Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

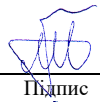
КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему *Інтелектуальна система передбачення слів при введені текстових повідомлень*

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 2 курсу, група КНм-21-1* Курс, група виконавця	 Підпис	*О.В. Мороз* Ініціали, прізвище
Керівник:	*к.т.н., доцент кафедри КН* Науковий ступінь, посада	 Підпис	*Р.О. Багрій* Ініціали, прізвище
Нормоконтроль:	*к.т.н., доцент кафедри КН* Науковий ступінь, посада	 Підпис	*Р.О. Багрій* Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КН, д.т.н., професор
__ *грудня* *2022* р.


Підпис

О.В. Бармак
Ініціали, прізвище

Хмельницький 2022

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет *інформаційних технологій*
Кафедра *комп'ютерних наук*
Освітній ступінь *магістр*
Галузь знань *12 – Інформаційні технології*
Спеціальність *122 – Комп'ютерні науки*

ЗАТВЕРДЖУЮ
Завідувач кафедри *комп'ютерних наук*

_____
(підпис)
д.т.н., професор О.В. Бармак
«*01*» *вересня* *2022* року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

1. Тема кваліфікаційної роботи магістра: *«Інтелектуальна система передбачення слів при введені текстових повідомлень »*

2. Завдання видано студенту *Морозу Олександру Вікторовичу*
(прізвище, ім'я, по батькові)

3. Керівник роботи *доцент кафедри КН Багрій Руслан Олександрович*
(прізвище, ім'я, по батькові)

4. Затверджені наказом університету від «*21*» *липня* *2022* р. № *83*

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

Мета кваліфікаційної роботи магістра полягає у розробці інтелектуальної система прискореного введення текстових повідомлень за рахунок передбачення найбільш ймовірних слів в процесі набору. Для досягнення поставленої мети визначено наступні задачі: провести аналіз методів передбачення слів при введені текстових повідомлень; запропонувати модель мови для оцінки ймовірності слова з урахуванням введеного тексту; розробити метод передбачення слів при введені текстових повідомлень; реалізувати інтелектуальну систему передбачення слів при введені текстових повідомлень; провести валідацію розробленого метода.

Реферат

Кваліфікаційна робота магістра присвячена розробці інтелектуальної системи передбачення слів при введені текстових повідомлень.

Актуальність теми. Кожного дня мільйони людей обмінюються інформацією між собою. Одні з найпопулярніших методів передачі інформації є: текстове, голосове, фото та відео повідомлення. Основним методом лишається передача інформації у вигляді тексту. Оскільки, це зрозуміло для оточуючих.

У світі 7,83 млрд осіб. З них 5,22 млрд у січні 2022 року користувалися мобільними телефонами, 4,66 млрд – інтернетом, 4,2 млрд – соцмережами. Порівняно з 2022 роком населення виросло на 1%, кількість користувачів мобільних телефонів – на 1,8%, інтернету – на 7,3%, соцмереж – на 13,2%.

Враховуючі ці дані в середньому користувач проводить 5 годин вдень відправляючи та отримуючі текстову інформацію. Тому оптимізація роботи користувача з текстом є дуже важливою.

Клавіатура на сучасному етапі розвитку є найбільш універсальним пристроєм для введення інформації. Понад сто років навчають людей швидко друкувати на клавіатурі та з рештою тенденція рухається в сторону зменшення її розмірів, тому ймовірність технічних помилок при наборі текстового повідомлення стає все частіше. Якщо раніше клавіатури були розміром з невеличку тумбу, то сучасна клавіатура вміщується на екрані смартфона.

Технічні помилки виникають з різних причин, але найбільш поширеною помилкою є, коли при введені текстового повідомлення, палець користувача на екрані займає більше місця ніж сама літера.

Проте метод прискореного введення тексту не враховує ці технічні помилки. Отже виникає необхідність в розробці покращеного метода прискореного введення текстів на основі N-грам.

Розробка системи яка буде допомагати. Інтелектуальна система передбачення слів при введені текстових повідомлень. Аналізуючи деякий час листування людей, тексти та інші джерела інформації, можна знайти ймовірність послідовності тих чи інших букв, слів, речень. Все це потрбіно занести в базу

даних та структурувати, після чого ми зможемо ці дані зручно використовувати та покращувати точність передбачення слів.

Таким чином розробка інтелектуальної системи перебачення слів є досить актуальною.

Мета і задачі роботи. Метою кваліфікаційної роботи магістра є розробка інтелектуальної системи прискореного введення текстових повідомлень за рахунок передбачення найбільш ймовірних слів в процесі набору.

Для досягнення поставленої мети потрібно виконати наступні завдання:

- Провести аналіз методів передбачення слів при введені текстових повідомлень;
- Запропонувати модель мови для оцінки ймовірності слова з урахуванням введеного тексту;
- Розробити метод передбачення слів при введені текстових повідомлень;
- Реалізувати інтелектуальну систему передбачення слів при введені текстових повідомлень;
- Провести валідацію розробленого метода.

Об’єкт дослідження - процес передбачення слів при наборі текстових повідомлень.

Предмет дослідження - моделі, методи, підходи та алгоритми прискореного введення текстових повідомлень для мобільних пристроїв.

Методи дослідження. Застосовані для вирішення поставлених завдань: для формування корпусу слів - методи обробки природної мови; для прогнозування тексту – статистичні моделі мови; для реалізації інформаційної системи - методології проектування інформаційних систем та об'єктно-орієнтовний підхід.

Наукова новизна одержаних результатів. В результаті роботи були отримані наступні положення наукової новизни:

Вдосконалено метод предиктивного введення тексту з використанням статистичної моделі мови, що дало можливість прискорити введення текстових

повідомлень. Відмінність від відомих методів полягає в тому, що при передбаченні можливих варіантів слів враховуються "літери-сусіди", які мають високу ймовірність помилкового натискання при наборі тексту на мобільних пристроях.

Апробація результатів кваліфікаційної роботи магістра та публікації. Основні наукові та практичні результати опубліковані у Збірник наукових праць за матеріалами XIV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2022». - Хмельницький, 2022. - С. 206-212.темою роботи: Мороз О. В., Багрій Р.О., Скрипник Т.К. метод передбачення слів при введені текстового повідомлення.

Структура та обсяг роботи. Дипломна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 19 найменувань та 4 додатків. Загальний обсяг дипломної роботи магістра становить 75 сторінок, з них 71 сторінок основного тексту та 30 сторінка додатків. У роботі наведено 21 рисунків.

Ключові слова: мобільні пристрої, N-грама, метод прискореного введення тексту, uni-gram, bi-gram, tri-gram.

Зміст

Перелік скорочень

Скорочення, термін, позначення	Пояснення
АІС	Автоматизована інформаційна система
БД	База даних
ДРМ	Дипломна робота магістра
ІТ	Інформаційні технології
КН	Комп’ютерні науки
ПЗ	Програмне забезпечення
ПП	Програмний продукт
СКБД	Система керування базами даних

Вступ

Актуальність теми. Кожного дня мільйони людей обмінюються інформацією між собою. Одні з найпопулярніших методів передачі інформації є: текстове, голосове, фото та відео повідомлення. Основним методом лишається передача інформації у вигляді тексту. Оскільки, це зрозуміло для оточуючих.

У світі 7,83 млрд осіб. З них 5,22 млрд у січні 2022 року користувалися мобільними телефонами, 4,66 млрд – інтернетом, 4,2 млрд – соцмережами. Порівняно з 2022 роком населення виросло на 1%, кількість користувачів мобільних телефонів – на 1,8%, інтернету – на 7,3%, соцмереж – на 13,2%.

Враховуючі ці дані в середньому користувач проводить 5 годин вдень відправляючи та отримуючі текстову інформацію. Тому оптимізація роботи користувача з текстом є дуже важливою.

Клавіатура на сучасному етапі розвитку є найбільш універсальним пристроєм для введення інформації. Понад сто років навчають людей швидко друкувати на клавіатурі та з рештою тенденція рухається в сторону зменшення її розмірів, тому ймовірність технічних помилок при наборі текстового повідомлення стає все частіше. Якщо раніше клавіатури були розміром з невеличку тумбу, то сучасна клавіатура вміщується на екрані смартфона.

Технічні помилки виникають з різних причин, але найбільш поширеною помилкою є, коли при введені текстового повідомлення, палець користувача на екрані займає більше місця ніж сама літера.

Проте метод прискореного введення тексту не враховує ці технічні помилки. Отже виникає необхідність в розробці покращеного метода прискореного введення текстів на основі N-грам.

Розробка системи яка буде допомагати. Інтелектуальна система передбачення слів при введені текстових повідомлень. Аналізуючи деякий час листування людей, тексти та інші джерела інформації, можна знайти ймовірність послідовності тих чи інших букв, слів, речень. Все це потрбіно занести в базу даних та структурувати, після чого ми зможемо ці дані зручно використовувати та покращувати точність передбачення слів.

Таким чином розробка інтелектуальної системи перебачення слів є досить актуальною.

Мета і задачі роботи. Метою кваліфікаційної роботи магістра є розробка інтелектуальної системи прискореного введення текстових повідомлень за рахунок передбачення найбільш ймовірних слів в процесі набору.

Для досягнення поставленої мети потрібно виконати наступні завдання:

- Провести аналіз методів передбачення слів при введені текстових повідомлень;
- Запропонувати модель мови для оцінки ймовірності слова з урахуванням введеного тексту;
- Розробити метод передбачення слів при введені текстових повідомлень;
- Реалізувати інтелектуальну систему передбачення слів при введені текстових повідомлень;
- Провести валідацію розробленого метода.

Об’єкт дослідження - процес передбачення слів при наборі текстових повідомлень.

Предмет дослідження - моделі, методи, підходи та алгоритми прискореного введення текстових повідомлень для мобільних пристроїв.

Методи дослідження. Застосовані для вирішення поставлених завдань: для формування корпусу слів - методи обробки природної мови; для прогнозування тексту – статистичні моделі мови; для реалізації інформаційної системи - методології проектування інформаційних систем та об'єктно-орієнтовний підхід.

Наукова новизна одержаних результатів. В результаті роботи були отримані наступні положення наукової новизни:

Вдосконалено метод предиктивного введення тексту з використанням статистичної моделі мови, що дало можливість прискорити введення текстових повідомлень. Відмінність від відомих методів полягає в тому, що при передбаченні можливих варіантів слів враховуються "літери-сусіди", які мають високу ймовірність помилкового натискання при наборі тексту на мобільних пристроях.

Апробація результатів кваліфікаційної роботи магістра та публікації. Основні наукові та практичні результати опубліковані у Збірник наукових праць за матеріалами XIV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп’ютерних наук АПКН-2022». - Хмельницький, 2022. - С. 206-212.темою роботи: Мороз О. В., Багрій Р.О., Скрипник Т.К. метод передбачення слів при введені текстового повідомлення.

Структура та обсяг роботи. Дипломна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 14 найменувань та 4 додатків. Загальний обсяг дипломної роботи магістра становить 75 сторінок, з них 71 сторінок основного тексту та 30 сторінка додатків. У роботі наведено 26 рисунків.

Ключові слова: мобільні пристрої, N-грама, метод прискореного введення тексту, uni-gram, bi-gram, tri-gram.

Розділ 1

Аналіз сучасного стану проблеми передбачення слів при введені текстових повідомлень

1.1 Аналіз предметної області

Сьогодні важко уявити своє життя без смартфона. Це всім доступний спосіб зв’язку з рідними та отримання будь-якої інформації. На рис.1 зображений графік пристроїв з яких найбільше заходять в інтернет.

В наший час люди з кожним роком все більш і більше використовують електронні пристрої для обміну інформацією. Тема дуже актуально, тому що як показало життя під час пандемії та війни примусово всі пішли на дистанційне навчання та роботу, а без смартфона, ноутбука чи настільного комп’ютера – це нереально було б реалізувати.

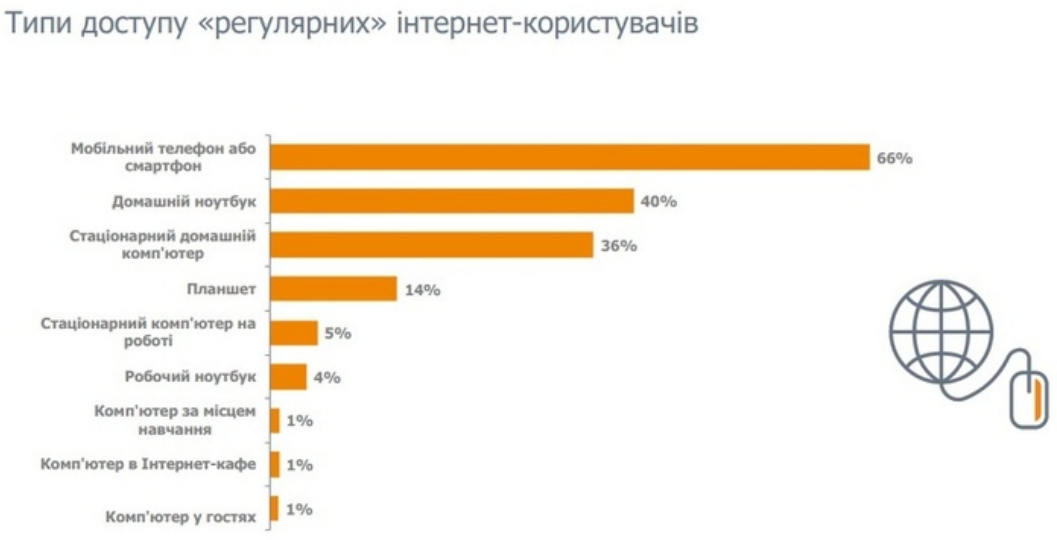


Рисунок 1.1 – Пристрої які найбільш використовуються для доступу в інтернет.

Відповідно зростають обсяги використання мобільних телефоні, а це означає, що розмір клавіатури зменшується через, це процес вводу тексту ускладняється та можуть виникати помилки.

Крім проблем зазначених вище, підвищується ймовірність введення помилок, оскільки розмір клавіатури зменшується. Це в загальному впливає на грамотність всіх користувачів мобільних пристроїв та населення загалом.

1.2 Аналіз сучасних методів прискореного введення тексту

Загалом більшість існуючих рішень розроблені доволі давно, ще за часів використання звичайних кнопкових телефонів. Тоді в нашому розпорядженні було всього 12 кнопок, але всього 9 відповідали за введення букв, так і з’явилась назва методу T9. Проте багато принципів роботи методів можуть бути застосовані в тому числі для електронних пристроїв які використовують сенсорні екрани для введення текстового повідомлення за допомогою сенсорної клавіатури. Такі клавіатури мають окремі кнопки для кожної літери, або більшості літер.

Інакше кажучи, спосіб введення тексту на сучасних мобільних пристроях є доволі схожим з способом введення тексту на клавіатурах з механічними кнопками. Такий тип клавіатур використовують для введення текстового повідомлення на комп’ютерах та ноутбуках.

Методи введення тексту, можуть бути однозначні та багатозначні, вся різниця в тому, що в багатозначному методі, послідовність набраних клавіш, може мати декілька значень, оскільки кожна кнопка відповідає декільком літерам одночасно, а в однозначному методі, кожна кнопка має свою літеру, тому послідовність набраних кнопок буде мати лише одне значення.

Виходячи з цього твердження всі існуючі методи введення тексту на мобільних пристроях можна поділити на групи:

1 – метод одного натискання;

2 – метод багатьох натискання;

3 – метод предиктивного введення тексту на основі N-грам.

Тому, розглянемо їх більше детально.

Метод одного натискання

Методи одного натискання - методи введення тексту, що направлені на зниження кількості натискань для введення одного символу. Такі методи є багатозначними, тому, що послідовність натискань клавіш під час введення текстового повідомлення, буде відноситися одночасно до декількох слів з словника[1]. Однак, самі методи одного натискання створені для вирішення цієї багатозначності завдяки створеним для них словникам, або так званим моделям статистичної мови.

Метод багатьох натискань

Методи багатьох натискань – методи введення тексту, згідно яких введення тексту повідомлення забезпечується шляхом натискання більше однієї клавіші/кнопки для введення одного символу. Такі методи дозволяють здійснювати однозначне введення тексту тому, що під час введення тексту користувач вводить кожну літеру окремо від одної за допомогою клавіатури QWERTY на мобільному телефоні, комп’ютері чи ноутбукові, тому однозначно обирає символ, який необхідно відобразити на екрані пристрою[2]. Методи багатьох натискань можуть застосовуватися зокрема самі, або ж використовуватися як частина великого та складного комплексу для введення тексту. Методи багатьох натискань не вимагає використання словника для забезпечення введення тексту.

Методи прискореного введення тексту

Метод прискореного введення тексту - це технологія введення, яка полегшує введення тексту на пристрої, пропонуючи слова, які користувач може забажати вставити в текстове поле. Прогнози засновані на контексті інших слів у текстовому наборі та перших набраних літерах. Користувач просто торкається слова замість того, щоб вводити його на клавіатурі комп'ютера чи програмній клавіатурі мобільного пристрою . Метод предиктивного введення тексту може значно прискорити процес введення.

Метод прискореного введення тексту використовує машинне навчання, щоб підбирати словник слів та фраз, які користувач часто вводить. Потім функція інтелектуального введення тексту сортує ці часто вживані слова та фрази, щоб спрогнозувати, коли вони будуть використані знову.

Одним із найперших застосувань інтелектуального введення тексту був T9 (текст на 9 клавішах). Запатентована технологія T9, спочатку розроблена Мартіном Кінгом та іншими винахідниками в Tegic Communications, яка зараз є частиною Nuance Communications. T9 був винайдений у 1995 році та полегшив набір тексту на мобільних телефонах та інших невеликих пристроях. До T9 введення тексту на мобільному пристрої вимагало багаторазового натискання, коли користувачеві доводилося натискати цифри до чотирьох разів, щоб обрати бажану літеру[3].

T9 - це технологія інтелектуального введення тексту, яка переважно використовується в кнопкових телефонах і пристроях зі стандартними дев'яти клавішними клавіатурами та інколи зустрічалась на телефонах з сенсорним екраном. Технологія T9 дозволяє вводити слова лише кількома натисканнями клавіш. Введення тексту на дев'яти клавішах полегшує та пришвидшує введення повідомлень, але на зміну прийшли нові технології та витіснили її.

T9 покращив роботу мобільного користувача, зв’язавши блоки-літер на кожній телефонній клавіші зі словами в словнику. Програмне забезпечення в пристрої спів ставляло послідовності натискань клавіш зі словами в словнику та визначало пріоритети за частотою використання.

Незважаючи на те, що технології прогнозування тексту стають дедалі складнішими, програмне забезпечення все ще дуже схильне до помилок.

T9 розроблений, щоб стати розумнішим на основі слів, які вводить користувач. Після введення певних цифр T9 шукає слова у своєму швидкодоступному словнику. Якщо числова послідовність може давати різні слова, T9 відображає слово, яке найчастіше вводить користувач.

Якщо вводиться нове слово, якого немає у словнику T9, програмне забезпечення додає його до своєї бази даних прогнозування, щоб воно відображалося наступного разу. Хоча T9 може навчатися на основі досвіду користувача, він не завжди правильно вгадує слово, йому потрібно. Якщо за допомогою однієї цифрової послідовності можна створити кілька слів, вони називаються текстонімами .

T9 також може вивчати пари слів, які ви часто використовуєте, щоб передбачити наступне слово. Наприклад, T9 може здогадатися, що ви збираєтеся ввести «ти як» після «привіт», якщо ви часто використовуєте «ти як».

Метод прискореного введення LetterWise

Метод LetterWise - метод предиктивного введення тексту. Даний метод дозволяє здійснювати введення тексту без використання великих словників[4]. Він працює зі збереженою базою даних ймовірностей префіксів. Префікс - це літери, що стоять перед поточним натисканням клавіші. Так, в українській мові після літери «п» у слові часто зустрічається літера «р», однак дуже рідко зустрічається літера «ї».

Найбільшим недоліком є те, що LetterWise не використовує словник збережених слів. Проте завжди аналіз словника використовується для визначення ймовірнісної інформації про послідовності літер у мові. Це дозволяє ефективно вводити слова та, на відміну від підходів, заснованих на словнику.

Продуктивність LetterWise покращується з урахуванням кількості попередніх літер. У LetterWise покращена продуктивність, тому це означає, що для метода потрібно менше натискань клавіш для введення слова. Бази даних LetterWise зберігають інформацію про вибрану підмножину префіксів. На практиці вимоги до пам'яті коливаються приблизно від 500 байт до 9000 байт, що є дуже хорошим результатом в плані економії пам’яті.

Оскільки LetterWise базується на префіксі, а не на словнику, він не дає катастрофічних помилок, наприклад, коли користувач намагається ввести слово поза словником, наприклад власний іменник, абревіатуру чи сленг.

Завдяки даному методу можливо досягнути досить малої кількості натискань, що вистачить для введення одного елемента.

Перевагою, є можливість працювати окремо та з більш складними методами, замість методів багатьох натискань, оскільки має вищу швидкість введення текстового повідомлення.

Методи прискореного введення тексту повідомлення за допомогою N-грам

Методи прискореного введення текстового повідомлення, що описані вище, мають недоліки. Під час роботи прискореного метода, прогнозуються ті слова, які користувач вводить в цей момент. Використовуючи при цьому словник з літерами, словами які не мають частоти використання, ігноруючи статистичні моделі мови в яких дана інформація присутня. Методи прискореного введення текстового повідомлення за допомогою N-грам , дають можливість прогнозувати наступне слово на основі попередніх слів [5]. Це

стало можливо через використання послідовності слів, отриманих, як правило, в результаті оброблення текстових корпусів тематичних текстів відповідної мови.

Текстовий корпус - це великий і структурований набір текстів, який використовується для наукових досліджень, статистичного аналізу та перевірки гіпотез, перевірки випадків або перевірки лінгвістичних правил на певній мовній території. N-грамою називають послідовність, що складається з N елементів. Елементами такої послідовності можуть бути літери, склади та слова. Зазвичай використовуються N-грами літер та слів. Послідовність слів створюють текстове повідомлення. Тобто до складу N-грами входять елементи, що розташовуються послідовно одне за одним, послідовність літер у слові або послідовність слів у тексті. N-грами які складається з одного елементу, називається uni-gram. N-грама, що складається з двох елементів має назву bi-gram. В свою чергу послідовність з трьох елементів називають tri-gram. Інколи використовують числові позначення для N-грам як складаються більше ніж з трьох елементів 4-gram, 5-gram і тому подібне[6].

Приклад процесу створення uni-gram, bi-gram та tri-gram зображено на рис. 2.

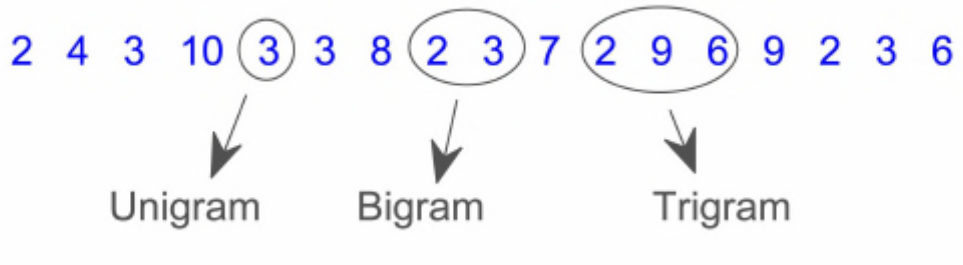


Рисунок 1.2 – Приклад процесу створення uni-gram, bi-gram та tri-gram

Більшість методів прискореного введення тексту, для прогнозування ймовірність використовують N-грам літер, проте даний метод в прогнозуванні

використовує ймовірність появи N-грам слів. Послідовність дій для створення N-грам слів на основі тематичного тексту, така сама як і для N-грам літер.

Завдяки даному методу можливо прогнозувати наступне слова одразу після закінчення введення попереднього. Це дозволяє з великою ймовірністю вводити слова за допомогою лише одного натискання. В результаті, середнє статистичне значення кількості натискань літер, потрібних для введення слова, має великий потенціал отримати значення, менше за одиницю.

1.3 Дослідження інформаційного забезпечення для предиктивного вводу тексту

Технологія T9 працює з комбінацією групи літер, введених натисканням клавіш та легким способом використовувати словник. У протилежність методу багаторазового натискання, який використовується при введенні тексту на звичайній клавіатурі, слова вводяться одним натисканням кожної клавіші, що представляє букву. Технологія шукає в словнику можливі слова, які відповідають послідовності введених ключів і частоті вживання. Іншими словами, слова передбачаються з меншою кількістю натискань клавіш, і користувачеві не потрібно вводити все слово. Слова, які часто вживаються, але не внесені до словника, можна додати до словникової бази даних. Вибір слів і процес стають швидшими, чим більше вони використовуються, оскільки текст на дев’яти клавішах стає знайомим із часто використовуваними фразами та словами користувача[7].

T9 дозволяє вводити цілі слова одним натисканням клавіші на літеру, замість того, щоб натискати клавішу кілька разів, щоб обертати всі можливі літери, доки не дійдете до потрібної. Наприклад, використовуючи метод мультитап без T9, вам доведеться натиснути «4» шість разів, щоб отримати

літеру «Л». Приклад клавіатури з якою використовували метод Т9 зображено на рис. 3

Рисунок 1.3 – Приклад клавіатури Т9 на мобільному телефоні

Подумайте про необхідність написати слово «добре»: ви б почали з «3», щоб отримати «д». Щоб отримати «о», вам потрібно було б проклацати «5» три рази, потім ще два рази «2», для букви «б», а якщо Т9 увімкнено, вам потрібно натискати кожну цифру лише один раз на літеру: «35263». Це тому, що Т9 «навчається» на основі досвіду користувача та зберігає зазвичай- використав слова у своєму інтелектуальному словнику.

Текст на дев'яти клавішах допомагає зменшити кількість натиснень клавіш і натиснень для отримання потрібного слова. Це, у свою чергу, призводить до швидшого набору тексту та часто друкування однією рукою. Технологія Т9 також може бути допоміжною технологією для людей з обмеженими можливостями.

Тому Т9 найчастіше використовується в службі коротких повідомлень і в протоколі бездротових додатків.

iTap – це технологія прогнозування введеного тексту для мобільних телефонів. Створена американською телекомунікаційною компанією « Motorola » як конкурент технології прискореного введення тексту T9 для використання в своїх пристроях, була ліцензована іншими компаніями[8]. Він був розроблений як заміна старим відображенням літер на телефонах, щоб полегшити введення слів. Це полегшує обмін текстовими повідомленнями та створення нотаток.

При введенні трьох або більше символів поспіль iTap вгадує решту слова. Наприклад, введення "прог" запропонує "програма". Якщо потрібно інше слово, наприклад «програміст» або слова, утворені різними літерами, але потребують однакових натискань клавіш, наприклад «продовження» або «пропонувати», можна натиснути клавішу зі стрілкою, щоб виділити інші слова в меню для вибору, щоб спадної спільності їх використання.

Подібно до T9, iTap також може завершувати слова та фрази. iTap вгадує найймовірніше слово на основі вбудованого словника, включаючи слова з введеним префіксом. Цей словник також містить фрази та загальновживані речення. Таким чином список прогнозування, який пропонує iTap, покращуються на основі контексту слова, яке вводить користувач.

iTap зазвичай використовує інший інтерфейс користувача UI, ніж T9. Однак T9 надає API, який можна використовувати для створення подібного інтерфейсу користувача, якщо виробникам телефоні потрібно буде це реалізувати. iTap приступає прогнозувати завершення слів вже після першого натискання клавіші в усіх випадках. Однак T9 доповнює власні слова після одного натискання клавіші, а на більшості телефонів решта слів, які користувачі вводили раніше, можна отримати після трьох натискань клавіш. T9 дозволяє приймати рішення щодо інтерфейсу користувача в основному виробнику

телефону, і наразі жоден із них не вирішив наслідувати інтерфейс iTap за допомогою T9.

QuickType - це інтелектуальна клавіатура Apple, яка стає розумнішою, коли ви її використовуєте. Оскільки QuickType розуміє контекст вашої розмови, може пропонувати відповіді на типові запитання[9]. І залежно від того, з ким ви листуєтеся, QuickType може підібрати навіть стиль вашої розмови. Все, що вам потрібно зробити, це почати використовувати його.

Тепер є можливість писати цілі речення кількома дотиками. Оскільки під час введення тексту пропонуються вибір слів або фраз, які, ймовірно, введете наступним, на основі ваших попередніх розмов і стилю написання. IOS враховує невимушений стиль, який користувач може використовувати в повідомленнях, і більш офіційну мову, яку ймовірно, використовуєте в листуванні, навчанні чи під час роботи. Саморегулювання залежно від людини, з якою ви спілкуєтеся, тому що ваш вибір слів, ймовірно, більш невимушений з вашим другом, ніж з вашим викладачем чи керівником. Дані ваших розмов зберігаються лише на вашому пристрої, тому вони завжди приватні.

Інтелектуальний текстовий механізм IOS оптимізований для мов у всьому світі. Це означає, що ви побачите запропоновані слова та фрази, які підходять для вашої мови. І коли ви з часом користуєтеся клавіатурою, вона навчиться вашому спілкуванню, дізнається ваші улюблені фрази та запропонує логічне наступне слово.

1.4 Постановка задачі

Мета кваліфікаційної роботи магістра полягає у розробці метода передбачення слів при введені текстових повідомлень, який при введені літер

буде прогнозувати найбільш ймовірне слово. Також потрібно створити інформаційну систему для виявлення ефективності роботи.

Для досягнення поставленої мети потрібно виконати наступні завдання:

- провести аналіз методів передбачення слів при введені текстових повідомлень;
- запропонувати модель мови для оцінки ймовірності слова з урахуванням введеного тексту;
- розробити метод передбачення слів при введені текстових повідомлень;
- реалізувати інтелектуальну систему передбачення слів при введені текстових повідомлень;
- провести валідацію розробленого метода.

Результатом виконання кваліфікаційної роботи магістра створено тестовий застосунок на якому, це все можна перевірити. Оскільки застосунок навчається на базах даних, його можна використати для будь-якої мови світу.

Висновки до розділу 1

Таким чином, розглянувши методи прискореного введення тексту на мобільних пристроях та програмні засоби створені на основі цих методів, зрозуміло, що методи прискореного введення тексту повинні бути гнучкими і пристосовуватись для введення тексту різними користувачами та використовувати при майбутніх прогнозуваннях, слова, які користувач вводив раніше. При тенденції зменшення екранів мобільних пристроїв, потрібно враховувати велику ймовірність введення помилкових літер, що знаходяться

поруч з потрібною. Тому, метод прискореного введення тексту при формуванні пропозиції ймовірних слів, повинен враховувати ці нюанси.

Розділ 2

Розробка метода передбачення слів при введені тексту

2.1 Вибір метода прискореного введення тексту для модифікації

Методи одного натискання, що використовують для своєї роботи статистичний опис даних літер алфавіту та зрештою, мають можливість прогнозувати наступні літери під час набирання текстового повідомлення, дають можливість значно покращити ефективність введення тексту в порівняння з методом багатьох натискань. Тому, значення оцінки критерія KSPC для методу одного натискання в загальному має нижчий результати, порівнюючи його з методом багатьох натискань. KSPC - це кількість натискань клавіш, в середньому, для створення кожного символу.

Метод багатьох натискань під час введення текстового повідомлення, не гарантує змоги прогнозування тому, що потребують більше одного натискання для введення одної літери. Отже, методи багатьох натискань не дадуть можливості досягати значень виміру KSPC, менших за одиницю, через те обрання таких методів для модифікації є недоцільним.

Врахувавши всі особливості введення тексту на мобільних пристроях, часто може траплятися ситуація, коли користувач має намір ввести слово, якого нема у словнику. В такому випадку, дане слово чи словосполучення не буде знаходитися у списку слів пропозиції для користувача. Більше того, щоб надати можливість користувачеві реалізувати введення слова, що відсутнє у словнику, необхідним є застосування звичайного методу багатьох натискань у якості допоміжного методу до методу одного натискання. Такий допоміжний метод забезпечить можливість однозначного введення необхідного слова користувачем.

Проте, така висока залежність від словника, відповідно зменшує швидкість введення тексту при частому введенні слів, що відсутні у словнику.

Одним з варіантів, вирішення цієї проблеми, буде використання підходу з прискореного методу введення тексту LetterWise. В методі прискореного введенні тексту LetterWise використовуються збережені бази даних ймовірностей префіксів. Префікс - це літери, що стоять перед поточним натисканням клавіші.

Здебільшого методи прискореного введення тексту не дають змоги здійснювати введення слова, кількістю літер, меншою ніж кількість літер в слові. Отже, такі методи не уможливлюють досягнути мінімальних значень критерія оцінки KSCP, що буде менше за одиницю. Проте методи одного натискання які мають можливість прогнозування закінчень слова, можуть при деяких обставинах приближуватися значень критерія оцінки KSCP, меншого за одиницю, не дають можливості реалізовувати прогнозування наступного слова з врахуванням вже введеного слова.

Більше того, для максимального зменшення значення критерія оцінки KSCP, потрібно використовувати такі методи, як методи прискореного введення текстового повідомлення на основі N-грам. Що в майбутньому дозволить реалізовувати прогнозування закінчення слів, ще на початковому етапі їхнього введення, тобто, коли користувач розпочне вводити перші літери, метод вже буде пропонувати список ймовірних пропозицій закінчення слів, так і можливість прогнозування наступного слова, враховуючи попереднє.

Однозначно, при умові коли бажане слово, після введення попереднього слова, знаходиться в списку пропозиції для введення, такі методи дають змогу завдяки кільком натисканням, вибрати слово з списку пропозицій для введення.

Таким чином, для слова, що складається з 4 літер, значення KSPC становитиме 0,25, а для слова з 6 літер всього 0,17. Тому, в якості методу для модифікації використовуватимемо метод прискореного введення тексту на основі N-грам.

2.1.1 Аналіз ефективності технології предиктивного введення слів

Для того, щоб визначити метод введення тексту та порівняти ефективності різних методів між собою потрібно обрати критерій ефективності.

KSPC - це кількість натискань клавіш, в середньому, для створення кожного символу тексту на даній мові за допомогою певної техніки введення тексту. Ми систематично описуємо обчислення KSPC і надаємо приклади різноманітних методів введення тексту. Доведено, що KSPC є корисним для порівняння методів введення тексту перед великими змінами та оцінками[10].

Важливим напрямком досліджень телефонів, смартфонів, ноутбуків, настільних комп'ютерів є розробка ефективних засобів введення тексту. Інтерес підігрівається такими тенденціями, як обмін текстовими повідомленнями на мобільних телефонах, мобільний Інтернет і доступ до електронної пошти. Запропонований критерій для характеристики методу введення тексту. Він обчислюється зазвичай з використанням мовної моделі та опису техніки на рівні натискання клавіш. Цей показник використовується для характеристики та порівняння методів на етапі проектування, таким чином полегшуючи аналіз перед глобальними змінами та оцінюванням.

KSPC - це також абревіатура для натискань клавіш на символ . Це кількість натискань клавіш, у середньому необхідна, щоб згенерувати символ тексту для певної техніки введення тексту на даній мові. Критерій KPSC ≠ 1 для

певних методів введення тексту, було зазначено раніше [наприклад, 1, 7, 8]. Використовується значення :

і - систематично описати обчислення KSPC ,

іі - надати приклади KSPC для широкого діапазону методів введення тексту, деякі з KSPC > 1, інші з KSPC < 1,

ііі - продемонструвати корисність KSPC як інструменту для аналізу.

Поширена клавіатура **«Qwerty»** служить корисною базовою умовою для вивчення KSPC. По-перше, розглядаючи лише малі літери, зрозуміло, що клавіатура Qwerty є однозначною, оскільки кожна літера має окрему клавішу. Іншими словами, кожне натискання клавіші створює символ тексту. Враховуючи це, робимо наступний висновок для базової клавіатури Qwerty:

$$KSPC = 1{,}0000$$

Зрозуміло, значення трохи зросте, якщо врахувати, використання клавіші «shift». Але зрештою дуже хороший результат. Проте процес не такий простий для інших клавіатур і технік. Є дві основні вимоги до обчислення KSPC. Для початку, це чіткий опис техніки введення на рівні натискання клавіш. Використовуємо під час представлення кожної техніки. По-друге, мовна модель, яка необхідно для нормалізації KSPC, тому це «середнє значення», що відображає як метод взаємодії, так і мову користувача.

Для кожного слова ми визначаємо натискання клавіш для введення слова в цікаву техніку взаємодії. З цією інформацією KSPC обчислюється таким чином:

де K w - кількість натискань клавіш необхідних для введення слова, C w - кількість символів у слові, а F w - частота слова в корпусі. Важливо, що K w і C

w налаштовані таким чином, щоб після кожного слова ставився кінцевий ПРОБІЛ.

CPS - це абревіатура символів за секунду, так звана одиниця виміру, яка використовується для опису швидкості передачі даних, оцінена на основі швидкості передачі даних і довжини символу[11]. Наприклад, зі швидкістю 2400 біт/с 8-бітові символи з початковими бітами (загалом десять біт на символ) передаються зі швидкістю приблизно 240 символів на секунду (cps). Швидкість лазерних і струменевих принтерів описується у вигляді сторінок за хвилину.

Символів за секунду також використовується при введенні даних для оцінки швидкості набору тексту людиною. CPS - це загальна кількість введених символів, поділена на час, витрачений на їх введення.

Ми починаємо з вивчення методів введення тексту, які вимагають більше одного натискання клавіші на символ.

На рис. 4 зображено інфографіку швидкості друку, підрахована критерієм CPS. Результати можуть змінюватись в залежності від навиків користувача.

Рисунок 2.1 Інфографіка швидкості друку

В свою чергу видно середній діапазон значень критерія CPS, що дає зрозуміти чи буде актуальна наша робота.

Проте критерій оцінки KSPC дає можливість, визначати ефективність методів прискореного введення тексту, без залежності від швидкості введення текстового повідомлення користувачем..

2.2 Модифікація методу прискореного введення тексту на основі N-грам

Методи прискореного введення тексту, що створені на основі N-грам дозволяють реалізувати прогнозування ймовірності наступного слова, що користувач бажає ввести, опираючись на дані про частоту повторювання

комбінацій слів в текстах мови, на якій здійснюється введення тексту. Ці послідовності слів називають N-грамами, де число N відповідає кількості слів у цій послідовності. Коли значення N-грами прирівнюється одиниці в даному випадку є слово, така послідовність називається uni-gram, а ймовірність використання послідовностей, що складаються з двох слів та більше застосовується, для того щоб прогнозувати наступне ймовірне слово після введення попереднього. Повторюваність застосування uni-gram, раціонально використовувати для прогнозування закінчень слова, перші літери якого на даний момент введено користувачем.

2.2.1 Визначення ймовірності послідовності

Зазвичай для N-грам, або деяких текстових корпусів для визначення ймовірності появи, використовують метод MLE - метод максимальної правдоподібності.

Метод максимальної правдоподібності займає головне місце в теорії статистичного оцінювання параметрів[12]. На нього свого часу зауважував Карл Гаусс, а розробив його Рональд Фішер. У статистиці оцінка максимальної правдоподібності MLE - це метод оцінки параметрів передбачуваного розподілу ймовірностей за деякими спостережуваними даними, який полягає у знаходженні максимуму функції одного або кількох оцінюваних параметрів. Це досягається шляхом максимізації функції правдоподібності таким чином, щоб згідно з припущеною статистичною моделлю спостережувані дані були найбільш імовірними. Точка в просторі параметрів , яка максимізує функцію правдоподібності , називається оцінкою максимальної правдоподібності. Логіка максимальної ймовірності є інтуїтивно зрозумілою та гнучкою, тому цей метод став домінуючим засобом статистичний висновок .

Якщо функція ймовірності є диференційованою , можна застосувати тест похідної для знаходження максимумів. У деяких випадках умови першого порядку функції ймовірності можна вирішити як звичайний метод найменших квадратів для моделі лінійної регресії максимізує вірогідність, коли припускається, що всі спостережувані результати мають нормальний розподіл із однаковою дисперсією.

Отже, згідно методу MLE, ймовірність появи деякого слова w_n може бути визначена шляхом отримання кількості появ uni-gram $C(w_n)$ та їхньої нормалізації зі кількістю всіх uni-gram. Тоді як поява слова w_n після w_{n-1} може бути знайдена методом кількості появ bi-gram $C(w_{n-1}, w_n)$ та їх нормалізації згідно кількості bi-gram, що і зображено на рис. 5

$$P(w_n) = \frac{C(w_n)}{\sum C(w)},$$

$$\downarrow$$

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}, w_n)}{\sum C(w_{n-1}, w)},$$

$$\downarrow$$

$$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}, w_n)}{C(w_{n-N+1}^{n-1})}.$$

Довільна N-грама

Рисунок 2.2 - Формула обрахунку ймовірності появи довільної N-грами

Таким чином, використовуючи для створення N-грам текстові набори чи корпуси, статистично репрезентативні для моделювання деякої мови, можна отримати досить вірні результати ймовірності про можливість появи N - грам під час використання даного метода оцінки MLE.

2.2.2 Використання розподілу літер між блоками

Слід зазначити, для отримання бажаного результату, а саме прогнозування слова, користувач повинен ввести всі літери цього слова в правильній послідовності. Якщо, ж користувач зробить помилку, наприклад, під час введення літер, замість потрібної літери, натисне на сусідню літеру, то бажане слово, не потрапить в список пропозиції, оскільки не буде відповідати її вимогам. В результаті роботи прискореного методу отримаємо хибний результат, який не задовільнить користувача.

Отож, пропонується модифікація прискореного методу введення тексту на основі N-грам. В даній модифікації пропонується використовувати для кожної літери свій власний блок літер, згрупованих відповідно до їхнього розташування на клавіатурі. Більшість сучасних пристроїв з яких можна реалізовувати введення тексту, оснащені клавіатурою QWERTY-клавіатура[13], кожна літера має свою унікальну клавішу на клавіатурі. Сучасні мобільні пристрої оснащені сенсорним дисплеєм, на сенсорному дисплеї для зручності введення текстового повідомлення розміщена віртуальна клавіатура, взаємодія з нею відбувається легким дотиком до області екрану з потрібною літерою. Тому, модифікація прискореного метода введення текстового повідомлення та огляд його роботи буде проводитись саме на такій клавіатурі.

Застосування для кожної літери, свого блоку літер, згрупованих відповідно до свого розташування на клавіатурі, покращить якість прогнозування метода прискореного введення тексту на основі N-грам. З великою ймовірністю, що після модифікації даного метода, навіть при помилковому введені літери, метод буде працювати краще в сторону збільшення ймовірності прогнозування правильного слова.

Тому, після модифікації метода прискореного введення текстового повідомлення, під час введення літери буде реалізовуватися пошук, не по одній літері, а по блоку літер.

Розподіл літер між блоками

Зрозуміло, що при використанні методу прискореного введення тексту на основі N-грам вірний пошук слів в словнику відбувається при умові коли послідовність літер, що вводилась, відповідають послідовності літер потрібного слова в словнику. Тобто, для отримання запропонованого слова, потрібний точний збіг послідовності введених літер, з послідовністю літер бажаного слова. Якщо, користувач під час введення текстового повідомлення натисне сусідню клавішу, необхідне йому слово, не буде запропоноване в списку пропозицій. Тому, що не буде відповідати введеній користувачем послідовності літер і в результаті роботи метода отримає хибний результат, або взагалі його відсутність. Для того запропонована модифікація метода прискореного введення тексту на основі N-грам, що полягає у створенні для кожної літери, свого блока літер, відповідно до місця розташування на клавіатурі.

Зрештою, найбільш поширеною клавіатурою, яка встановлюється на сучасних смартфонах та решті сучасних пристроях, є QWERTY-клавіатура, де кожна кнопка відповідає певній літері. На мобільних пристроях із сенсорним екраном така клавіатура розміщується безпосередньо на екрані пристрою і натискання клавіш на ній відбувається шляхом натискання у відповідній області екрану, проте все частіше виникають технічні помилки. Причина полягає в тому, що при натисканні на екран розмір пальця більший ніж розмір бажаної літери на сенсорному дисплеї і користувач натискає не ту літеру, що йому потрібна. Такий варіант розподілу літер між блоками буде досить зручно використовувати для українського варіанту розкладки QWERTY-клавіатури

Оскільки метод предиктивного введення тексту не враховує ці технічні помилки то і не може розпізнати слово. Тому, користувач не отримує бажаного результату.

Для початку потрібно розділимо клавіатуру на блоки літер, так щоб кожна літера утворювали власний блок з літер, які розташовані праворуч, знизу, ліворуч та зверху. Таким чином в кожному блоці буде від трьох до семи літер.

Приклад створення блоку, утвореного з трьох літер, наведено на рис. 2.3 та семи літер на рис. 2.4.

Рисунок 2.3 – Блок з трьох літер, для українського варіанту QWERTY-клавіатури

Рисунок 2.4 – Блок з семи літер, для українського варіанту QWERTY-клавіатури

В цьому випадку блоки перетинаються між собою, а кожна літера може знаходитись в декількох сусідні блоках одночасно. Це сприятиме тому, що при помилковому натисканні літер, що розташовані ліворуч або праворуч, вище або

нижче від бажаних, потрібна літера з великою ймовірністю потрапить до списку пропозиції, який буде розглядатись при реалізації прогнозування слів.

Так, наприклад, у блоці, що утворює літера «В», розташовуватиметься літера «І», що розташована ліворуч від літери «В», літера «А», що розташована праворуч від літери «В», літери «У» та «К», що розташовані вище від літери «В», літери «Ч» та «С», що розташовані нижче від літери «В», а також безпосередньо літера «В». Приклад створення блоку, що утворює літера «В», наведено на рис. 2.5

Рисунок 2.5 – Приклад створення блоку літери «В», для українського варіанту QWERTY-клавіатури

Аналогічно блоки створюються для інших літер. Оскільки кожна літера створює свій окремий блок, то кількість блоків буде рівна кількості літер на клавіатурі. Проте кількість літер у блоці буде не завжди рівна, так до прикладу блок літери «Я» буде мати всього три сусідні літери, праворуч «Ч» та зверху «Ф», «І».

Для прогнозування дуже важливо створити статистичну модель мови на основі тематичних текстів, що розширить перелік можливих слів-кандидатів та дасть можливість підвищити швидкість набору текстових повідомлень на мобільних пристроях. Зазвичай використовують N-грами декількох видів. Послідовність з одного елемента називають Uni-Gram, послідовність з двох елементів

Bi-Gram та відповідно з трьох Tri-Gram. Принцип формування такого корпусу слів на основі текстів наведено на рис. 2,6:

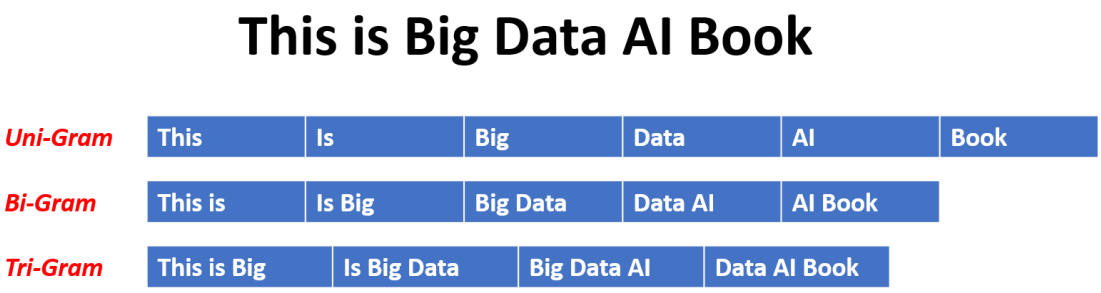


Рисунок 2.6 – Приклад розбивки тексту

Після розбиття Uni-Gram зберігаються у реляційній БД для зручності використання. В базі даних елементи Uni-Gram будуть мати вигляд таблиці з двома стовпчиками, де в першому назва слова, а другий це кількість раз його використання в тексті для навчання наведено на рис. 2,7.

word	count
а	621
або	283
абсцес	1
аварію	2
аварія	3
авто	8
автобус	16
автобуси	7

Рисунок 2.7 – Uni-Gram в базі даних

Отже, модифікований метод прискореного введення тексту на основі N-грам можна показати наступним чином на рис.2.8

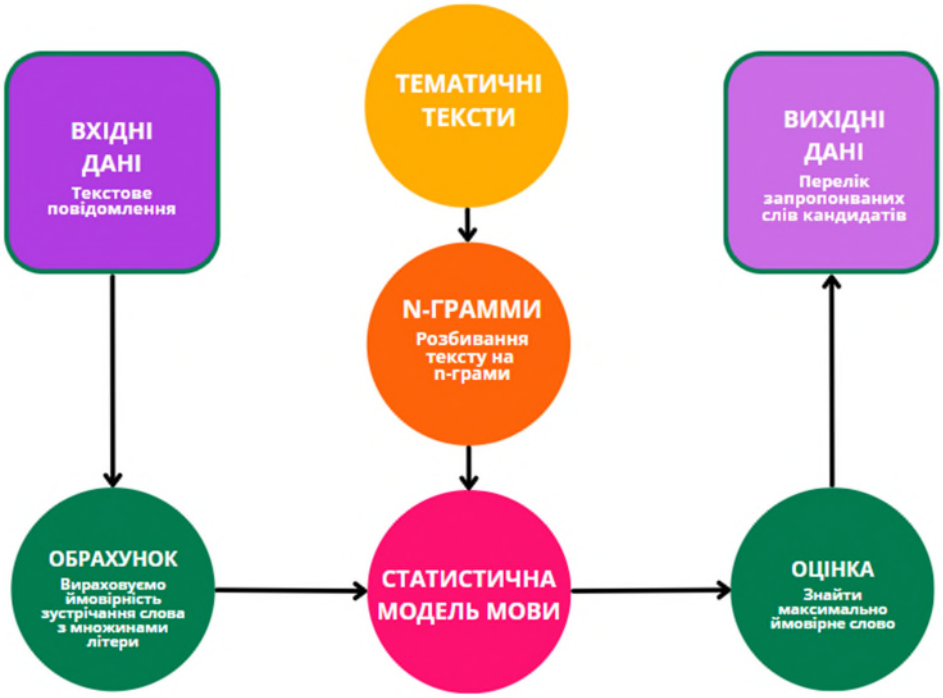


Рисунок 2.8 – Загальний приклад схеми роботи методу

В результаті введення текстового повідомлення, формується деяка послідовність літер, кожна літера має свій індивідуальний блок літер. До блоку літер які натиснули, входить літера яка відповідає натиснутій клавіші на клавіатурі, а також літери-сусіди, що розташовуються ліворуч, праворуч, вище та нижче. Після цього серед слів статистичної моделі мови відбувається пошук, що відповідає заданій послідовності блоків літер.

Користувачеві відображаються перші слова, які мають максимальну оцінку ймовірності.

Висновки до розділу 2

В результаті роботи було вдосконалено метод предиктивного введення тексту з використанням статистичної моделі мови, що дало можливість

прискорити введення текстових повідомлень. Відмінність від відомих методів полягає в тому, що при передбаченні можливих варіантів слів враховуються "літери-сусіди", які мають високу ймовірність помилкового натискання при наборі тексту на мобільних пристроях.

В подальшому планується реалізувати дану інформаційну технологію та провести експериментальні дослідження.

Розділ 3
Розробка методів та компонентів для інтелектуальної системи передбачення слів при введені текстового повідомлення

3.1 Вимоги до розроблених програмних засобів

При розробці та тестуванні модифікованої інтелектуальної системи прискореного введення текстового повідомлення на основі N-грам необхідним є розробка та створення програмних засобів, що забезпечуватимуть можливість введення текстового повідомлення користувачем та відображення користувачеві списку пропозицій. Для взаємодії користувача із програмними засобами є розробка застосунку за допомогою Windows Forms[14].

Отже, необхідною є розробка та створення програмних засобів, що забезпечуватимуть наступну функціональність:

1) створення статистичної моделі мови за допомогою N-грам слів;

2) створення інтерфейсу користувача у вигляді Windows Forms та забезпечення можливості введення тексту;

3) створення списку перелік запропонованих слів кандидатів під час введення тексту.

3.2 Вибір мови програмування

На сьогодні є безліч мов програмування, тому для вирішення різних проблем можна використати програми написані на різних мовах. Однак для кожної задачі є мови які найкраще підходять. Для реалізації програмних засобів, що дадуть можливість створити статистичну модель мови на основі N-грам з використанням деякого корпусу слів, дозволятимуть користувачеві вводити текстове повідомлення та пропонуватимуть під час введення бажане

слово, доцільно використовувати просту та зручну мову програмування, що дозволить швидко створити бажані компоненти програмних засобів та не потребує написання великого об’єму коду для їхньої реалізації. Для можливості протестувати роботу програмних засобів за допомогою введення тексту і створенням списку пропозиції для введення в реальному часі є доцільним застосування мови програмування, що надаватиме можливість здійснювати всі потрібні операції за прийнятний час.

3.2.1 Мова програмування C++

C++ - це універсальна мова високого рівня, що надає перевагу програмування програм низького рівня драйвери, ядра і навіть більш високого рівня ігор, графічного інтерфейсу користувача, настільних програм[15]. Основний синтаксис і структура коду C і C++ однакові.

C++ - створена відомим комп’ютерним науковцем Бйорне Страустропом у рамках еволюції сімейства мов C. Він був розроблений як крос-платформне вдосконалення C, щоб забезпечити розробникам вищий ступінь контролю над пам’яттю та системними ресурсами.

Дехто називає C++ «C з класами», тому що він вводить принципи об’єктно-орієнтованого програмування, включаючи використання визначених класів, у структуру мови програмування C. Згодом C++ залишився дуже корисною мовою не лише в самому комп’ютерному програмуванні, але й у навчанні нових програмістів принципам роботи об’єктно-орієнтованого програмування. Однак він підтримує не тільки об'єктно-орієнтований, але також процедурний і функціональний. Завдяки високій гнучкості та масштабованості C++ можна використовувати для розробки широкого

діапазону програмного забезпечення, програм, браузерів, графічних інтерфейсів користувача, операційних систем та ігор.

Сьогодні C++ все ще дуже цінується за його помітну портативність, яка дозволяє розробникам створювати програми, які можуть працювати на різних операційних системах або платформах дуже легко. Незважаючи на те, що C++ є мовою високого рівня, оскільки C++ все ще близький до C, його можна використовувати для низькорівневих маніпуляцій завдяки тісному зв’язку з машинною мовою.

Б’ярн Страуструп розробив C++ у Bell Labs на початку 1980-х років, щоб об’єднати найкращі переваги кількох інших мов. Він хотів поєднати швидкість BCPL, високий рівень Simula та універсальність C Денніса Річі. Він також черпав натхнення з інших мов, таких як Ada, ML та ALGOL 68, щоб створити добре структуровану, мова загального призначення, яка могла б компілювати майже всі програми на Сі без зміни їх вихідного коду.

Коли C++ був новим, об'єктно-орієнтоване програмування тільки виходило на сцену. Цей революційний тип комп’ютерного програмування змінив світ кодування, обіцяючи більш складні віртуальні типи даних і об’єкти.

В об’єктно-орієнтованому програмуванні об’єкт - це тип даних, який має як дані, так і функції, властиві його дизайну. До появи об’єктно-орієнтованого програмування програмісти зазвичай бачили кодову базу, що складається з окремих інструкцій командного рядка. Ідентифікація об’єктів із вбудованими даними та функціями призвела до нового способу упаковки та автоматизації роботи з кодом.

Для чудового прикладу об’єктно-орієнтованого програмування на C++ однією з найбільш помітних і корисних функцій мови був стек C++.

Стек C++ - це клас у C++, який має такі характеристики - це віртуальний послідовний контейнер для зберігання «останній у першому», який має визначений набір елементів. Функції «push» і «pop» або виштовхують новий елемент у нижню частину стека, або висувають перший доступний елемент із верхньої частини стека.

Програмісти використовували стек C++ різними способами для досягнення цілей, пов’язаних із оцінкою змінних і функціональними операціями в кодовій базі.

Мова також застосовує принципи інкапсуляції, яка визначає моделі використання, і успадкування, коли один клас може успадкувати певні атрибути або властивості від іншого.

Недоліки C++

C++ поділяє деякі важкі для розуміння концепції, характерні для C. Зокрема, вказівники є складною концепцією для розуміння, і їх неправильне використання може призвести до збоїв системи та ненормального споживання пам’яті. Відсутність збирача сміття також ускладнює фільтрацію непотрібних даних. Іншим обмеженням C++ є наявність проблем із безпекою, пов’язаних із доступністю покажчиків, глобальних змінних і функцій друзів[16].

3.2.2 Мова програмування C#

C# - об’єктно-орієнтована, компонентно-орієнтована мова програмування. C# надає мовні конструкції для безпосередньої підтримки цих концепцій, що робить C# природною мовою для створення та використання програмних компонентів[17]. З моменту свого створення C# додав функції для підтримки нових робочих навантажень і новітніх практик проектування

програмного забезпечення. За своєю суттю C# є об’єктно-орієнтованою мовою. Що дає змогу користувачу визначати типи та їх поведінку.

Кілька функцій C# допомагають створювати надійні та довговічні програми. Збирання сміття «Garbage collection» автоматично звільняє пам’ять, зайняту недоступними невикористаними об’єктами. Типи, що допускають значення Nullable, захищають від змінних, які не посилаються на виділені об’єкти. Обробка виключень забезпечує структурований і розширюваний підхід до виявлення та відновлення помилок. Лямбда-вирази підтримують методи функціонального програмування. Синтаксис Language Integrated Query (LINQ) створює загальний шаблон для роботи з даними з будь-якого джерела. Підтримка мови для асинхронних операцій забезпечує синтаксис для побудови розподілених систем. C# має уніфіковану систему типів. Усі типи C#, включаючи примітивні типи, такі як inti double, успадковуються від одного кореневого object типу. Усі типи мають спільний набір спільних операцій. Цінності будь-якого типу можна зберігати, транспортувати та працювати з ними узгоджено. Крім того, C# підтримує як визначені користувачем типи посилань, так і типи значень. C# дозволяє під час робити розміщувати об’єкти та вбудоване зберігання легких структур. C# підтримує загальні методи та типи, що забезпечує підвищену безпеку типів і продуктивність. C# надає ітератори, які дозволяють розробникам класів колекцій визначати користувальницьку поведінку для клієнтського коду.

C# наголошує на версії, щоб програми та бібліотеки могли розвиватися з часом сумісним чином. Аспекти дизайну C#, на які безпосередньо вплинули міркування щодо керування версіями, включають окремі модифікатори virtual та override правила вирішення перевантаження методів і підтримку явних декларацій членів інтерфейсу.

Програми C# виконуються на .NET, віртуальній системі виконання, що називається загальномовним середовищем виконання CLR, і наборі бібліотек класів. CLR - це реалізація Microsoft спільної мовної інфраструктури CLI, міжнародного стандарту. CLI є основою для створення середовищ виконання та розробки, у яких мови та бібліотеки безперебійно працюють разом.

Вихідний код, написаний мовою C#, компілюється в проміжну мову IL, яка відповідає специфікації CLI. Код IL і ресурси, такі як растрові зображення та рядки, зберігаються в збірці, зазвичай із розширенням .dll. Збірка містить маніфест, який надає інформацію про типи, версію та культуру збірки.

Коли програма C# виконується, збірка завантажується в CLR. CLR виконує компіляцію Just-In-Time (JIT) для перетворення коду IL у власні машинні інструкції. CLR надає інші служби, пов’язані з автоматичним збиранням сміття, обробкою винятків і керуванням ресурсами. Код, який виконує CLR, іноді називають «керованим кодом». «Некерований код» компілюється в рідну машинну мову, націлену на певну платформу.

Сумісність мов є ключовою особливістю .NET. Код IL, створений компілятором C#, відповідає специфікації загального типу CTS. Код IL, згенерований з C#, може взаємодіяти з кодом, який було згенеровано з версій .NET F#, Visual Basic, C++. Існує понад двадцяти інших CTS-сумісних мов. Одна збірка може містити декілька модулів, написаних різними мовами .NET. Типи можуть посилатися один на одного, як якщо б вони були написані однією мовою.

На додаток до служб часу виконання, .NET також містить великі бібліотеки. Ці бібліотеки підтримують багато різних робочих навантажень. Вони організовані в простори імен, які надають широкий спектр корисних функцій. Бібліотеки включають в себе: від введення та виведення файлів до маніпулювання рядками та синтаксичного аналізу XML, інфраструктури веб-

додатків до елементів керування Windows Forms. Типова програма на C# широко використовує бібліотеку класів .NET для виконання типових робіт для розрахунків.

3.2.3 Мова програмування Java

Java - це широко використовувана об’єктно-орієнтована мова програмування та програмна платформа, яка працює на мільярдах пристроїв, включаючи ноутбуки, мобільні пристрої, ігрові консолі, медичні пристрої та багато інших[18]. Правила і синтаксис Java базуються на мовах C і C++. Тому Java є швидкою, безпечною та надійною.

Однією з основних переваг розробки програмного забезпечення за допомогою Java є його переносимість. Після того як ви написали код для програми на Java на ноутбуці, його дуже легко перемістити на мобільний пристрій. Коли в 1991 році Джеймсом Гослінгом із Sun Microsystems (пізніше придбаної Oracle) цю мову винайшов, головною метою була можливість «написати один раз і запустити будь-де».

Також важливо розуміти, що Java значно відрізняється від JavaScript. Javascript не потрібно компілювати, тоді як код Java потрібно компілювати. Крім того, Javascript працює лише у веб-браузерах, тоді як Java можна запускати будь-де.

Нові та вдосконалені інструменти розробки програмного забезпечення надходять на ринок із надзвичайною швидкістю, витісняючи існуючі продукти, які раніше вважалися незамінними. У світлі цього постійного обороту довговічність Java вражає; Більш ніж через два десятиліття після свого створення Java все ще залишається найпопулярнішою мовою для розробки програмного забезпечення — розробники продовжують обирати її замість

таких мов, як Python, Ruby, PHP, Swift, C++ та інших. Як результат, Java залишається важливою вимогою для конкуренції на ринку праці.

Java - це технологія, що складається як з мови програмування, так і з програмної платформи. Щоб створити програму за допомогою Java, вам потрібно завантажити Java Development Kit JDK, який доступний для Windows, macOS і Linux. Ви пишете програму на мові програмування Java, потім компілятор перетворює програму на байт-код Java - набір інструкцій для віртуальної машини Java JVM, яка є частиною середовища виконання Java JRE. Байт-код Java працює без змін у будь-якій системі, яка підтримує JVM, що дозволяє виконувати ваш код Java будь-де.

Програмна платформа Java складається з JVM, Java API і повного середовища розробки. JVM аналізує та запускає байт-код Java. Java API складається з великого набору бібліотек, включаючи базові об’єкти, функції мережі та безпеки. Генерація розширюваної мови розмітки XML і веб-сервіси. Разом мова Java і програмна платформа Java створюють потужну, перевірену технологію для розробки корпоративного програмного забезпечення.

3.2.4 Мова програмування Python

Python - це популярна мова програмування загального призначення, яку можна використовувати для різноманітних програм. Він включає високорівневі структури даних, динамічний тип, динамічне зв’язування та багато інших функцій, які роблять його таким же корисним для розробки складних програм, як і для сценаріїв або який з’єднує компоненти разом. Його також можна розширити, щоб здійснювати системні виклики майже до всіх операційних

систем і виконувати код, написаний на C або C++. Завдяки повсюдному поширенню та здатності працювати майже на будь-якій системній архітектурі, Python є універсальною мовою, яку можна знайти в різних програмах.

Мова програмування включає тисячі сторонніх модулів, доступних в індексі пакетів Python. PyPI надає популярні стандарти для різного досвіду, наприклад Django для веб-розробки та NumPy, Pandas і Mathplotlib для обробки даних .

Вперше розроблений наприкінці 1980-х Гвідо ван Россумом, Python просунувся як мова програмування з відкритим вихідним кодом, керуючи громадськими обговореннями через пропозиції вдосконалення Python. У 2018 році ван Россум залишив посаду доброзичливого довічного диктатора мови BDFL і, як офіційно зазначено в PEP 13 , було створено керівну раду, яка буде керувати мовою.

Python Software Foundation - це некомерційна корпорація, яка володіє правами інтелектуальної власності на мову програмування Python. Це включає Python версії 2.1 і пізніших, PyPI, еталонну реалізацію CPython та інфраструктуру для підтримки мови. PSF також надає гранти на розробку програмного забезпечення та проводить кілька конференцій PyCon на рік.

Зазвичай очікується, що програми на Python працюватимуть повільніше, ніж програми на Java, але їх розробка потребує набагато менше часу. Програми на Python зазвичай у 3-5 разів коротші за еквівалентні програми на Java. Цю різницю можна пояснити вбудованими високорівневими типами даних Python і його динамічною типізацією. Наприклад, програміст на Python не витрачає час на оголошення типів аргументів або змінних, а потужні поліморфні типи списків і словників Python, для яких розширена синтаксична підтримка вбудована прямо в мову, знаходять застосування майже в кожній програмі на Python. Через типізацію під час виконання, час виконання Python має

працювати більше, ніж Java. Наприклад, коли обчислюється вираз a + b, він повинен спочатку перевірити об’єкти a і b, щоб дізнатися їхній тип, який невідомий під час компіляції. Потім він викликає відповідну операцію додавання, яка може бути перевантаженим методом, визначеним користувачем. Java, з іншого боку, може виконувати ефективне додавання цілих чисел або з плаваючою комою, але вимагає оголошення змінних для a і b і не дозволяє перевантажувати оператор + для екземплярів визначених користувачем класів.

Наразі Python має третю основну версію та регулярно оновлюється.

Існує кілька причин, чому Python є хорошим вибором як мови програмування, залежно від вашої точки зору та досвіду.

3.2.6 Порівняння мов програмування

Кожна з наведених мов програмування підходить добре для виконання своєї задачі.

Мова C++ з появою перших трансляторів знайшла відразу ж дуже широке розповсюдження, на ній було створено величезну кількість програм і застосунків. В процесі створення великих програмних систем знайшли недоліки, які сприяли пошуку альтернативних рішень. Таким альтернативним рішенням стала мова Java, яка в деяких галузях стала конкурувати у популярності з C++, а фірма Майкрософт запропонувала мову C# як нову мову, що розвиває принципи C++ і що використовує переваги мови Java.

Головною відмінністю між двома мовами програмування C# або Java полягає в їх передбачуваному використанні. Java в основному призначений для розробки мобільних додатків, точніше - Android, а C# фокусується на веб-розробці, розробці застосунків та ігор . Варто згадати, що ці дві мови насправді

більш схожі, ніж різні, тому що, обидві можуть бути використані для веб-розробки, але також мають своє власне цільове призначення.

Проте якщо порівнювати мови програмування Java, C# та Python, зрозуміло, що код на Python буде займати в три-п’ять разів менше коду, тому що має величезний набір стандартних бібліотек, що допомагають швидко та акуратно програмувати.

Отже, розробка програмних засобів на мові програмування C# є комфортнішою, програмний код написаний завдяки мові програмування C#, легко читається та вважається зручнішим для написання. Перевагою буде той факт, що C# потребує менше обсягу пам’яті для написання коду ніж Java. Сама розробка інтерфейсу користувача проходить в рази швидше ніж на інших мова програмування, оскільки тут використовуються стандартні засоби розробки .NET WindowsForms. Розроблений програмний засіб, буде надавати можливість користувачу вводити текстове повідомлення та отримувати, список пропозицій ймовірного слова.

3.3. Організація програмних засобів

Програмні засоби розроблено на мові програмування C# версії 7.0 з використанням середовища розробки Windows Forms .NET Framework 4 на операційній системі Windows[20]. Під час створення програмних засобів, сторонні бібліотеки не використовувались, всі процеси та операції в програмі виконуються лише на стандартних бібліотеках мови програмування C#.

Текстові поля WindowsForms використовуються для отримання вхідних даних від користувача або для відображення введеного тексту, але для перегляду результату роботи використовується елемент dataGridView, який дає змогу отримувати та переглядати дані з статистично моделі мови. TextBox та

dataGridView взяті з стандартного набору елементів WindowsForms. dataGridView може відображати декілька рядків, регулювати розмір тексту за розміром елемента керування та додавати основне форматування. Це все дає змогу користувачу працювати з стандартною клавіатурою при роботі на операційній системі Windows.

Слід зазначити, що елементи Windows Forms використовуються лише для взаємодії з користувачем, відповідно робота інших розроблених програмних засобів є незалежною від операційної системи. Для забезпечення коректної роботи розроблених засобів між собою, необхідно вдосконалювати модуль взаємодії з користувачем. Інші компоненти, до яких, в тому числі, відносяться модулі, що реалізовують: створення списку пропозицій для введення на основі слів, що вводить користувач, що є основою розроблених програмних засобів, таких покращень не потребують і можуть працювати також і на інших операційних системах.

Всі програмні засоби, що були розроблені, складаються з наступних модулів:

- presage-0.9.1 - модуль генерування N-грамів слів та підрахунок їх частоти повторювань на основі деякого тематичного тексту;

- _1_gram.sql - модуль створення статистичної моделі мови на основі N-грам;

- sqlConnection.cs - модуль підключення до статистичної моделі мови;

- Adapter.cs - модуль створення списку пропозиції;

- search_letter.cs - модуль, який відповідає за пошук літер-сусідів;

- Form1.cs - модуль інтерфейс взаємодії із користувачем.

Також розроблені програмні засоби мають допоміжні текстові файли:

- t9database.sql - файл, у якому записана статистична модель мови;

- block_letter.txt - файл, де міститься інформація про літери та блоки в яких вони знаходяться;

Загальну структуру розроблених програмних засобів наведено на рис.3.1.

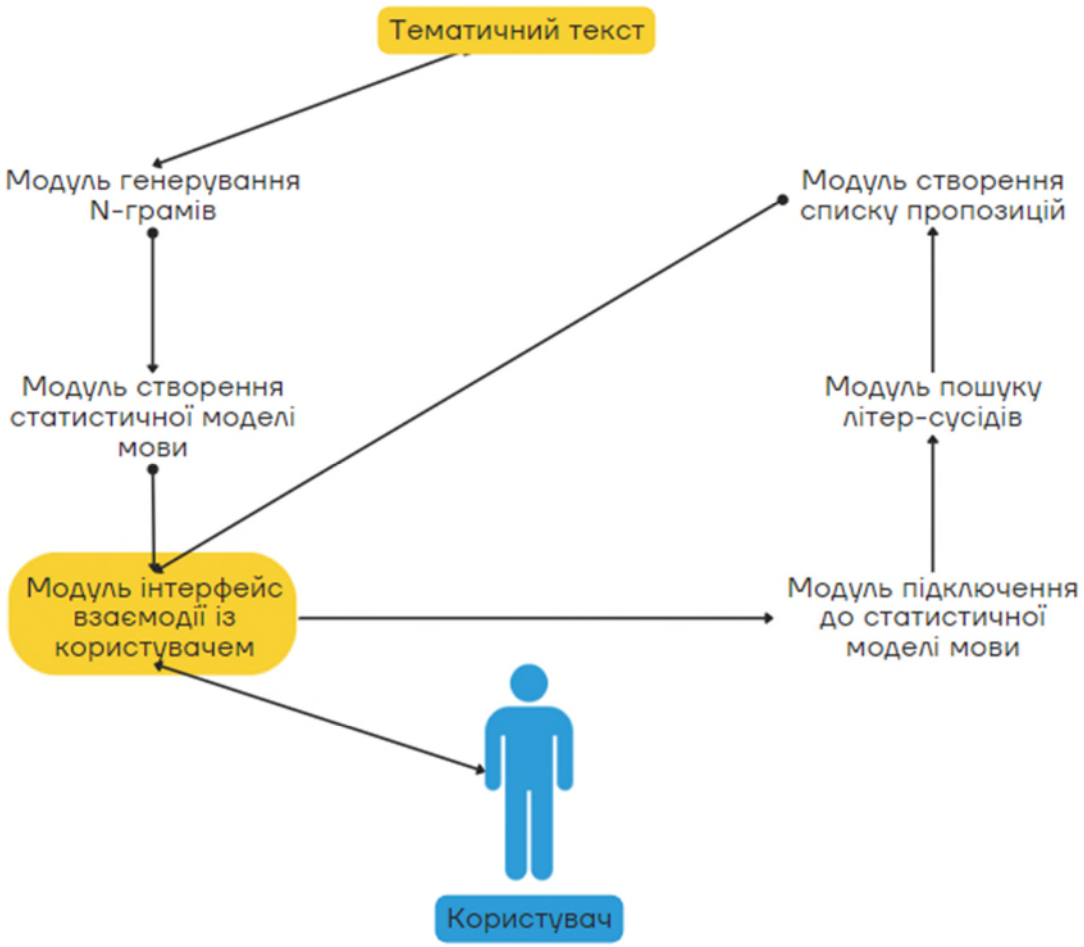


Рисунок 3.1 – Загальна структура розроблених програмних засобів

3.3.1 Модуль генерування N-грам слів та підрахунок їх частоти

Модуль генерування N-грам слів та підрахунку їх частоти повторювань на основі тематичних текстів отримує на вхід заданий текст та на його основі генерує uni-gram, bi-gram та tri-gram слів та підраховую кількість їх повторювань в даному тексті. Після чого передає дані у модуль зчитувань та

відповідно записує в допоміжні файли, SQL-запити, відповідно до частоти повторювань uni-gram (_1_gram.sql), bi-gram (_2_gram.sql), та tri-gram (_3_gram.sql).

Алгоритм роботи модуля генерування N-грам слів та підрахунок їх частоти повторювань на основі тематичних текстів виглядає наступним чином:

1. Отримати на вхід шлях по якому розташований файл з тематичними текстами.

2. Отримати з тематичного тексту набір прихованих повідомлень шляхом зчитування даних, що розміщуються в даній директорії та розбиття по відповідним файлам.

3. З отриманого набору прихованих повідомлень генеруємо список Uni-gram та підраховується кількість повторювань

4. На основі отриманих Uni-gram формується списки Bi-gram та Tri-gram з підрахунком кількості повторювань

5. Здійснюється виклик відповідного методу який записує отримані Uni-gram, Bi-gram та Tri-gram у відповідні файли

При здійсненні генерування uni-gram, bi-gram та tri-gram слів на основі тематичного тексту, відбудеться створення статистичної моделі мови, оскільки всі слова з тексту будуть мати число, їх повторювань в даному тексті. Зазвичай в тексті більшість слів починатися з малої літери, тому у файл записуються слова написані з малої літери, а кількість їх повторювань в тексті, що будуть отримані з цього ж слова, лише написаного з великої літери, будуть додаватися до кількості повторювань слова написаного з малої літери. Це реалізовано так по одній причині, ці слова є змістовно однакові, а написані з маленької чи великої літери, значення не має. Проте параметри прогнозування могли б знизитись, якби записували ці два слова окремо, оскільки б в списку пропозицій вони стояли поруч.

3.3.2 Модуль створення статистичної моделі мови

Модуль створення статистичної моделі мови на основі попередньої обробки тематичних текстів, отримує на вхід файл-запит « _1_gram.sql ». Після запуску якого в базі даних MySQL створюється реляційна таблиця з словами та кількістю їх повторювань, так звана «статистична модель мови». Модуль використовує стандартні функції створення та додавання елементів в таблицю[2]. Для створення таблиці використовується функція *CREATE TABLE* в якій вказуємо назву таблиці та назву рядків які хочемо додати. Для додавання інформації в рядки таблиці використовуємо також стандартну функцію *INSERT INTO* в якій вказуємо назву створеної таблиці та рядки в які хочем додати інформацію. Отримавши статистичну модель мови на основі N-грам маємо можливість підключитись до неї та використовувати всі необіхдні дані.

Запит на створення та заповнення БД наведено на рис. 3.2

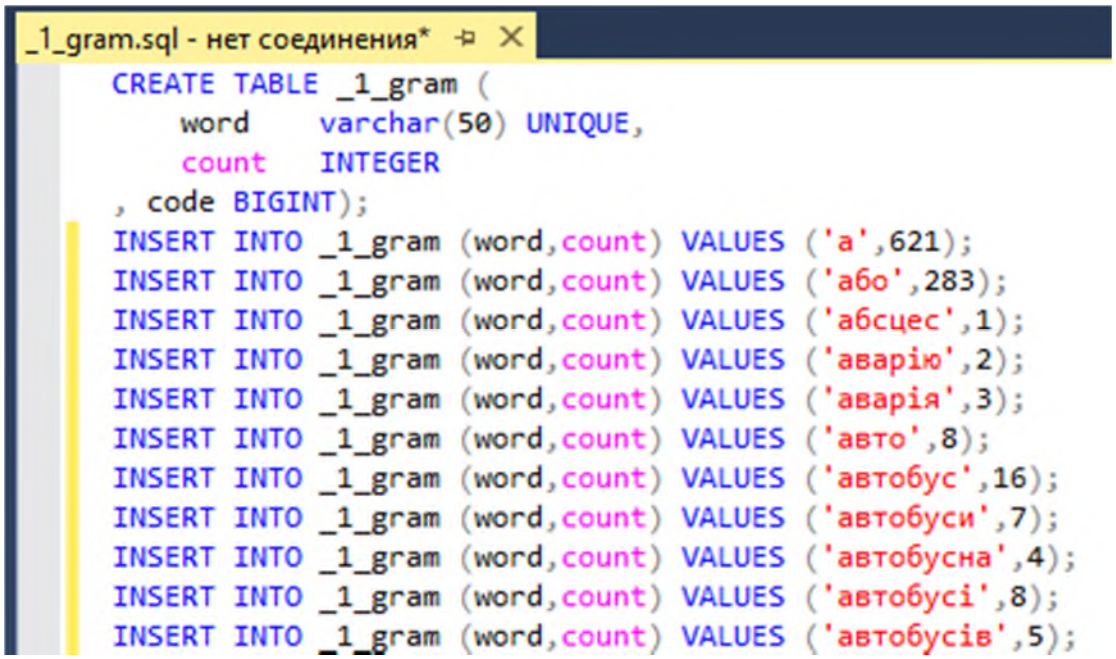


Рисунок 3.2 – Приклад запит на створення

3.3.3 Модуль підключення до статистичної моделі мови

Модуль підключення до статистичної моделі мови складається з невеличкого проте дуже зручного метода «SqlConnection». Об'єкт «SqlConnection». надає унікальний сеанс для джерела даних SQL Server. У системі баз даних клієнта чи сервера, це прирівнюється мережному підключенню до сервера. SqlConnection використовується разом з SqlDataAdapter і SqlCommand для підвищення продуктивності при підключенні до бази даних Microsoft SQL Server. На вході отримуємо шлях до бази даних SQL, назву самої БД та пароль. Після підключення, маємо зв'язок між застосунком та статистичною моделю мови.

Метод підключення до статистичної моделі мови наведено на рис. 3.3.

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace t9pred
{
    Ссылок: 0
    internal class _1_gram_add
    {
        Ссылок: 0
        class Program
        {
            Ссылок: 0
            static void Main(string[] args)
            {
                string connectionString = @"Data Source=DESKTOP-JRN4DSR\TEW_SQLEXPRESS;Initial Catalog=t9database;Integrated Security=True";
                string sqlExpression = "INSERT INTO _1_gram (word, count) VALUES ('a', 621)";
                using (SqlConnection connection = new SqlConnection(connectionString))
                {
                    connection.Open();
                    SqlCommand command = new SqlCommand(sqlExpression, connection);
                    int number = command.ExecuteNonQuery();
                    Console.WriteLine("Добавлено объектов: {0}", number);
                }
                Console.Read();
            }
        }
}
```

Рисунок 3.3 – Загальний вигляд підключення до статистичної моделі мови

3.3.4 Модуль створення списку пропозицій

Модуль створення списку пропозицій на вході отримує текстове повідомлення, яке вводить користувач, після чого запускається метод перебору кожної літери. Оскільки кожна літера має свій блок-літер в якому може знаходитись від мінімальних трьох, до максимальних семи літер, а тобто літер-сусідів які знаходяться ліворуч, праворуч, зверху та знизу. Отримуємо масив літер по яким буде здійснюватися пошук. Комбінація введених літер створює масив літер по яким буде робитись пошук, послідовність літер яка найбільше повторюється і буде бажаним результатом.

Модуль створення списку пропозицій наведено на рис. 3.4.

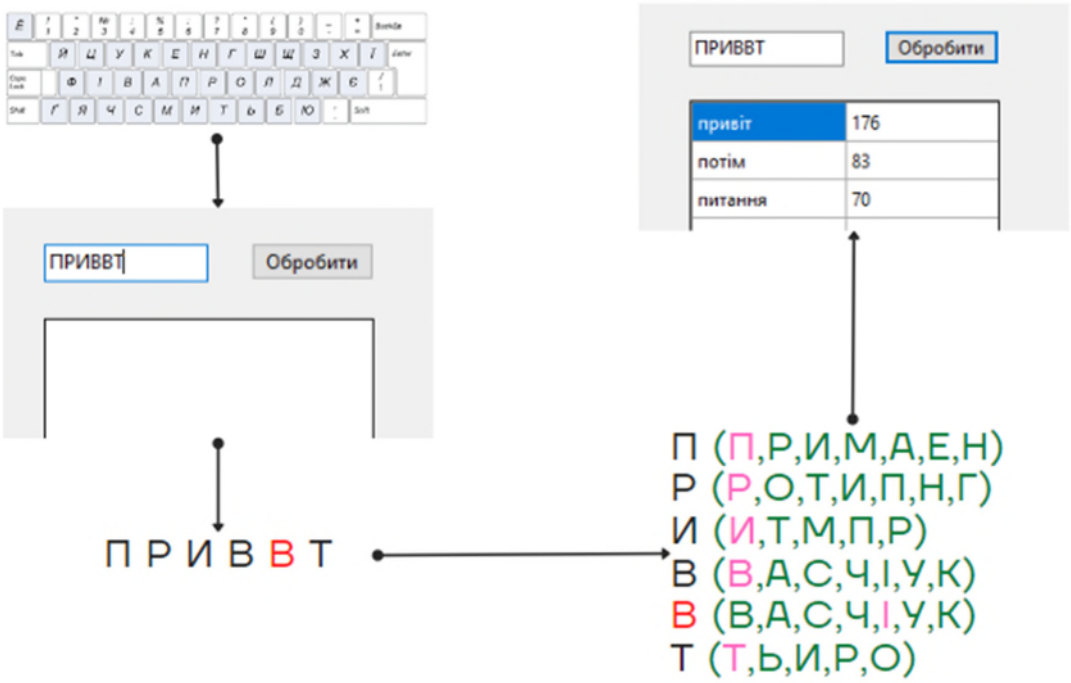


Рисунок 3.4 – Загальний вигляд модуль списку пропозицій

Реалізація методу генерування списку пропозицій для введення на основі uni-gram, введення продовження слова, або прогнозування бажаного не було розпочато - модуль передбачає такі дії:

1. Послідовність елементів, поданих на вхід методу, оброблюється таким чином, щоб врахувати регістр кожної літери.

2. Серед N-грам слів чи словосполучень, отриманих під час введення тексту користувачем, відбувається пошук N-грам слів таких, першими елементами яких слова чи словосполучення, що містяться в обробленому списку елементів. Після чого отриманий набір N-грам слів сортується і його відсортовані елементи додаються до списку пропозицій для введення.

3. Якщо, отриманий список пропозицій для введення перевищує максимальну кількість елементів списку пропозицій для введення, до кінцевого списку пропозицій для введення додаються лише перші елементи отриманого

списку. Схема розрахунку ймовірності бажаного слова при введені текстового повідомлення, зображено на рис. 3.5

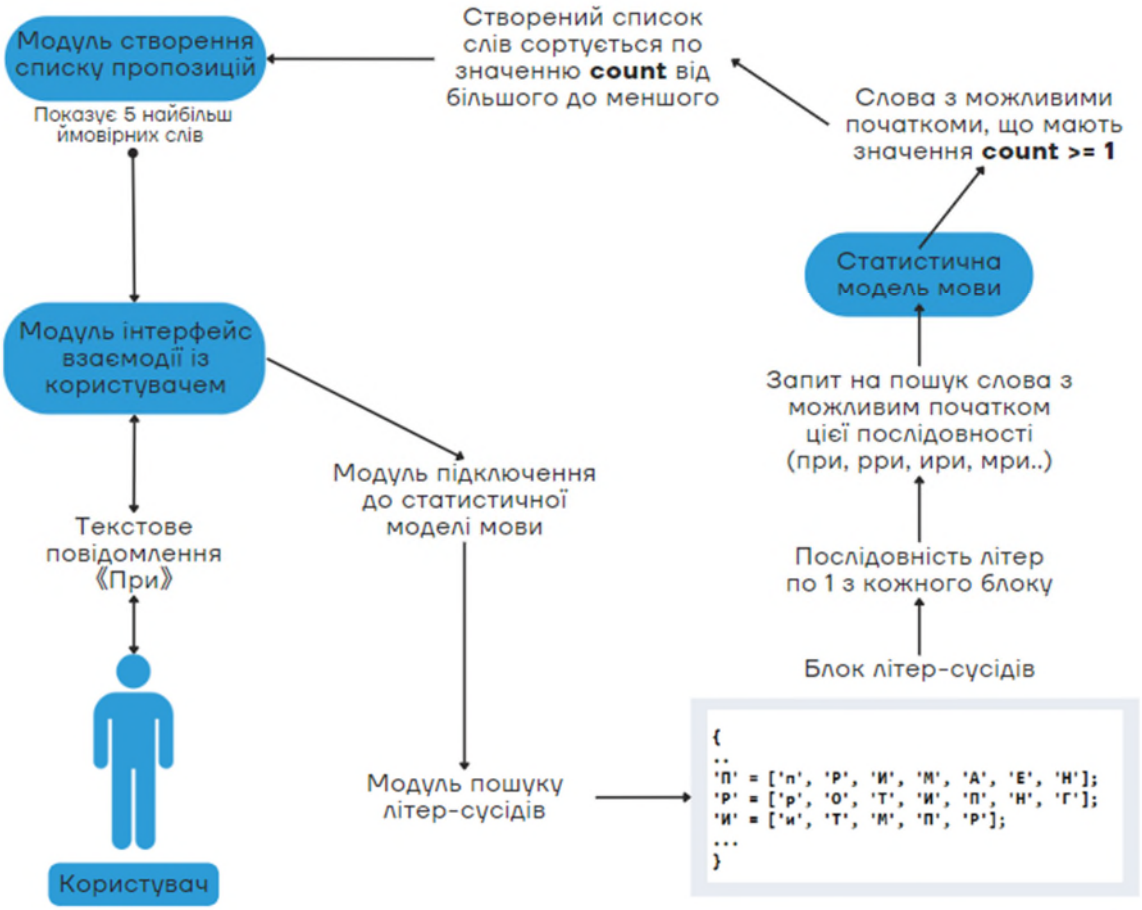


Рисунок 3.5 – Покрокова схема розрахунку вибору того чи іншого слова для прогнозування.

Розглянемо, випадок коли користувач залишить поле для введення тексту пустим, тоді метод запропоную найбільш ймовірне слово для початку розмови, даний випадок зображений на рис.3.6.

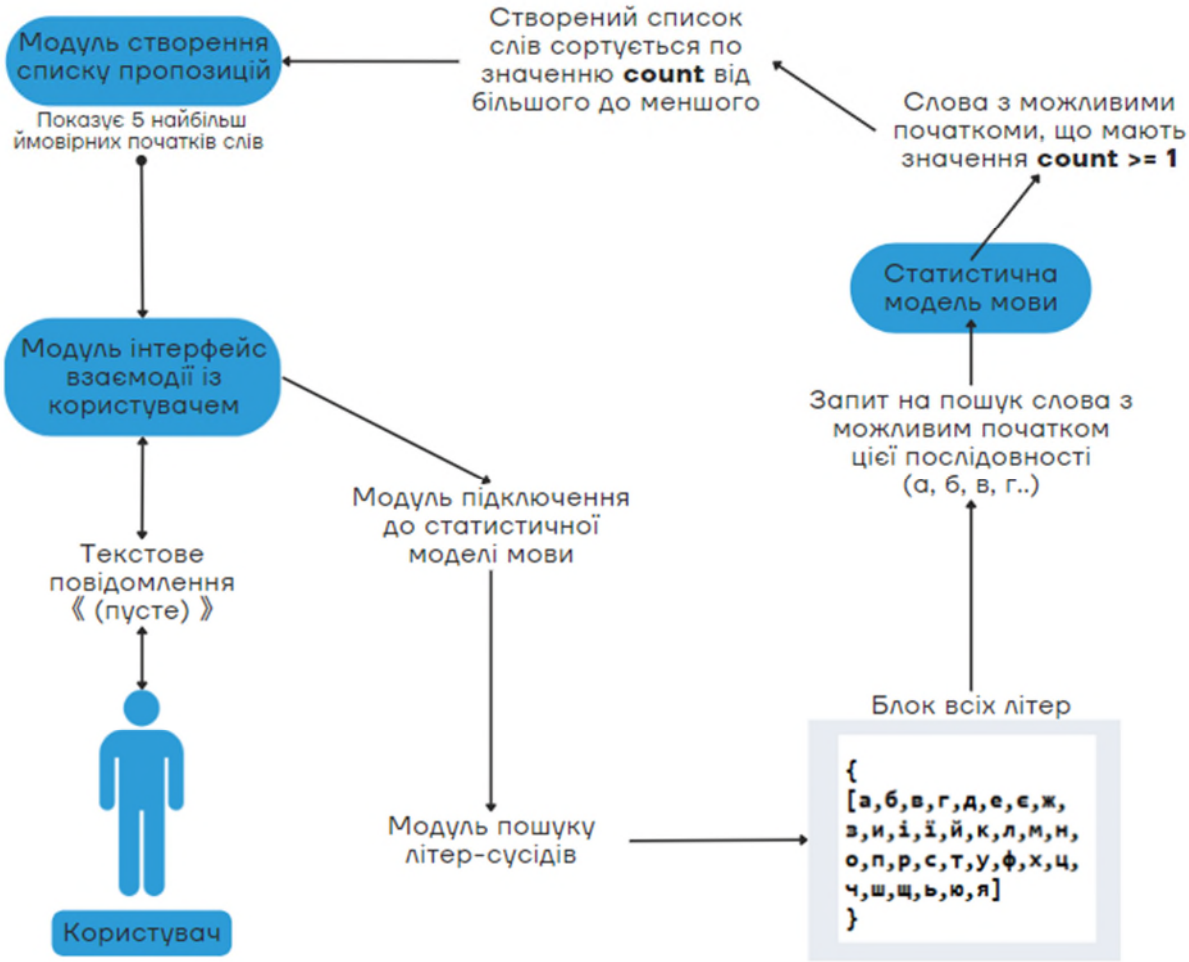


Рисунок 3.6 – Покрокова схема розрахунку вибору слова, коли поле для введення тексту залишилось порожнім.

3.3.5 Модуль який відповідає за пошук літер сусідів

Модуль який відповідає за пошук літер сусідів є частиною великого процесу. Потрібно розуміти, що кожна літера має свій блок-літер, таких блоків буде рівно стільки як і самих літер на клавіатурі. Завчасно для кожної літери створено свій блок, тобто при помилковому натисканні шанси не зменшуються до нуля, а будуть в діапазоні 100 : S = R, де S - кількість літер сусідів, R –

відсоток, що метод підбере бажану літеру. Мінімальне значення буде при S = 7, а максимальне значення буде при S = 3.

Модуль створення списку пропозицій наведено на рис. 3.6.

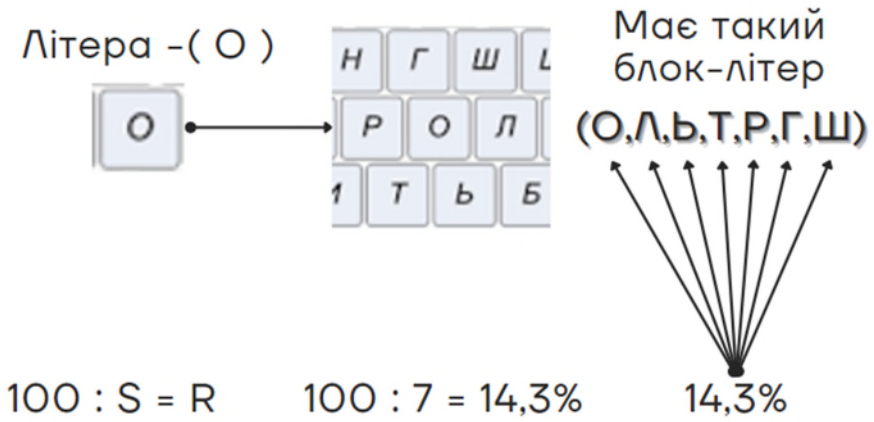


Рисунок 3.6 – Загальний вигляд модуль списку пропозицій

Реалізація модуля який відповідає за пошук літер-сусідів або прогнозування бажаної літери - передбачає такі дії:

1. Метод отримує на вхід деяку послідовність літер, після чого цей набір потрапляє в цикл, де розбивається на одиночні елементи.

2. Кожний елемент має в собі лише одну літеру, по якій здійснюється запит в базу даних. На виході маємо масив літер-сусідів, а саме для букви «А», сусідами будуть літери зверху та знизу, праворуч та ліворуч відповідно.

3. «А» перетвориться на масив літер «А, П, М, С, В, К, Е», який далі буде застосовано в прогнозуванні бажаного слова.

4. Відмінність між блоками-літер, лише одна, чим менше літер-сусідів в блоці, тим більший шанс, швидше знайти бажану літеру.

3.3.6 Модуль інтерфейс взаємодії із користувачем

Модуль інтерфейсу взаємодії із користувачем дає можливість користувачу вводити текстове повідомлення та переглядати можливі передбаченні слова для введення, відсортовані від найбільш ймовірного до менш ймовірного.

Інтерфейс користувача складається з головної панелі застосунка на якій розміщений блок-textbox - що призначений для введення текстового повідомлення, блок-gridbox - відповідає за відображення списку пропозицій, прогнозованого слова та блок-клавіша « Обробка » яка відповідає за запуск функції обробки метода. Також до інтерфейсу користувача відноситься фізична клавіатура пристрою на якому відбувається введення текстового повідомлення. При натисканні клавіші « Esc » відбувається вихід з програми.

Модуль інтерфейсу користувача наведено на рис. 3.7

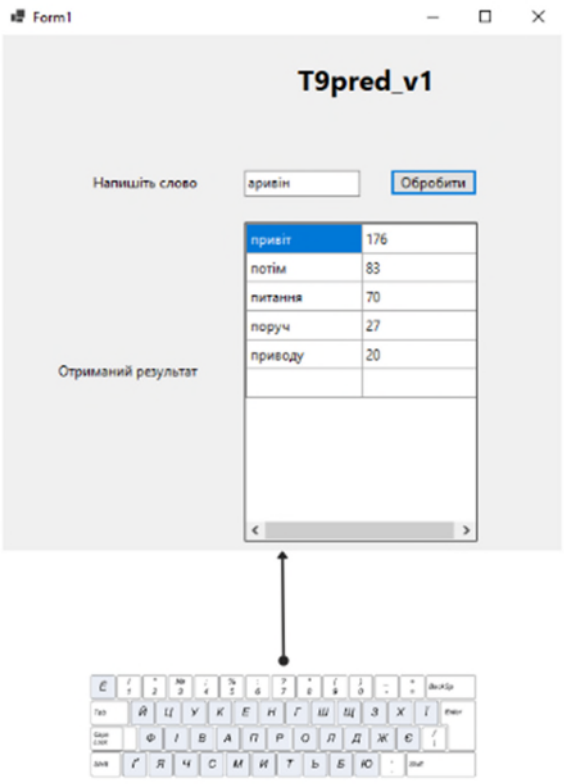


Рисунок 3.7 – Загальний вигляд інтерфейсу користувача

Під час натисканні клавіш-літер на клавіатурі, відбувається введення літери відповідно до нажатої клавіші, послідовність літер створює текстове повідомлення. Після чого обробляється методом прискореного введення тексту. При редагуванні слова в текстовому полі, пропозиція також змінюється.

3.4 Опис структури даних

Для збереження тематичного тексту використовується файл з розширенням «.docx» – формат документу, який використовує Microsoft Word, є ефективним та створює менш пошкоджені файли, його функціоналу вистачає вдосталь.

Для зберігання даних використовуються реляційні бази даних створені SQL запитом. Такий спосіб зберігання даних є дуже зручним, оскільки в пріоритеті є швидкість, надійність та зручність використання. Стандартні функції дозволяють легко сортувати, додавати, видаляти, редагувати текст.

Тому, була створена статистична модель мови на основі MySQL.

Приклад зберігання даних в базі даних MySQL зображено на рис 3.8.

Рисунок 3.8 – Приклад зберігання даних в базі даних MySQL

Збереження даних про літери та блоки до яких вони відносяться відбувається також з допомогою файлів розширення «.docx». Такий формат зберігання даних, дає можливість легко замінити файл з блоками літер на іншу мову, після чого застосунок почне працювати на цій мові, проте схожих слів не знайде, оскільки комплексно потрібно замінити тематичні тексти, що призведе до змін в базі даних.

Але не лише одними текстовими файлами можна користуватись для зберігання блоків-літер, їх можна записати в самому коді, що є менш зручно. При цьому приріст швидкості роботи метода прискореного введення тексту буде не значним.

Приклад зберігання блоків-літер зображено на рис. 3.9.

```
{
    'Й' = ['й', 'Ц', 'І', 'Ф'] ;
    'Ц' = ['ц', 'У', 'В', 'І', 'Ф', 'Й'];
    'У' = ['у', 'К', 'А', 'В', 'І', 'Ц'];
    'К' = ['к', 'Е', 'П', 'А', 'В', 'У'];
    'Е' = ['е', 'Н', 'Р', 'П', 'А', 'К'];
    'Н' = ['н', 'Г', 'О', 'Р', 'П', 'Е'];
    'Г' = ['г', 'Ш', 'О', 'Р', 'Н'];
    'Ш' = ['ш', 'Щ', 'Л', 'О', 'Г'];
    'Щ' = ['щ', 'З', 'Д', 'Л', 'Ш'];
```

Рисунок 3.9 – Приклад зберігання блоків-літер

Висновки до розділу 3

На основі аналізу мов програмування, їхніх перевагах та недоліках, було обрано найбільш зручну та легку мову програмування C#, яка ідеально підходить для розробки та вдосконалення даного метода прискореного введення текст на основі N-грам. Застосувавши візуальний конструктор Windows Forms, розроблено програмний застосунок, описано всі елементи застосунка та їхній функціонал.

Система для зрозумілості розбита на модулі, деякі з них працюють самостійно, решта між собою, це все дає зручність використання та вдосконалення існуючих модулів для оптимізації їх роботи. Зв'язок модулів між собою є дуже важливим, оскільки без роботи одного модуля, інші будуть працювати некоректно.

Розділ 4

Аналіз отриманих результатів

4.1. Характеристика тестових наборів даних

4.1.1. Характеристика текстових корпусів

Для тестування запропонованої модифікації методу прискореного введення тексту на основі *N*-грам за допомогою розробленого програмного забезпечення було використано дані текстового корпусу української мови, на основі якого згенеровано початковий список *N*-грам та здійснений підрахунок їх частот. Використаний текстовий корпус описаний в табл.1.

Таблиця 1 . Містить опис текстових корпусів, що були використані для тестування запропонованої модифікації методу предиктивного введення тексту на основі N-грам

№	Назва	Опис
1	Генеральний регіонально анотований корпус української мови	Генеральний регіонально анотований корпус української мови - це дуже велика, репрезентативна, структурована колекція текстів на українській мові яка дозволяє будувати на базі корпусу власні підкорпуси, шукати слова, граматичні форми та їх сполучення, а також обробляти результати пошуку, сортувати, робити збалансовані вибірки і

		одержувати різну статистичну інформацію.
2	Корпус слів розмовної української мови на основі текстів діалогів на побутові теми.	Корпус слів розмовної української мови на основі текстів діалогів на побутові теми, що використовуються в розмовних-словниках. Такі діалоги моделюють бесіди між людьми, які взаємодіють наживо, охоплюють поширені побутові ситуації та використовують набір слів і фраз. Розмір словника 15 000 слів.

Корпус набір текстів або текстових уривків, зібраних для використання як зразка мови чи мовного різновиду. Він складається з текстів, створених у «природному контексті» опубліковані книги, звичайна розмова, листи, газети, лекції, що означає, що він відображає природну мову. Добре скомпонований корпус можна використовувати для відповіді на запитання про використання мови.

Корпус можуть бути використані для дослідження або порівняння різних мовних різновидів, таких як мова з певної області, яка охоплює певний жанр або тип тексту, створений певними користувачами мови тощо.

Корпуси можуть бути синхронними, що охоплюють один час або діахронними, охоплюють кілька періодів часу, складатися з різних носіїв письмової чи усної мови і складатися з різних мов.

Корпус не є ідеалом для нормативної української мови, в ньому можуть трапитись слова та сполучення, які не відповідають сучасним стандартам літературної мови.

4.1.2. Характеристика текстів для введення

Тестування запропонованої модифікації методу прискореного введення текстового повідомлення на основі N-грам за допомогою розробленого програмного забезпечення було використано тематичний текст.

В першому тексті міститься переважно загальна лексику, що застосовується при написанні статей та публікацій, а другий має в собі переважно розмовну лексику, що використовується у повсякденному спілкуванні. Зміст цих текстів наведено у табл. 2.

Для кожного з цих текстів, було створено два тексти, при цьому замінили деякі літери на сусідні, що має зімітувати помилкові натискання на клавіатурі. Перший містить 10%, а другий 20% літер, що були замінені на сусідні.

Щоб умови тестування були однакові, кожний раз очищаються дані з статистичної моделі мови, про кількість повторювань слів.

Таблиця 2 Зміст текстів для введення, застосованих для тестування запропонованої модифікації методу прискореного введення текстового повідомлення на основі N-грам слів

№	Лексика	Зміст

1	Загальна	Метод прискореного введення текстового повідомлення дає змогу набирати цілі речення лише кількома натисканнями. Під час набору тексту пропонуються найбільш ймовірні варіанти слів чи фраз, а кожна наступна набрана літера буде тільки покращувати точність прогнозування. Це в свою чергу скоротить час набору тексту та покращити якість написання слів.
2	Розмовна	Привіт, Сашко! Радий тебе бачити! Привіт! Ти як ? Як відпочив у бабусі? Дуже класно! Я і тато їздили в Карпати. Там так чудово! Я попробував кататися на лижах! Супер! Не страшно було, там ж багато людей? Спочатку страшнувато. Гірка така довга, хоч і не дуже гостра. Мені допомагав тато, тому в мене все дуже швидко виходило, мені сподобалось!

4.2. Порівняння отриманих результатів

Що б дослідити ефективність запропонованої модифікації методу прискореного введення текстового повідомлення на основі N-грам у порівнянні з стандартним методом прискореного введення тексту на основі N-грам було проведене тестування розробленого програмного забезпечення з використанням зазначених вище тематичних текстів.

Тестування програмного забезпечення з використанням модифікованого методу прискореного введення тексту здійснювалося з використанням даних про розподіл літер між блоками, а стандартного методу, без застосування цих даних.

Ефективність методів потрібно перевіряти з використання критерія KSPC. Обидва методи в тесті мають використовувати однакові текстові корпуси та перевіряти на одних і тих самих текстах для введення, а також підрахунку значення KSPC для кожного тестового випадку.

4.2.1. Результат для корпусу №1

Результати, отримані під час проведеного тестування з використанням текстового корпусу №1, зображено у табл. 3.

Таблиця 3

Результати, отримані під час проведеного тестування з використанням текстового корпусу №1

Відсоток замінених літер	Лексика тексту для введення	Значення KSPC для тексту	
		Стандартний метод	Модифікований метод
10%	Загальна	0,927	0,852
	Розмовна	0,721	0,589
20%	Загальна	1,124	0,846
	Розмовна	0,887	0,651

На рис. 4.1 зображена діаграму порівняння результатів роботи стандартного та модифікованого методу, отриманих з використанням

текстового корпусу №1 для тексту із загальною лексикою та з розмовною на рис. 4.2.

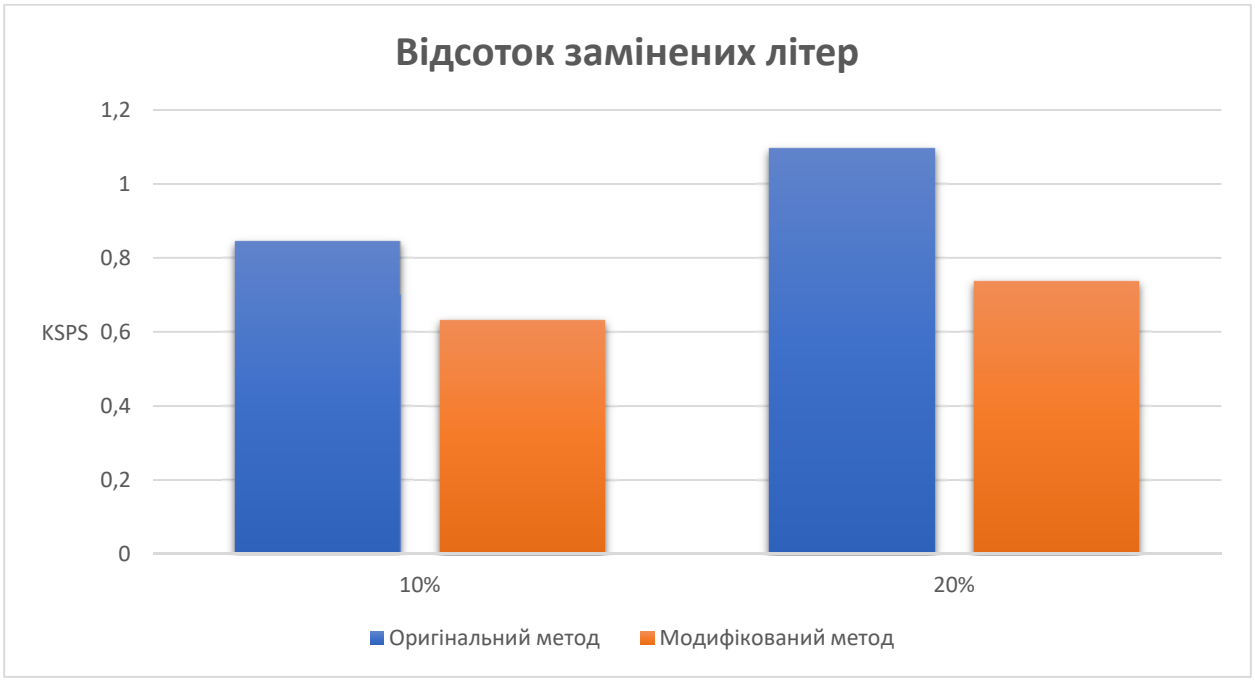


Рисунок 4.1 – Порівняння результатів роботи для текстового корпусу №1 і тексту із загальною лексикою

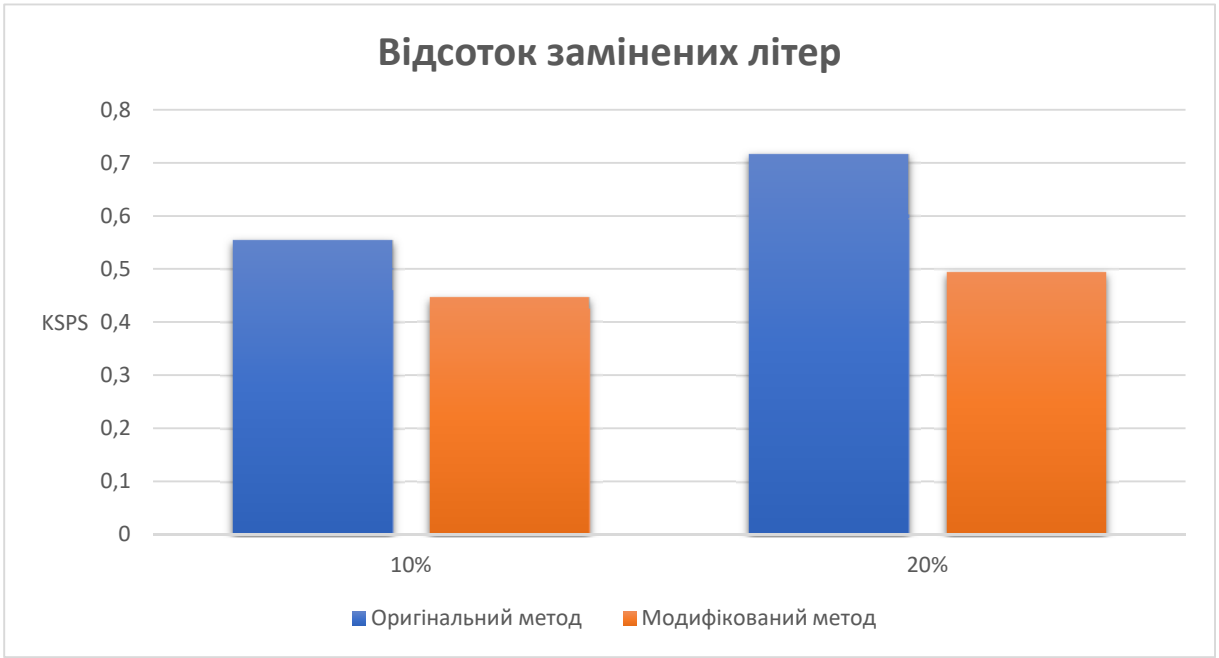


Рисунок 4.2 – Порівняння результатів роботи для текстового корпусу №1 і тексту із розмовною лексикою

4.2.2. Результат для корпусу №2

Результати, отримані під час проведеного тестування з використанням текстового корпусу №2, зображено у табл. 4.

Таблиця 4. Результати, отримані під час проведеного тестування з використанням текстового корпусу №2

Відсоток замінених літер	Лексика тексту для введення	Значення KSPC для тексту	
		Стандартний метод	Модифікований метод
10%	Загальна	0,952	0,745
	Розмовна	0,784	0,651
20%	Загальна	1,027	0,660
	Розмовна	0,964	0,679

На рис. 4.3 наведено діаграму порівняння результатів роботи стандартного та модифікованого методу, отриманих з використанням текстового корпусу №2 для тексту із загальною лексикою, а на рис. 4.4 – із розмовною.

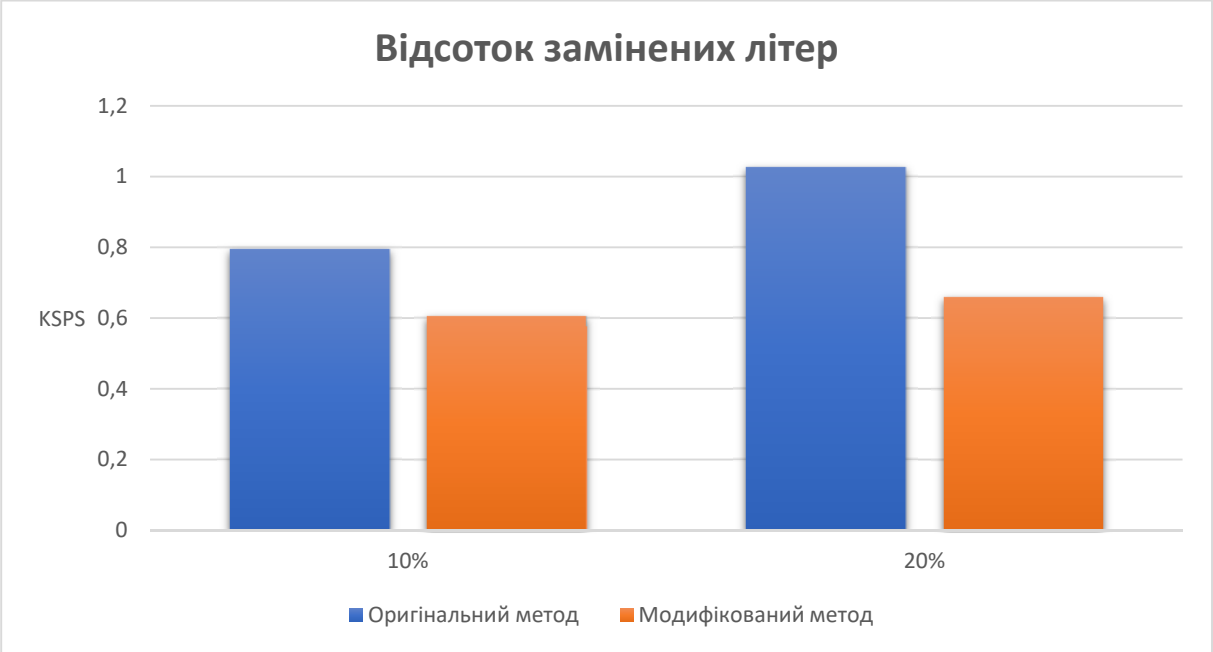


Рисунок 4.3 – Порівняння результатів роботи для текстового корпусу №2 і тексту із загальною лексикою

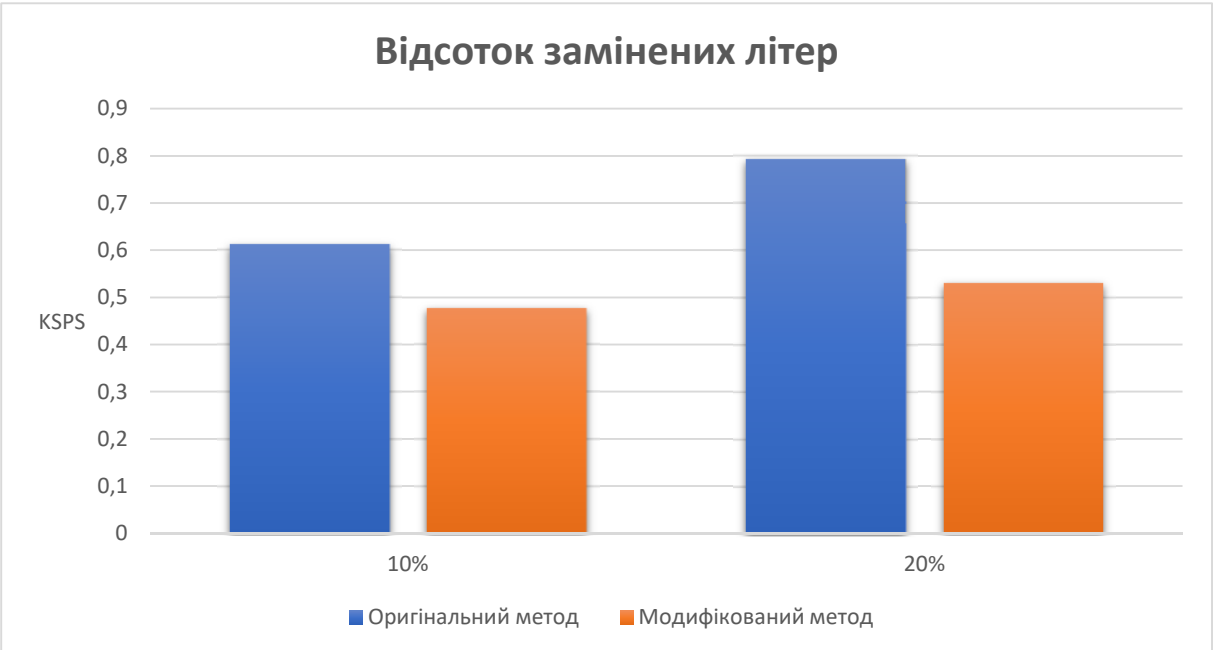


Рисунок 4.3 – Порівняння результатів роботи для текстового корпусу №2 і тексту із розмовною лексикою

4.2.3. Аналіз отриманих результатів

В результаті тестування, можна побачити, що результати двох тестових корпусів є досить різними, для стандартного корпусу значення KSPC завжди менше ніж у модифікованого. Причому чим більший відсоток змінених літер в тексті тим більша різниця в значенні KSPC стандартного та модифікованого методу прискореного введення тексту.

Під час тестування тексти, що мали 10% та 20% замінених літер, значення оцінки KSPC для модифікованого методу прискореного введення текстового повідомлення, всі текстові корпуси які використовували, загальну лексику, розмовну лексику, будуть мати цей критерій нижчий чим в стандартного метода. При чому залежність між цими методами, буде рости відповідно до збільшення відсотку замінених літер в тексті. Так при заміні 20% літер, різниця між значення KSPC на 17,34 та 19.73% відповідно.

На рис. 4.4 зображено діаграму порівняння результатів роботи стандартного та модифікованого методу в середньому значенні для всіх текстових корпусів і тексту із загальною лексикою, та розмовною на рис. 4.5.

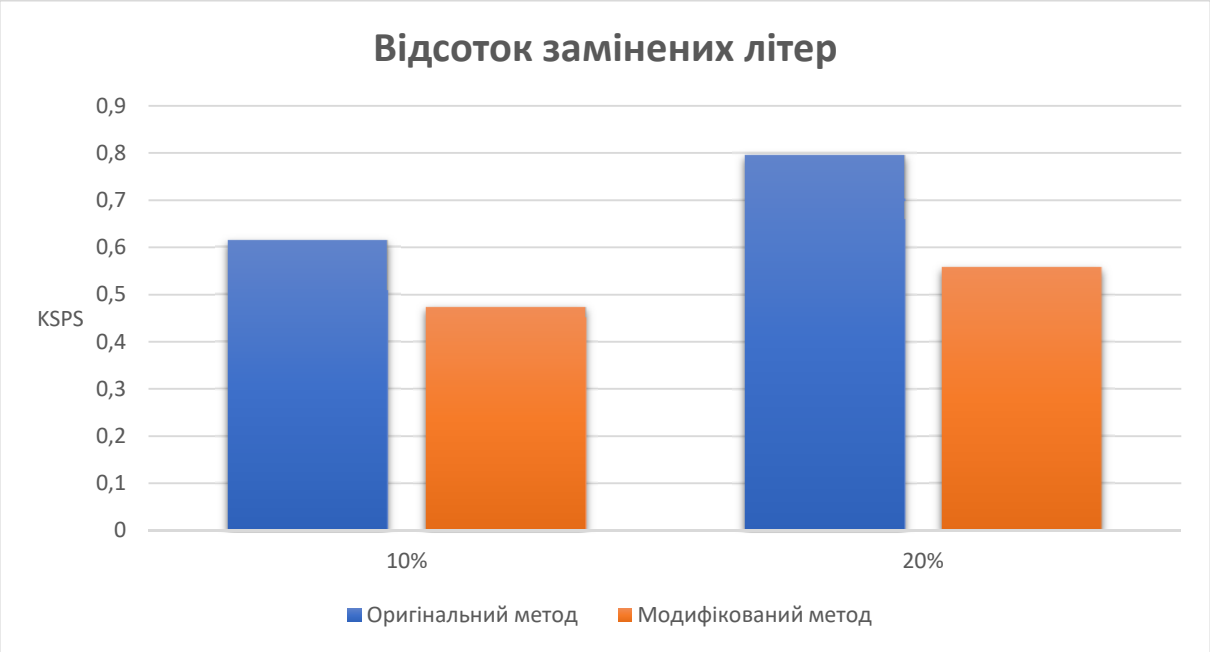


Рисунок 4.4 – Порівняння результатів роботи середнього значення для всіх текстових корпусів і тексту із загальною лексикою

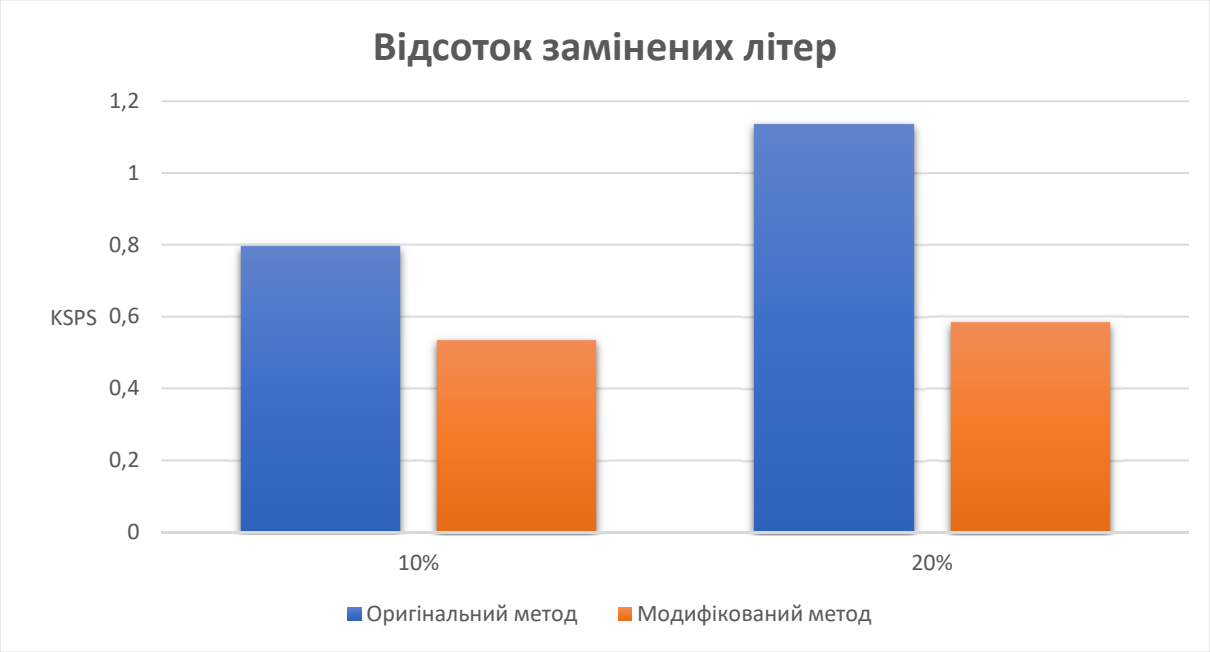


Рисунок 4.5 – Порівняння результатів роботи середнього значення для всіх текстових корпусів і тексту із розмовною лексикою

Потрібно звернути увагу, що різниця між значенням KSPC для стандартного та модифікованого методу в середньому значенні є вищою для тексту, що використовує загальну лексику. Це зв'язано з тим, що тексти з загальною лексикою середня довжина слова є більшою ніж у текстів з розмовною лексикою. З цього маємо наслідок, чим більшою є довжина слова, тим більшу кількість натискань допомагає зекономити вибір бажаного слова зі списку пропозицій на початку його введення. Тому, коли використовується модифікований метод потрібне слово досить швидко потрапляє до списку пропозицій для введення, а при використанні стандартного методу - не потрапляє, більша довжина слова зазвичай сприяє збільшенню різниці між значеннями KSPC для цих методів.

На рис. 4.6 зображена діаграму порівняння результатів роботи стандартного та модифікованого методу в середньому для всіх текстових корпусів та обох текстів для введення.

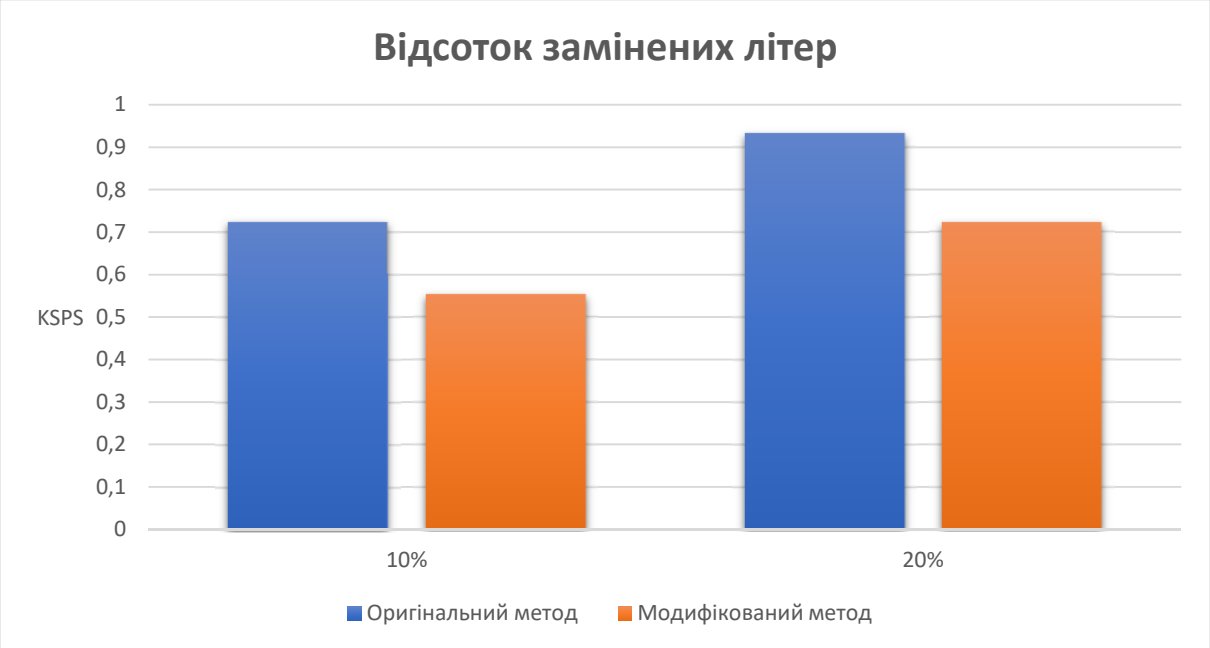


Рисунок 4.6 – Порівняння результатів роботи в середньому для всіх текстових корпусів і обох текстів для введення

Таким чином в середньому значення для всіх текстових корпусів і тексту для введення із загальною лексикою значення критерія KSPC для модифікованого методу є нижчим на 16,34%, для тексту для введення із розмовною лексикою - на 14,21%, а в середньому для обох текстів для введення на 12,93%.

4.3. Шляхи подальшого вдосконалення

Запропонована модифікація методу прискореного введення текстового повідомлення на основі N-грам враховує всі можливі помилкові натискання сусідніх літер під час введення тексту користувачем. Проте, якщо користувач помилково здійснив натискання літери декілька разів підряд, або ж взагалі помилково не натиснув на клавішу, продовживши вводити наступні літери, правильна робота методу не забезпечується тому, що введена користувачем послідовність літер не відповідатиме бажаній послідовності літер, навіть за умови врахування помилкових натискань сусідніх літер. Тому, одним з можливих шляхів подальшого вдосконалення метода є забезпечення можливості врахування випадків, коли у введеній послідовності літер, що ввів користувач, будуть відсутні деякі літери або присутні зайві.

Потрібно зазначити, що у запропонованій модифікації методу прискореного введення текстового повідомлення на основі N-грам при створенні списку пропозицій для введення, враховуються лише дані про кількість повторювань N-грам слів, отриманих в результаті оброблення тематичних текстів, текстового корпусу та в процесі введення тексту користувачем. При цьому не враховуються дані про кількість повторювань N-грам літер, що є достатньо стандартними для тієї чи іншої мови та відповідно,

не потребують додаткової взаємодії текстових корпусів чи оновлення їх в процесі введення тексту користувачем. Використання даних про кількість повторювань N-грам літер може підвищити ефективність роботи методу у випадку відсутності або недостатньої кількості даних про кількість повторювань N-грам слів через відсутність необхідних текстових корпусів для деякої мови. Тому забезпечення можливості врахування даних про кількість повторювань N-грам літер, може бути одним з можливих шляхів вдосконалення даного метода.

Висновки до розділу 4

В загальному, результати проведеного тестування довели, що при великому відсотку замінених літер оригінальний метод прискореного введення тексту на основі *N*-грам поступається перед запропонованою модифікацією, що становить близько 13%. Зрозуміло, що така вразливість є достатньо великою, одним з можливих шляхів подальшого вирішення дослідження може також бути забезпечення ефективності запропонованої модифікації на рівні, не нижчому за рівень ефективності вже модифікованого методу.

Загальні висновки

Мета даної магістерської дисертації полягає у розробці інтелектуальної системи прискореного введення текстових повідомлень за рахунок передбачення найбільш ймовірних слів в процесі набору.

Під час дослідження, проведеного в рамках даної магістерської дисертації, проаналізовано існуючі методи введення тексту на мобільних пристроях та програмні засоби, що реалізують ці методи. Також описано основні проблеми недостатньої ефективності існуючих засобів прискореного введення тексту.

Для підвищення швидкості введення тексту запропоновано модифікацію методу прискореного введення тексту на основі N-грам, що полягає у застосуванні розподілу літер між блоками-літер таким чином, що кожна літера утворює свій окремий блок з літер-сусідів. Такий варіант розміщення літер дозволяє забезпечити врахування можливих помилкових натискань сусідніх літер під час швидкого введення тексту, або помилкових натискань та збільшити швидкість його введення в умовах обмежених розмірів екранної клавіатури на сучасних пристроях.

З метою реалізації та тестування запропонованої модифікації методу прискореного введення тексту на основі N-грам розроблено програмні засоби, що дають можливість для користувача вводити текст, пропонуючи йому ймовірно бажані слова для вводу, так званий список прогнозування.

Для дослідження ефективності запропонованої модифікації у порівнянні з стандартним методом проведено тестування розроблених програмних засобів з використанням двох текстових корпусів, а також двох текстів для введення, що переважно містять здебільшого загальну та розмовну лексику. Для кожного з двох текстів додатково створено набір з двох текстів, що містять відповідно

10% та 20% замінених літер. Для тестування обох методів в умовах помилкових натискань сусідніх клавіш. У якості критерію ефективності обрано критерій KSPC, що показує кількість натискань, необхідних для введення одного символу.

Результати, отримані в ході тестування, показали, що запропонована модифікація в середньому на 16,34% ефективнішою за стандартний метод для введення прискореного тексту із загальною лексикою, та на 14,21% - для тексту із розмовною.

Можливими варіантом подальшого дослідження є забезпечення можливості врахування випадків, коли у введеній послідовності літер відсутні деякі літери або присутні зайві. Також оптимізувати роботи методу за умови відсутності помилкових чи хибних натискань сусідніх літер.

Перелік посилань

1. Метод одного натискання URL:
https://uk.upwiki.one/wiki/Predictive_text
2. Метод багатьох натискань URL:
https://en.wikipedia.org/wiki/Multi-tap
3. T9 передбачення тексту URL:
https://www.techtarget.com/whatis/definition/predictive-text#:~:text=Predictive%20text%20is%20an%20input,and%20the%20first%20letters%20typed
4. Як працює метод введення тексту WordWise URL:
https://www.yorku.ca/mack/uist01.html
5. Як застосовувати N-грами WordWise URL:
https://towardsdatascience.com/text-generation-using-n-gram-model
6. Застосування uni-gram, bi-gram, tri-gram URL:
https://en.wikipedia.org/wiki/N-gram
7. Метод предиктивного введення T9 URL:
https://www.lifewire.com/definition-of-t9-predictive-text-7
8. Метод прискореного введення iTap URL:
http://wiki-org.ua/wiki/ITap
9. Застосування QuickType на мобільних пристроях URL:
https://dvax.com/quicktype-keyboard-polnoe-rukovodstvo
10. Метод оцінки ефективності KSPC URL:
https://elearn.nubip.edu.ua/mod/book/tool/print/index.php?id=357320
11. Метод CPS URL:
https://wikipedia.org/wiki/CPS
12. Оцінка максимальної правдоподібності URL:
https://en.wikipedia.org/wiki/Maximum_likelihood_estimation

13. QWERTY keyboard and its advantages URL: https://wikipedia.org/wiki/QWERTY
14. Документація по класничному додатку Windows Forms .NET URL: https://learn.microsoft.com/dotnet/desktop/winforms/overview/?view=netdesk
15. Опис мови програмування C++ URL: https://www.w3schools.com/cpp/cpp_intro.asp
16. Мови програмування C++ URL: https://www.techopedia.com/definition/26184/c-plus-plus-programming-language
17. Основи розробці на C# URL:
18. https://programm.top/uk/c-sharp/tutorial/introduction/Переваги Java в .NET Мови програмування C++ URL:
19. Мова програмування Java URL: https://www.w3schools.com/java/default.asp
20. Мова програмування Phyton URL: https://opensource.com/resources/python
21. СКБД MySQL та стандартны функції URL: https://www.w3schools.com/sql/default.asp

ДОДАТКИ

Додаток А

Структура таблиці юніграм

word	count
а	621
або	283
абсцес	1
аварію	2
аварія	3
авто	8
автобус	16
автобуси	7
автобусна	4
автобусі	8
автобусів	5
автовідповідач	1
автовідповідача	1
автомат	2
автоматичним	1
автомобілем	1
автомобіль	8
автостанції	1
агенції	1
адміністратора	1
адреса	11
адреси	4
адресу	19

Додаток Б

Розгорнута структура класів автоматизованої системи

Лістинг Form1.cs:

```
using System;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.Xml;
using System.Data;


namespace t9pred
{
   public partial class Form1 : Form
   {

      DataBase dataBase = new DataBase();

      private SqlConnection sqlConnection = null;
      private SqlDataAdapter adapter = null;
      private DataTable table = null;

      //int selectedRow();
      public Form1()
      {
         InitializeComponent();
      }



      private void label3_Click(object sender, EventArgs e)
      {

      }

      private void textBox1_TextChanged(object sender, EventArgs e)
      {
         //string prename = textBox1.Text;
        //string i = "'" + prename + "'";
      }

      private void textBox2_TextChanged(object sender, EventArgs e)
      {
```

```
    }

    private void button1_Click(object sender, EventArgs e)
    {
        SqlConnection sqlConnection = new SqlConnection(@"Data Source=DESKTOP-JRN4DSR\TEW_SQLEXPRESS;Initial
Catalog=t9database;Integrated Security=True");
        //Підключення до БД
        sqlConnection.Open();
        SqlCommand sqlCommand = sqlConnection.CreateCommand();
        sqlCommand.CommandText = "SELECT TOP 1 pre_name, pre_repeat FROM t9database.dbo.unigramm_db WHERE pre_name LIKE 'о'
ORDER BY pre_repeat DESC";
        /sqlCommand.CommandText = "SELECT TOP 1 pre_name, pre_repeat FROM t9database.dbo.unigramm_db WHERE pre_name LIKE '" +
textBox1.Text + "%'  ORDER BY pre_repeat DESC";
        //sqlCommand.CommandText = "SELECT TOP 3 word, count FROM t9database.dbo.one_gram WHERE word LIKE '" + textBox1.Text + "%'
ORDER BY count DESC";
        SqlDataReader sqlReader = sqlCommand.ExecuteReader();
        string res = string.Empty;
        while (sqlReader.Read())
        {
            res += sqlReader["word"];
        }
        sqlReader.Close();
        SqlDataReader sqnReader = sqlCommand.ExecuteReader();
        string red = string.Empty;
        while (sqnReader.Read())
        {
            red += sqnReader["count"];
        }
        sqnReader.Close();
        sqlConnection.Close();
        textBox2.Text = (res) + "\t" + (red) + '\r' + '\n';
        textBox2.Text += Environment.NewLine + (res) + '\n' + (red) + '\n';

        sqlConnection = new SqlConnection(@"Data Source=DESKTOP-JRN4DSR\TEW_SQLEXPRESS;Initial Catalog=t9database;Integrated
Security=True");

        sqlConnection.Open();

        //adapter = new SqlDataAdapter("SELECT TOP 5 word, count FROM t9database.dbo.one_gram WHERE word LIKE '" + textBox1.Text + "%'
ORDER BY count DESC", sqlConnection);
        adapter = new SqlDataAdapter("SELECT TOP 5 word, count FROM t9database.dbo.one_gram WHERE word LIKE
'[а,б,в,г,д,е,є,ж,з,и,і,ї,й,к,л,м,н,о,п,р,с,т,у,ф,х,ц,ч,ш,щ,ь,ю,я]%'  ORDER BY count DESC", sqlConnection);
        //adapter = new SqlDataAdapter("SELECT TOP 5 word, count FROM t9database.dbo.one_gram WHERE word LIKE
'[п,р,и,м,а,е,н][р,о,т,и,п,н,г][и,т,м,п,р][в,а,с,ч,і,у,к]%'  ORDER BY count DESC", sqlConnection);
```

```
            table = new DataTable();

            table.Clear();

            adapter.Fill(table);
            dataGridView1.DataSource = table;

        }
        private void Form1_Load(object sender, EventArgs e)
        {
        }
        private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
        {
        }
        private void richTextBox1_TextChanged(object sender, EventArgs e)
        {
        }
        private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
        }
        private void label1_Click(object sender, EventArgs e)
        {

        }
    }
```

Лістинг _1_gramm_add.cs:

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace t9pred
{
    internal class _1_gram_add
    {
        class Program
        {
            static void Main(string[] args)
            {
                string connectionString = @"Data Source=DESKTOP-JRN4DSR\TEW_SQLEXPRESS;Initial Catalog=t9database;Integrated Security=True";
                string sqlExpression = "INSERT INTO _1_gram (word, count) VALUES ('a', 621)";
                using (SqlConnection connection = new SqlConnection(connectionString))
```

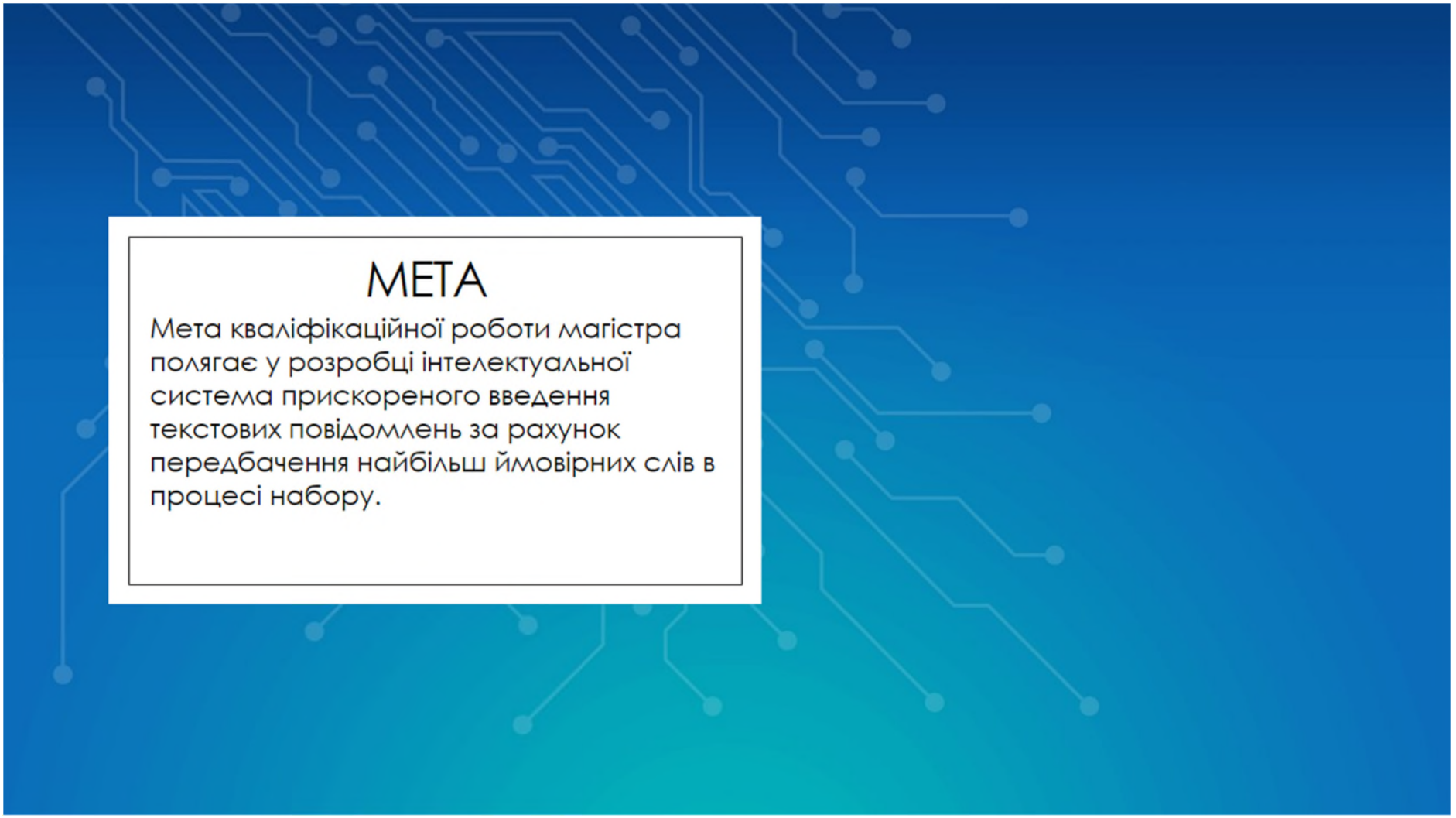
```
            {
                connection.Open();
                SqlCommand command = new SqlCommand(sqlExpression, connection);
                int number = command.ExecuteNonQuery();
                Console.WriteLine("Добавлено объектов: {0}", number);
            }
            Console.Read();
        }
    }
}
```

Додаток В

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

«Інтелектуальна система передбачення слів при введені текстових повідомлень»

Виконав студент 2 курсу групи КНм-21-1 Мороз О.В.
Керівник к.т.н, доцент кафедри КН Багрій Р.О.


МЕТА
Мета кваліфікаційної роботи магістра полягає у розробці інтелектуальної система прискореного введення текстових повідомлень за рахунок передбачення найбільш ймовірних слів в процесі набору.

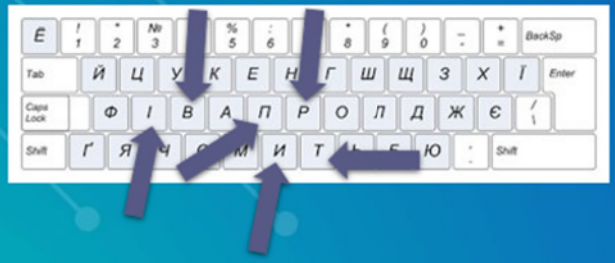
Завдання роботи

1. Провести аналіз методів передбачення слів при введені текстових повідомлень.
2. Запропонувати модель мови для оцінки ймовірності слова з урахуванням введеного тексту.
3. Розробити метод передбачення слів при введені текстових повідомлень.
4. Реалізувати інтелектуальну систему передбачення слів при введені текстових повідомлень.
5. Провести валідацію розробленого метода.

ОБ'ЄКТ ДОСЛІДЖЕННЯ

Об'єктом дослідження даної роботи є процес передбачення слів при наборі текстових повідомлень.

Привіт



Предмет дослідження

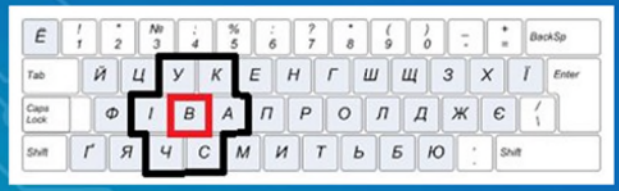
Предметом дослідження даної роботи є моделі, методи, підходи та алгоритми прискореного введення текстових повідомлень для мобільних пристроїв .

НАУКОВА НОВИЗНА

В результаті роботи були отримані наступні положення наукової новизни:

- *Вдосконалено метод предиктивного введення тексту з використанням статистичної моделі мови, що дало можливість прискорити введення текстових повідомлень. Відмінність від відомих методів полягає в тому, що при передбаченні можливих варіантів слів враховуються "літери-сусіди", які мають високу ймовірність помилкового натискання при наборі тексту на мобільних пристроях.*

Приввт



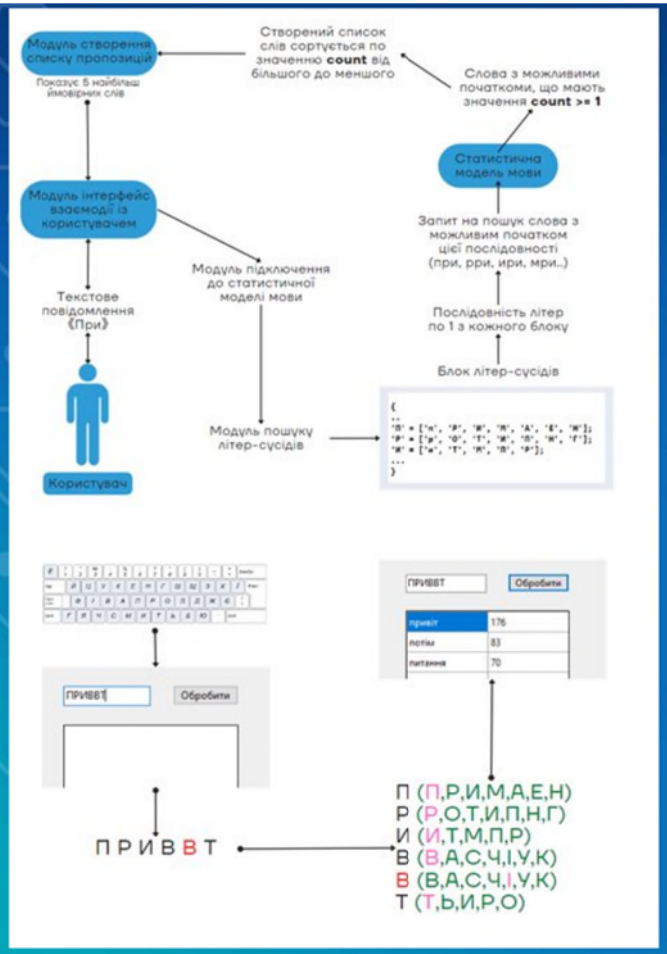
Привіт

Стаття на тему:

Метод передбачення слів при введені текстових повідомлень

Розбивання тексту на N-грамми

СТВОРЕННЯ СПИСКУ ПРОПОЗИЦІЇ ПІД ЧАС ВВЕДЕННЯ ТЕКСТУ



Міністерство освіти і науки України
Хмельницький національний університет

ЗБІРНИК НАУКОВИХ ПРАЦЬ

за матеріалами XIV Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2022»

18-19 листопада 2022

Хмельницький 2022

УДК 004.4

Мороз О.В., Багрій Р.О., Скрипник Т.К.

Хмельницький національний університет

МЕТОД ПЕРЕДБАЧЕННЯ СЛІВ ПРИ ВВЕДЕНІ ТЕКСТОВОГО ПОВІДОМЛЕННЯ

Зроблено огляд існуючих методів предиктивного введення тексту, що пропонують закінчення слів в процесі набору. При введенні текстових повідомлень часто виникають технічні помилки, що пов'язані з невеликими розмірами клавіатури мобільних пристроїв. В таких випадках існуючі методи показують низьку якість передбачення можливих слів. Запропоновано метод предиктивного введення тексту на основі N-грам, що враховує можливі помилкові натискання сусідніх клавіш для QWERTY-клавіатур при наборі тексту на мобільних пристроях. Для визначення найбільш ймовірних закінчень слів використано статистичну модель мови на основі наборів тематичних текстів українською мовою.

An overview of existing methods of predictive text input, that suggesting the ending of words in the typing process. When entering text messages, technical errors often occur due to the small size of the keyboard of mobile devices. In such cases, the existing methods show a low quality of prediction of possible words. A method of predictive text input based on N-grams is proposed, which consider possible erroneous presses of adjacent keys for QWERTY keyboards when typing text on mobile devices. To determine the most probable word endings, a statistical language model based on sets of thematic texts in Ukrainian was used.

Вступ

Клавіатура на сучасному етапі розвитку є найбільш універсальним пристроєм для введення інформації. Понад сто років навчають людей швидко друкувати на клавіатурі та з рештою тенденція рухається в сторону зменшення її розмірів, тому ймовірність технічних помилок при наборі текстового повідомлення стає все частіше. Якщо раніше клавіатури були розміром з невеличку тумбу, то сучасна клавіатура вміщується на екрані смартфона.

Технічні помилки виникають з різних причин, але найбільш поширеною помилкою є, коли при введені текстового повідомлення, палець користувача на екрані займає більше місця ніж сама літера.

Проте метод прискореного введення тексту не враховує ці технічні помилки. Отже виникає необхідність в розробці покращеного метода прискореного введення текстів на основі N-грам.

Мета дослідження полягає у розробці інтелектуальної системи прискореного введення текстових повідомлень за рахунок передбачення найбільш ймовірних слів в процесі набору.

Для досягнення поставленої мети потрібно виконати наступні завдання:

- Провести аналіз методів передбачення слів при введені текстових повідомлень;
- Запропонувати модель мови для оцінки ймовірності слова з урахуванням введеного тексту;
- Розробити метод передбачення слів при введені текстових повідомлень;
- Реалізувати інтелектуальну систему передбачення слів при введені текстових повідомлень;
- Провести валідацію розробленого метода.

Основна частина

Метод прискореного введення текстового повідомлення дає змогу набирати цілі речення лише кількома натисканнями. Під час набору тексту пропонуються найбільш ймовірні варіанти слів чи фраз, а кожна наступна набрана літера буде тільки покращувати точність прогнозування. Це в свою чергу скоротить час набору тексту та покращити якість написання слів.

Розглянемо декілька способів набору тексту. Розглянемо їх більш детально.

WordWise – метод предиктивного набору тексту [1]. Передбачає використання допоміжних клавіш. Вибір літери в даному методі відбувається шляхом одночасного натискання цієї клавіші, а також допоміжної яка вказує позицію літери на клавіатурі. Найбільшої популярності здобув на мобільних телефонах. Іноді натискання двох клавіш було складно зробити через маленький розмір клавіатури.

Метод багатьох натискань - працює шляхом багатьох натискань більше однієї клавіші для введення одного слова [2]. Може використовуватись окремо, так і з складнішими методами. З плюсів, це можливість набрати слово якого нема в словнику, оскільки одна клавіша - дорівнює одній букві.

Метод одного натискання - введення тексту, що розрахований на зменшення кількості натискань [3]. Тому, кожний натиск може відповідати одночасно декільком словам.

Одним з перших способів для передбачення слів при наборі був програмний засіб T9, який використовував методи багатьох натискань для вводу тексту [4]. Мав велику популярність на мобільних пристроях які мали 12 кнопок. Під час роботи, T9 поєднує групи літер, які мають своє розміщення на клавіатурі мобільного пристрою. Після чого для послідовності літер, що сформувалась після натискання клавіш, виконується пошук по словнику та сортується за частотою використання. Але з появою сенсорних екранів перестав широко використовуватись.

Конкурентом T9 стала програма iTap, яка при послідовності трьох, або більше символів намагалась вгадати закінчення. На відміну від T9, яка намагалась підставити слово, що має стільки літер, скільки є в даний момент, iTap намагається передбачити і слова з більшою кількістю літер, аналізуючи не лише набрані літери поточного слова, а й попередній текст[5]. До того ж iTap може передбачати короткі

фрази. Також, ще однією з переваг є введення слів з автоматичним пробілом між словами.

Ще одним з прикладів використання методів предиктивного вводу тексту є «Пошуковий сервіс Google». В процесі набору тексту, користувачеві надаються пропозиції, які базуються на його попередніх запитах, а також найбільш популярних запитах загалом серед користувачів. Google використовує N-грами для покращення прогнозування, та використовує в своїх базах даних приховані параметри такі як «Середня кількість запитів на місяць». Створений список пропозицій може враховувати місцезнаходження користувача, для покращення точності.

Метод предиктивного введення тексту на основі N-грам дозволяють здійснювати прогнозування наступного слова, що користувач має намір ввести, базуючись на даних про частоту використання комбінацій слів у текстах мови, на якій здійснюється введення[6]. Такі комбінації або послідовності слів називають N-грамами, де число N відповідає кількості слів у цій послідовності. При значенні N рівним одиниці в даному випадку є слово, така послідовність називається Uni-Gram, а ймовірності використання послідовностей, що складаються з двох слів та більше використовуються для прогнозування наступного слова після введення попереднього, значення ймовірностей використання тієї чи іншої Uni-Gram можна використовувати для прогнозування закінчення певного слова, початок якого вже введено користувачем.

Слід зазначити, для того, щоб користувачеві було запропоноване слово, яке він має намір ввести, необхідним є точний збіг всіх введених до цього літер із літерами слова. Помилкове ж натискання сусідньої клавіші під час введення слова одразу ж виключить необхідне слово зі списку пропозицій, оскільки це слово не відповідатиме вже введеній послідовності літер, що в свою чергу, призведе до хибного результату роботи методу.

Тому, необхідна модифікація даного методу, яка полягає у використанні блоків літер, згрупованих відповідно до їх розташування на клавіатурі. Враховуючи це, розглядатимемо модифікацію методу предиктивного введення тексту на основі N-грам, що пропонується, у контексті введення тексту за допомогою саме такої клавіатури.

Використання блоків літер, згрупованих відповідно до розташування цих літер на клавіатурі, покращить роботі методу предиктивного введення тексту на основі N-грам навіть при помилковому натисканні сусідньої літери, що при швидкому введенні тексту має велику ймовірність. Тому, відповідно модифікації, що пропонується, при натисканні деякої клавіші на клавіатурі відбувається вибір не окремої літери, яка відповідає даній клавіші, а певного блоку літер.

Розподіл літер між блоками

Найбільш поширеною клавіатурою, яка встановлюється на сучасних смартфонах та решті сучасних пристроях, є QWERTY-клавіатура, де кожна кнопка відповідає певній літері. На мобільних пристроях із сенсорним екраном така

клавіатура розміщується безпосередньо на екрані пристрою і натискання клавіш на ній відбувається шляхом натискання у відповідній області екрану, проте все частіше виникають технічні помилки. Причина полягає в тому, що при натисканні на екран розмір пальця більший ніж розмір бажаної літери на сенсорному дисплеї і користувач натискає не ту літеру, що хотів.

Оскільки метод предиктивного введення тексту не враховує ці технічні помилки то і не може розпізнати слово. Тому, користувач не отримує бажаного результату.

Для початку потрібно розділимо клавіатуру на блоки літер, так щоб кожна літера утворювали власний блок з літер, які розташовані праворуч, знизу, ліворуч та зверху. Таким чином в кожному блоці буде від трьох до семи літер.

Приклад створення блоку, утвореного з трьох літер, наведено на рисунок 1 та семи літер на рисунок 2.

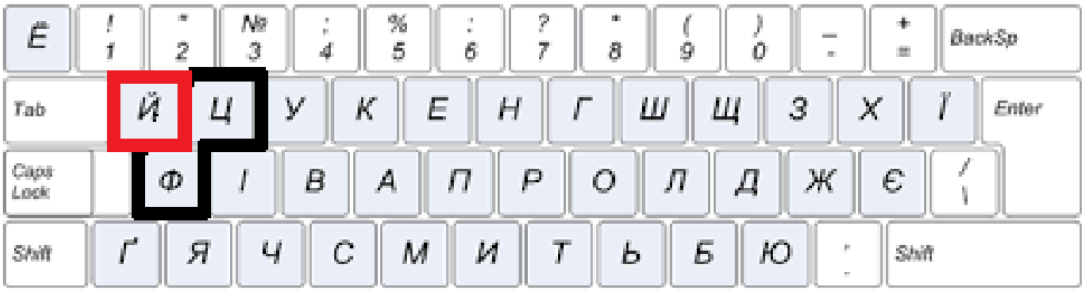


Рисунок 1 – Блок з трьох літер, для українського варіанту QWERTY-клавіатури

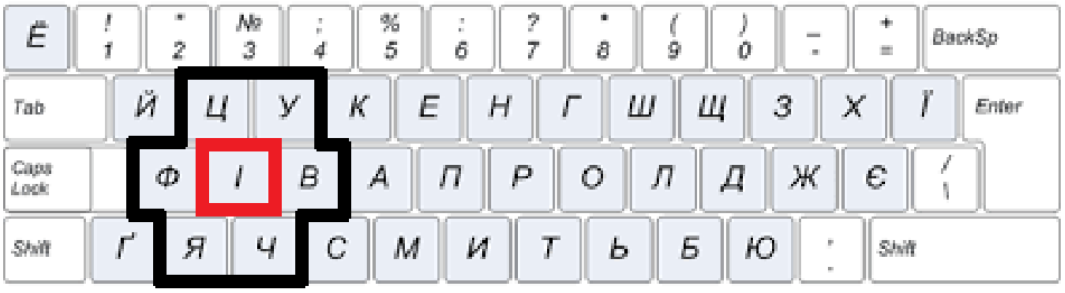


Рисунок 2 – Блок з семи літер, для українського варіанту QWERTY-клавіатури

В цьому випадку блоки перетинаються між собою, а кожна літера може бути у декількох блоках одночасно. Це сприятиме тому, що при помилковому натисканні літер, що розташовані ліворуч або праворуч, вище або нижче від бажаних, потрібна літера гарантовано потраплятиме до набору літер, що розглядатимуться при здійсненні прогнозування слова.

Так, наприклад, у блоці, що утворює літера «В», розташовуватиметься літера «І», що розташована ліворуч від літери «В», літера «А», що розташована праворуч від літери «В», літери «У» та «К», що розташовані вище від літери «В», літери «Ч» та «С», що розташовані нижче від літери «В», а також безпосередньо літера «В». Приклад створення блоку, що утворює літера «В», наведено на рисунок 3.

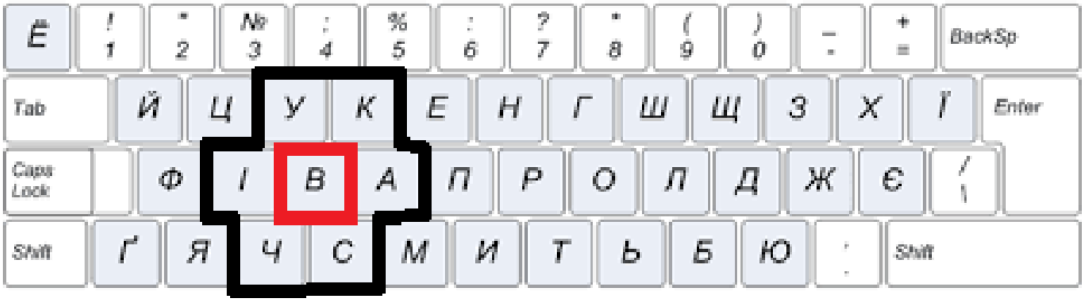


Рисунок 3 – Приклад створення блоку літери «В», для українського варіанту QWERTY-клавіатури

Аналогічно блоки створюються для інших літер. Оскільки кожна літера створює свій окремий блок, то кількість блоків буде рівна кількості літер на клавіатурі. Проте кількість літер у блоці буде не завжди рівна, так до прикладу блок літери «Я» буде мати всього три сусідні літери, праворуч «Ч» та зверху «Ф», «І».

Для прогнозування дуже важливо створити статистичну модель мови на основі тематичних текстів, що розширить перелік можливих слів-кандидатів та дасть можливість підвищити швидкість набору текстових повідомлень на мобільних пристроях. Зазвичай використовують N-грами декількох видів. Послідовність з одного елемента називають Uni-Gram, послідовність з двох елементів Bi-Gram та відповідно з трьох Tri-Gram. Принцип формування такого корпусу слів на основі текстів наведено на рисунок 4.

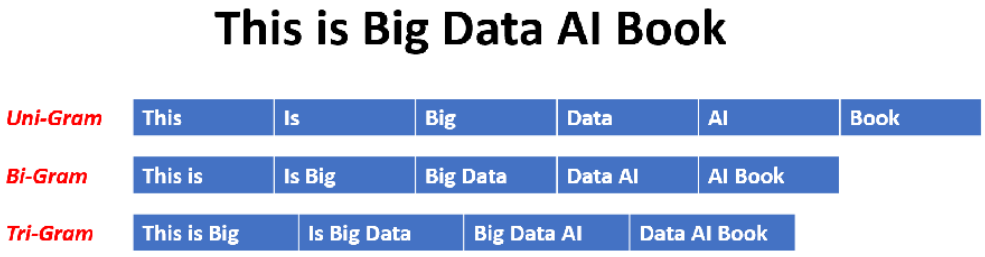


Рисунок 4 – Приклад розбивки тексту

Після розбиття Uni-Gram зберігаються у реляційній БД для зручності використання. В базі даних елементи Uni-Gram будуть мати вигляд таблиці з двома стовпчиками, де в першому назва слова, а другий це кількість раз його використання в тексті для навчання наведено на рисунку 5.

word	count
а	621
або	283
абсцес	1
аварію	2
аварія	3
авто	8
автобус	16
автобуси	7

Рисунок 5 – Uni-Gram в базі даних

Отже, модифікований метод прискореного введення тексту на основі N-грам можна показати наступним чином на рисунку 6.

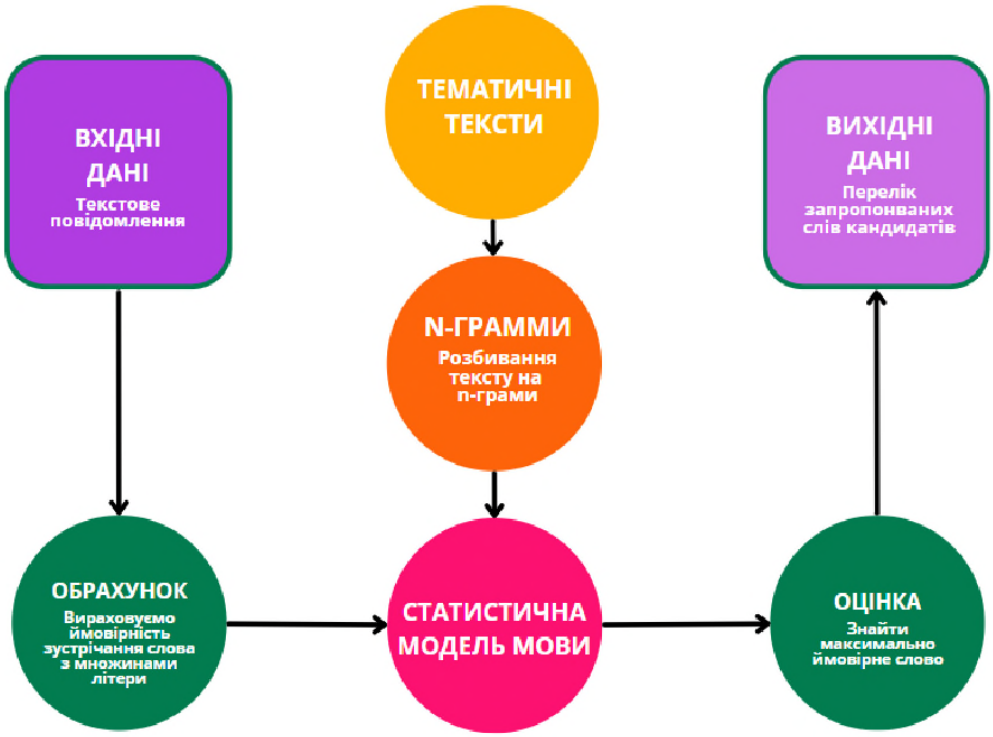


Рисунок 6 – Загальний приклад схеми роботи методу

На основі введеної користувачем послідовності літер формується послідовність блоків таким чином, що до кожного блоку входить літера, яку натиснули, а також літери-сусіди, що розташовуються ліворуч, праворуч, вище та нижче. Після цього серед слів статистичної моделі мови відбувається пошук, що відповідає заданій послідовності блоків.

Користувачеві відображаються перші слова, які мають максимальну оцінку ймовірності.

Висновки

В результаті роботи було вдосконалено метод предиктивного введення тексту з використанням статистичної моделі мови, що дало можливість прискорити введення текстових повідомлень. Відмінність від відомих методів полягає в тому, що при передбаченні можливих варіантів слів враховуються "літери-сусіди", які мають високу ймовірність помилкового натискання при наборі тексту на мобільних пристроях.

В подальшому планується реалізувати дану інформаційну технологію та провести експериментальні дослідження.

Перелік посилань

1. Як працює метод введення тексту WordWise URL: https://github.com/jaketae/wordwise
2. Метод багатьох натискань URL: https://en.wikipedia.org/wiki/Multi-tap
3. Метод одного натискання URL: https://uk.upwiki.one/wiki/Predictive_text
4. T9 передбачення тексту URL: https://uk.upwiki.one/wiki/T9_(predictive_text)
5. iTap інтелектуальний текст URL: https://uk.upwiki.one/wiki/ITap
6. Як застосовувати N-грами WordWise URL: https://towardsdatascience.com/text-generation-using-n-gram-model-8d12d9802aa0

Sun Dec 11 15:59:20 EET 2022, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 6%**

	Документ		Сумарний збіг по Базі Даних	
ID: 109357 Назва: КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА на тему Інтелектуальна система передбачення слів при введені текстових повідомлень Додано в БД: 2022-12-11 Автора: О.В. Мороз Керівники: Р.О. Багрій Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	72594	1117	2053 (3%)	34 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

UNICHECK

Ім'я користувача:
Кафедра КН

ID перевірки:
1013276889

Дата перевірки:
12.12.2022 12:33:48 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
12.12.2022 12:35:38 EET

ID користувача:
100005671

Назва документа: **КНм-21-1_Мороз_02**

Кількість сторінок: **77** Кількість слів: **12052** Кількість символів: **88379** Розмір файлу: **1.21 MB** ID файлу: **1013035558**

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

9.82%
Схожість

Найбільша схожість: **5.48%** з Інтернет-джерелом (https://core.ac.uk/download/pdf/323529261.pdf)

8.08% Джерела з Інтернету	49	Сторінка 79
3.08% Джерела з Бібліотеки	109	Сторінка 79

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

61.4%
Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за **8 слів** та **0%**)

Немає вилучених Інтернет-джерел

61.4% Вилученого тексту з Бібліотеки	1	Сторінка 79

Модифікації

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

Замінені символи	1
Підозріле форматування	15 сторінок

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ МАГІСТРА ДО ЗАХИСТУ ЗА РЕЗУЛЬТАТАМИ АНАЛІЗУ ЗВІТУ ПОДІБНОСТІ

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

Назва: *Інтелектуальна система передбачення слів при введені текстових повідомлень*

Автор: Мороз Олександр Вікторович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.т.н., доц. Багрій Р.О.

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.	відповідає
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, є законними і не є плагіатом, оскільки:

1) за програмою Anti-Plagiarism виявлені 1% є фрагментарними – містять поширені конструкції, загальновідомі терміни, скорочення та визначення.
2) За програмою UNICHECK виявлені 9.82%, що є запозиченнями, які розміщені в розділах аналізу існуючих технологій та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 1% і 9.82% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ Руслан Багрій

Гарант ОП _____ Руслан Багрій

Завідувач кафедри КН _____ Олександр Бармак

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ
кафедра комп'ютерних наук

ВІДГУК ОПОНЕНТА

на кваліфікаційну роботу магістра

гр. КНм-21-1 Мороза Олександра Вікторовича за темою: *Інтелектуальна система передбачення слів при введені текстових повідомлень*

1. **Актуальність обраної теми**

Тема передбачення слів при введені текстових повідомлень є актуальною тому що основним способом обміну інформацією є використання саме текстових повідомлень за допомогою різних месенджерів. Актуальність задачі добре відображено у кваліфікаційній роботі магістра.

2. **Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт**

Робота відповідає вимогам предметної області спеціальності 122 Комп'ютерні науки, оскільки в роботі виконуються теоретичні та експериментальні дослідження в галузі комп'ютерних наук, застосовуються алгоритмічні принципи в моделюванні, проектуванні, розробці та супроводі інформаційних систем.

3. **Повнота розкриття мети та завдань дослідження**

Метою магістерської роботи є розробка інтелектуальної системи прискореного введення текстових повідомлень за рахунок передбачення найбільш ймовірних слів в процесі набору, що дозволяє підвищити швидкість набору тексту на мобільних пристроях за рахунок обробки ситуацій, коли користувач помилкового натискає невірні літери. Завдання дослідження визначені вірно і дозволяють досягти поставлену мету. Робота повністю розкриває поставлену мету та усі завдання, що були поставлені.

4. **Наявність наукової новизни**

В роботі описуються загальновідомі методи для передбачення найбільш ймовірних слів в процесі набору текстових повідомлень на мобільних пристроях. Вдосконалення методу полягало в тому, що при передбаченні можливих варіантів слів враховуються "літери-сусіди", які мають високу ймовірність помилкового натискання при наборі тексту на мобільних пристроях. Окрім цього в роботі було проведено валідацію роботи методу з використанням двох корпусів слів української мови. Зважаючи на це можна зробити висновок, що в роботі наявна наукова новизна в достатньому обсязі. Також слід відзначити, що проведене дослідження було опубліковано на XIV всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2022» 18-19 листопада 2022 р., м.

Хмельницький, Україна.

5. Зміст кожного розділу роботи

В першому розділі проведено аналіз існуючих методів введення тексту на мобільних пристроях, а також програмні засоби, що реалізують ці методи. Визначено, що під час швидкого введення та в умовах обмежених розмірів екранних клавіатур на сучасних мобільних пристроях достатньо ймовірним є помилкове натискання сусідньої клавіші. Отже, поставлено мету розробити метод, що буде враховувати можливість таких помилкових натискань.

В другому розділі вдосконалено метод предиктивного введення тексту з використанням статистичної моделі мови, що дало можливість прискорити введення текстових повідомлень. Відмінність від відомих методів полягає в тому, що при передбаченні можливих варіантів слів враховуються "літери-сусіди", які мають високу ймовірність помилкового натискання при наборі тексту на мобільних пристроях.

У третьому розділі проаналізовано мови програмування, їхніх переваги та недоліки, обрано об'єктно-орієнтовних підхід для розробки та вдосконалення метода прискореного введення текст на основі N-грам. Розроблені модулі працюють незалежно, що покращує їх використання та вдосконалення для різних налаштувань системи передбачення слів.

У четвертому розділі проведено тестування розробленого методу для різних текстових корпусів української мови та отримано, що запропонований підхід кращий за не модифікований на 13%, що є достатньо значним результатом.

6. Ступінь розкриття теми роботи

В роботі недостатньо уваги приділено існуючим інтелектуальним методам передбачення слів під час набору тексту. Основний акцент роботи був спрямований на методі, що використовує N-gram.

Якість оформлення кваліфікаційної роботи

Загалом оформлення роботи відповідає поставленим вимогам, але є певні недоліки що стосуються мови та граматики.

7. Недоліки кваліфікаційної роботи

В роботі недостатньо уваги приділено існуючим інтелектуальним методам передбачення слів під час набору тексту.

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

Роботі виконана студентом в достатньому обсязі, наявна наукова новизна та виконані усі поставлені завдання. Кваліфікаційна робота допускається до захисту, але враховуючи допущені помилки, заслуговує оцінку “задовільно”.

Опонент _____ д. ф-м. н., професор Бедратюк Л.П.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ
кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА

на кваліфікаційну роботу магістра

гр. КНм-21-1 Мороза Олександра Вікторовича за темою: Інтелектуальна система передбачення слів при введені текстових повідомлень

1. Актуальність теми

Актуальність теми достатньо обґрунтована, оскільки при обміні текстовими повідомленнями з використання мобільних пристроїв, через малу розмірність екрану, часто виникають технічні помилки в процесі вводу літер. Отже, підвищення ефективності введення тексту на сучасних мобільних пристроях з використанням інтелектуальної системи передбачення слів є актуальною задачею.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Теми кваліфікаційної роботи "Інтелектуальна система передбачення слів при введені текстових повідомлень" відповідає предметній області спеціальності 122 Комп'ютерні науки та вимогам до кваліфікаційної роботи магістра, оскільки об'єктом дослідження є процес передбачення слів при наборі текстових повідомлень, предметом дослідження – моделі, методи, підходи та алгоритми прискореного введення текстових повідомлень для мобільних пристроїв.

3. Професійні та особистісні якості магістранта

Мороз О. В. під час роботи над кваліфікаційною роботою магістра продемонстрував посередній рівень знань та умінь за спеціальністю "Комп'ютерні науки".

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Робота виконана самостійно, академічного плагіату не виявлено, стосовно всіх запозичень наведено відповідні посилання на джерела.

5. Наукова новизна та оригінальність запропонованих підходів

Отримані такі результати: вдосконалено метод предиктивного введення тексту з використанням статистичної моделі мови, що дало можливість прискорити введення текстових повідомлень. Відмінність від відомих методів полягає в тому, що при передбаченні можливих варіантів слів враховуються "літери-сусіди", які мають високу ймовірність помилкового натискання при наборі тексту на мобільних пристроях. Отримані результати оприлюднені на XIII всеукраїнської науково-практичної конференції

«Актуальні проблеми комп'ютерних наук АПКН-2022», 18-19 листопада 2022 р., м. Хмельницький, Україна, доповідь на тему «Метод передбачення слів при введені текстового повідомлення».

6. Ступінь оволодіння методами дослідження

Студент Мороз О. В. має посередній ступінь володіння методами дослідження, що були використанні у роботі.

7. Повнота та якість розкриття теми роботи

Мета роботи повністю розкрита, отримані результати підтверджують достовірність наукових положень.

8. Логічність, послідовність, аргументованість, літературна грамотність викладу матеріалу

Викладення матеріалу логічне, послідовне та аргументоване. Мова і стиль викладення кваліфікаційної роботи магістра відповідають стандартам, що забезпечує доступність сприймання матеріалу і відповідає вимогам до сучасних наукових робіт.

9. Можливість практичного застосування кваліфікаційної роботи, окремих її частин

Може мати практичне значення при введення тексту в умовах обмежених розмірів екранних клавіатур сучасних мобільних пристроїв.

10. Висновок про можливість допуску кваліфікаційної роботи до захисту, на яку оцінку заслуговує робота

Вважаю, що кваліфікаційна робота студента Мороза Олександра Вікторовича може бути рекомендована до захисту та заслуговує на оцінку "задовільно".

Науковий керівник _____ к.т.н., доц. Руслан Багрій