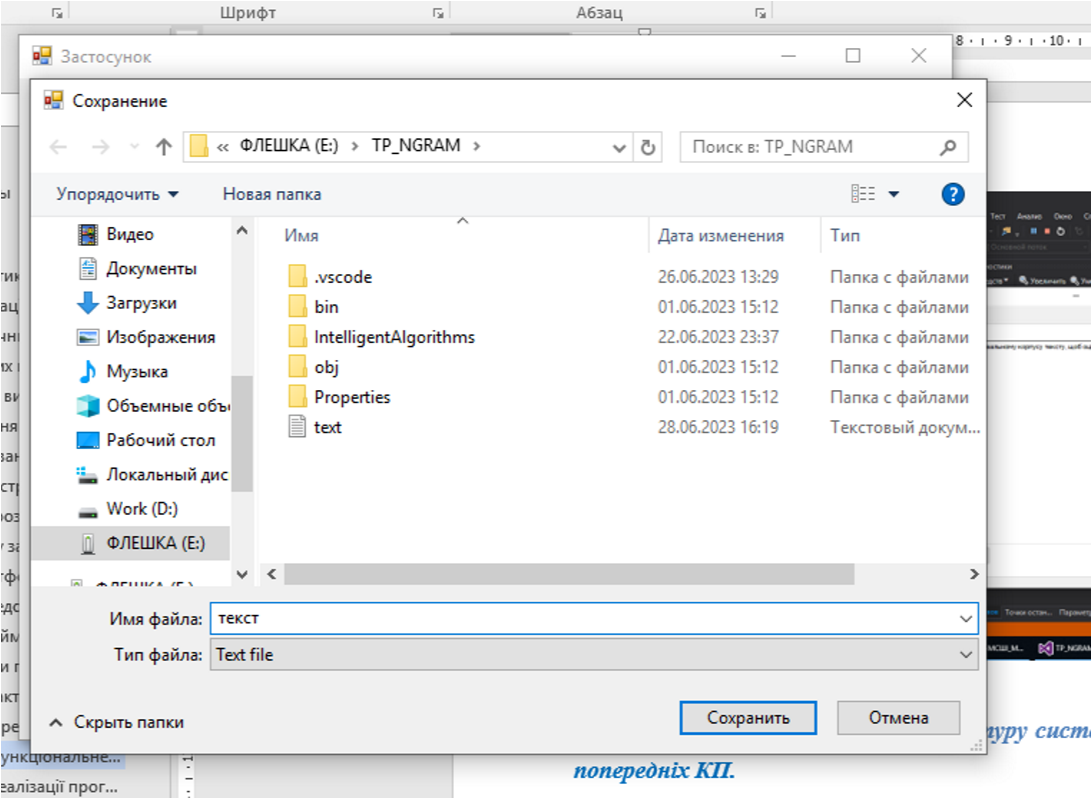


Рисунок 3.10 – Діалогове вікно збереження уведеного тексту

Отже, таким чином було створено інструкцію користувача для організації більш зручної взаємодії майбутніх користувачів та програмного забезпечення.

## 3.5 Вимоги до розгортання інформаційної системи передбачення та пропозицій введення слів

Операційна система: ОС повинна бути сумісною з операційною системою, на якій планується розгортання застосування. .NET підтримує різні операційні системи, такі як Windows, macOS і Linux.

Наявність установленої платформи .NET Framework.

Процесор: Intel Core i3 або еквівалентний, з частотою не менше 1,6 ГГц.

Оперативна пам'ять: Мінімум 4 ГБ оперативної пам'яті.

Місце на диску: Вільне місце на жорсткому диску не менше 10 ГБ для встановлення додатку та збереження даних.

Відеокарта: Карта з підтримкою DirectX 9 або новіше, яка підтримує відображення графіки.

Монітор: Роздільна здатність екрану 1280x800 або вище.

# Висновки

Метою роботи є підвищення ефективності та точності введення тексту шляхом прогнозування найбільш ймовірного наступного слова або фрази на основі попереднього слова, а саме зменшення кількості натискань клавіш, необхідних для введення тексту, підвищення швидкості введення тексту і підвищення задоволеності користувачів процесом введення тексту. При вирішенні задачі використовується великий масив текстових даних для навчання алгоритмів прогнозування, а також метод поєднання частот біграм і довжини слів для точного прогнозування. Крім того подальшим напрямком досліджень визначено можливість адаптуватися до індивідуальних особливостей набору тексту користувачами, щоб з часом покращувати свої прогнози.

В результаті виконання курсового проєкту було розроблено метод застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм та відповідну інформаційну систему передбачення та пропозицій введення слів на платформі .NET, що виконує наступні основні функції:

* Введення навчального тексту для автоматизованого визначення закономірностей чергування слів;
* Очищення навчального тексту від спеціальних символів;
* Переведення тексту до нижнього регістру;
* Формування послідовного масиву слів навчального тексту;
* Формування масиву біграм (пар оригінальних слів) навчального тексту;
* Обрахунок кількості появ біграм у навчальному тексті й формування статистичної біграмної таблиці;
* Введення користувачем робочого тексту;
* Пропозиція введення варіантів наступного слова за допомогою статистичної біграмної таблиці;
* Вибір користувачем наступного слова із числа запропонованих варіантів або введення нового слова;
* Збереження нового користувацького тексту.

Розроблений метод застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм може бути використаний для покращення точності передбачення слів. Також може допомогти користувачу набрати текст, тим самим полегшуючи процес введення тексту.

# Перелік посилань

1. Віртуальна клавіатура Google способи введення URL: [https://www.google.com/inputtools/services/features/virtual-keyboard.html.](https://www.google.com/inputtools/services/features/virtual-keyboard.html)
2. Gboard клавіатура Google Додатки в Google Play URL:

https://play.google.com/store/apps/details?id=com.google.android.inputmethod.latin 3. T9 Wikipedia URL:  [https://en.wikipedia.org/wiki/T9\_%28predictive\_text%](https://en.wikipedia.org/wiki/T9_%28predictive_text%29)

[29.](https://en.wikipedia.org/wiki/T9_%28predictive_text%29)

1. Електронний документ Вікіпедія (wikipedia.org) URL: [https://uk.wikipedi](https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B8%D0%B9_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82)

[a.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0](https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B8%D0%B9_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82)

[%BD%D0%BD%D0%B8%D0%B9\_%D0%B4%D0%BE%D0%BA%D1%83%D0% BC%D0%B5%D0%BD%D1%82.](https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B8%D0%B9_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82)

1. Цифровий текст: характеристика, структура, типи та приклади - Наука - 2023 (warbletoncouncil.org) URL: [https://uk.warbletoncouncil.org/texto-digital15228.](https://uk.warbletoncouncil.org/texto-digital-15228)
2. HMS: предиктивний метод введення тексту за допомогою біграм URL: [https://dl.acm.org/doi/pdf/10.5555/1628195.1628201.](https://dl.acm.org/doi/pdf/10.5555/1628195.1628201)
3. Форматування тексту - Microsoft Style Guide Microsoft Learn URL: [https://learn.microsoft.com/en-us/style-guide/text-formatting/.](https://learn.microsoft.com/en-us/style-guide/text-formatting/)
4. Word: Форматування тексту (gcfglobal.org) URL: [https://edu.gcfglobal.or g/en/word/formatting-text/1/#.](https://edu.gcfglobal.org/en/word/formatting-text/1/)
5. Обробка тексту Вікіпедія (wikipedia.org) URL: [https://en.wikipedia.org/wi ki/Text\_processing.](https://en.wikipedia.org/wiki/Text_processing)
6. Таблиця сиволів Вікіпедія URL: [https://en.wikipedia.org/wiki/Character \_Map\_%28Windows%29.](https://en.wikipedia.org/wiki/Character_Map_%28Windows%29)
7. Посібник про Microsoft Word URL: [https://support.microsoft.com/ukua/office/створення-документа-у-word-aafc163a-3a06-45a9-b451-cb7250dcbaa1.](https://support.microsoft.com/uk-ua/office/створення-документа-у-word-aafc163a-3a06-45a9-b451-cb7250dcbaa1)
8. Ласкаво просимо до довідки LibreOffice Writer URL: [https://help.libreoffice.org/6.4/uk/text/swriter/main0000.html.](https://help.libreoffice.org/6.4/uk/text/swriter/main0000.html)
9. Як працювати застосунком "Google Docs" URL: [https://support.google.com/docs/answer/7068618?hl=uk&ref\_topic=2811805.](https://support.google.com/docs/answer/7068618?hl=uk&ref_topic=2811805)
10. Застосування форматування тексту при введенні (libreoffice.org) URL: [https://help.libreoffice.org/6.2/uk/text/swriter/guide/shortcut\_writing.html.](https://help.libreoffice.org/6.2/uk/text/swriter/guide/shortcut_writing.html)
11. Робота з текстами ScienceDirect URL: [https://www.sciencedirect.com/b ook/9781843347491/working-with-text.](https://www.sciencedirect.com/book/9781843347491/working-with-text)
12. Інтелектуальний аналіз тексту Вікіпедія (wikipedia.org) URL: [https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D0%BB%D0 %B5%D0%BA%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D0%BB%D0%B5%D0%BA%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D1%82%D0%B5%D0%BA%D1%81%D1%82%D1%83)

[%D0%B9\_%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D0%B7\_%D1%82% D0%B5%D0%BA%D1%81%D1%82%D1%83.](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D0%BB%D0%B5%D0%BA%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D1%82%D0%B5%D0%BA%D1%81%D1%82%D1%83)

1. [Інтелектуальний аналіз тексту Wikiwand URL:https://www.wikiwand.c om/uk/%D0%86%D0%BD%D1%82%D0%B5%D0%BB%D0%B5%D0%BA%D1%8 2%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9\_%D0%B0%](https://www.wikiwand.com/uk/%D0%86%D0%BD%D1%82%D0%B5%D0%BB%D0%B5%D0%BA%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D1%82%D0%B5%D0%BA%D1%81%D1%82%D1%83)

[D0%BD%D0%B0%D0%BB%D1%96%D0%B7\_%D1%82%D0%B5%D0%BA%D1 %81%D1%82%D1%83.](https://www.wikiwand.com/uk/%D0%86%D0%BD%D1%82%D0%B5%D0%BB%D0%B5%D0%BA%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D0%B7_%D1%82%D0%B5%D0%BA%D1%81%D1%82%D1%83)

1. Введення тексту Wikipedia URL: [https://en.wikipedia.org/wiki/Typing.](https://en.wikipedia.org/wiki/Typing)
2. Модель мови Вікіпедія (wikipedia.org) URL: [https://uk.wikipedia.org/w iki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C\_%D0%BC%D0%BE %D0%B2%D0%B8.](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%BC%D0%BE%D0%B2%D0%B8)
3. Мовна модель в НЛП Побудова мовної моделі на Python

(analyticsvidhya.com) URL: [https://www.analyticsvidhya.com/blog/2019/08/compreh ensive-guide-language-model-nlp-python-code/.](https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-language-model-nlp-python-code/)

1. Інтелектуальні алгоритми (jhu.edu) URL:<https://ep.jhu.edu/courses/>

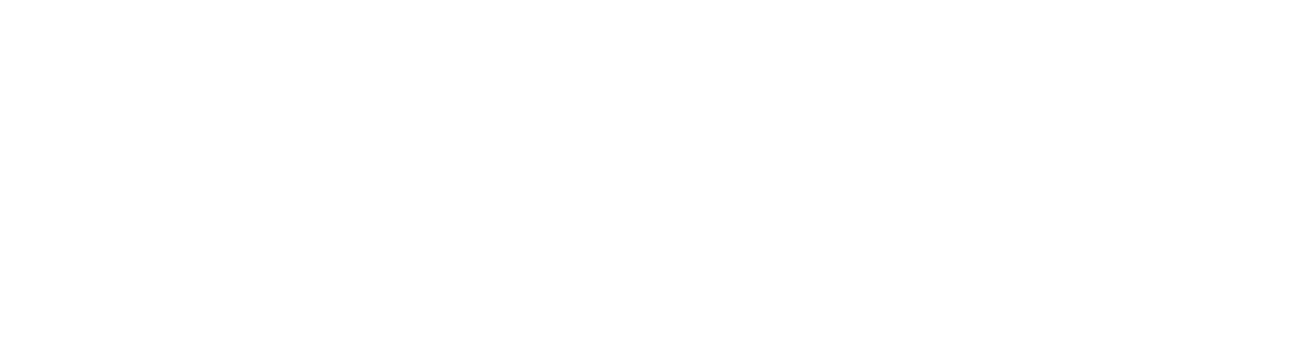
525770-intelligent-algorithms/#:~:text=Intelligent%20algorithms%20are%2C%20in %20many%20cases%2C%20practical%20alternative,when%20information%20on%2 0the%20physical%20system%20is%20limited.

1. Що таке обробка природної мови? MachineLearningMastery.com URL:<https://machinelearningmastery.com/natural-language-processing/#:~:text=What>%20Is%20Natural%20Language%20Processing%3F%201%20Natural%20Language,.. .%206%20Further%20Reading%20...%207%20Summary%20.
2. HMS (lth.se) [URL:https://fileadmin.cs.lth.se/cs/Education/EDA171/Repor ts/2002/HMS.pdf.](https://fileadmin.cs.lth.se/cs/Education/EDA171/Reports/2002/HMS.pdf)
3. Марковська модель Вікіпедія (wikipedia.org) URL: https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D0%BA%D0%BE%D0 %B2%D1%81%D1%8C%D0%BA%D0%B0\_%D0%BC%D0%BE%D0%B4%D0%B 5%D0%BB%D1%8C.

**ДОДАТКИ**

**Додаток А**

**Схема методу застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм**



**Етап 1**

**–**

**Ф**

**ормування статистичної біграмної таблиці**

**:**

7)

Введення навчального тексту для автоматизованого визначення закономірностей

чергування слів;

8)

Очищення навчального тексту від спеціальних символів;

9)

Переведення тексту до нижнього регістру;

10)

Формування

послідовного масиву слів навчального тексту;

11)

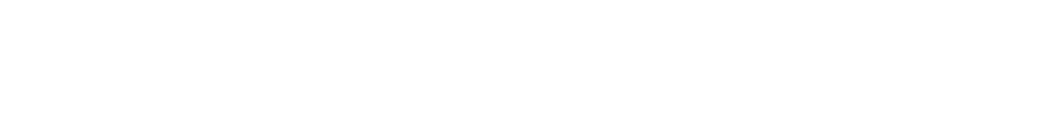
Формування масиву біграм (пар оригінальних слів) навчального тексту;

12)

Обрахунок кількості появ біграм у навчальному тексті й формування статистичної

біграмної таблиці

.



–

Навчальний текст для

формування статистичної біграмної

таблиці

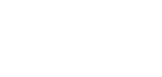
;

–

Процес в

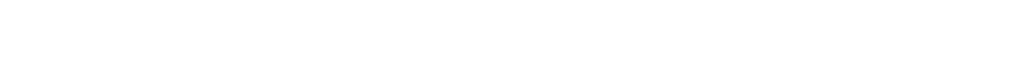
ведення користувачем робочого тексту

.



**Вхідні**

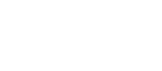
**дані**



Сформований

користувацьк

ий текст.

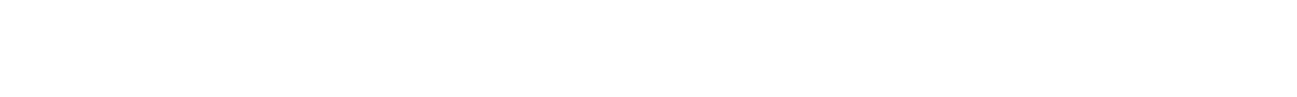


**В**

**и**

**хідні**

**дані**



С

татистичн

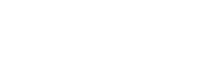
а

біграмн

а

таблиц

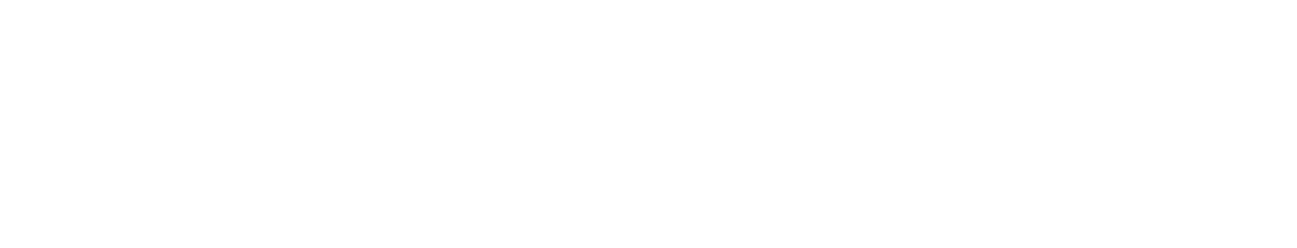
я.



**Проміж**

**ні**

**дані**



**Етап 2**

**–**

**Д**

**опомог**

**а**

**при введенні тексту з використанням біграм**

**:**

5)

Введення користувачем робочого тексту;

6)

Пропозиція введення варіантів наступного слова за допомогою статистичної біграмної

таблиці;

7)

Вибір користувачем наступного слова із числа запропонованих варіантів або

введення

нового слова;

8)

Збереження нового користувацького тексту.

**Додаток Б**

**Діаграма активності методу застосування інтелектуального аналізу для допомоги при введенні тексту з використанням біграм**



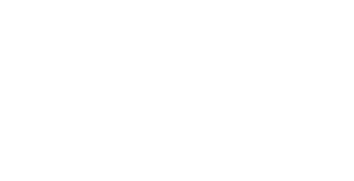
Введення навчального тексту



Початок діяльності



Формування статистичної біграмної таблиці

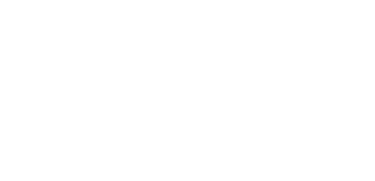


Вивід у дисплей

статистичну бігра

мну

таблицю



Допомога при введенні

тексту з використанням

біграм



Зберегти користувацький текст



Кінець

діяльності

**Додаток В**

**Програмні коди**

Лістинг Form1.cs:

using System; using System.Data; using System.IO; using System.Windows.Forms; using TP\_NGRAM.IntelligentAlgorithms;

namespace TP\_NGRAM

{

public partial class Form1 : Form

{

private BigramLM \_bigram = new BigramLM(); private DataTable \_bigramTable = new DataTable(); private DataTable \_predictable = new DataTable();

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

\_bigramTable.Columns.Add("Перше слово", typeof(string));

\_bigramTable.Columns.Add("Друге слово", typeof(string));

\_bigramTable.Columns.Add("Частота появи", typeof(double)); dataGridView1.DataSource = \_bigramTable;

\_predictable.Columns.Add("Передбачені слова", typeof(string)); dataGridView2.DataSource = \_predictable;

}

private void TabControl1\_SelectedIndexChanged(object sender, EventArgs e)

{

if (tabControl1.SelectedIndex == 1) IntermediateData();

}

private void richTextBox1\_TextChanged(object sender, EventArgs e)

{

\_bigram.Train(richTextBox1.Text);

}

private void richTextBox2\_TextChanged(object sender, EventArgs e)

{

PredictableWords();

}

private void Insert\_Click(object sender, EventArgs e)

{

richTextBox2.Text += " " + dataGridView2.Rows[dataGridView2.CurrentCell.RowIndex].Cells[0].Value;

}

private void SaveText\_Click(object sender, EventArgs e)

{

SaveFileDialog save = new SaveFileDialog(); save.Filter = "Text file|\*.txt|All file|\*.\*";

if (save.ShowDialog() == DialogResult.OK)

{

using (StreamWriter writer = new StreamWriter(save.FileName))

{

writer.Write(richTextBox2.Text);

}

}

}

private void IntermediateData()

{

\_bigramTable.Rows.Clear();

foreach (var first in \_bigram.Keys)

{

foreach (var second in \_bigram[first].Keys)

{

\_bigramTable.Rows.Add(new object[] { first, second, \_bigram[first][second] });

} }

}

private void PredictableWords()

{

\_predictable.Rows.Clear();

foreach (var item in \_bigram.Predict(richTextBox2.Text))

{

\_predictable.Rows.Add(item);

}

}

}

}