ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп’ютерних систем та автоматики

Кафедра комп’ютерних систем управління

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології

Освітньо-професійна програма Інтелектуальні комп’ютерні системи

ЗАТВЕРДЖУЮ

Завідувач кафедри КСУ
Дубовой В.М.

«___» _____ 2019 року

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

Розробка методів та дослідження моделей обробки текстових документів в інтелектуальних системах автоматизації

08-01.МКР.004.00.000

Студент групи 2АКІТ-18м Калінчук Р.С.

Керівник к.т.н., доцент Биков М.М.

Рецензент к.т.н., доцент Кривогубченко С.Г.

Вінниця ВНТУ – 2019

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп’ютерних систем та автоматики

Кафедра комп’ютерних систем управління

Освітньо-кваліфікаційний рівень магістр

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані

Освітньо-професійна програма Інтелектуальні комп’ютерні системи

ЗАТВЕРДЖУЮ
Завідувач кафедри КСУ
Дубовой В.М.

«_2__» ____09_____ 2019 року

Протокол № 1 засідання кафедри
КСУ від 2.09.2019р.

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Калінчуку Роману Сергійовичу

(прізвище, ім’я, по батькові)

1. Тема магістерської кваліфікаційної роботи «Розробка методів та дослідження моделей обробки текстових документів в інтелектуальних системах автоматизації»
керівник магістерської кваліфікаційної роботи Биков Микола Максимович, к. т. н., доцент
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від “02” 10 2019 року № 254
2. Строк подання студентом магістерської кваліфікаційної роботи 10.12. 2019 року
3. Вихідні дані до магістерської кваліфікаційної роботи україномовний текст без формул, абревіатур і скорочень обсягом не менше 90 000 слів, тип шрифта – друкований заданого стилю. середовища моделювання – Матлаб, Visual Studio 2017.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Вступ (актуальність проблеми, постановка задачі); Варіантний аналіз методів електронізації текстових документів та обґрунтування вибраного

варіанту; Розробка ефективних методів і моделей електронізації текстових документів; Розробка алгоритмів та програмного забезпечення; Машинний експеримент; Економічна частина.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) 1. Структурна схема системи розпізнавання тексту – 1 плакат ; 2. Схема взаємодії програмного забезпечення – 1 аркуш ; 3. Схема алгоритму пошуку морфем української мови – 1 аркуш; 4. Схема алгоритму визначення статистичних ймовірностей появи морфем – 1 аркуш; 5. Схема алгоритму визначення залежності довжини слова від кількості складів –1 аркуш; 6. Результати машинного експерименту – 1 плакат.6. Консультанти розділів магістерської кваліфікаційної роботи

Розділ змістової частини роботи	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	к.т.н, професор кафедри КСУ Биков М.М.		
2	к.т.н, професор кафедри КСУ Биков М.М.		
3	к.т.н, професор кафедри КСУ Биков М.М.		
4	к.т.н., доцент кафедри ЕПВМ Ратушняк О.Г		

Календарний план

№ з/п	Назва етапів Роботи	Строк виконання етапів роботи	Примітка
1	Вступ, варіантний аналіз методів обробки тексту та обгрунтування обраного варіанту. Постановка задач дослідження	02.10.2019р.	
2	Розробка і дослідження ефективних методів і моделей електронізації текстових документів	20.10.2019 р.	
3	Розробка алгоритмів та програмного забезпечення	02.11.2019 р.	
4	Проведення експериментальних досліджень.	15.11.2019 р.	
5	Підготовка економічної частини	22.11.2019 р.	
6	Оформлення пояснювальної записки, графічного матеріалу і презентації	30.11.2019 р.	
7	Апробація результатів дослідження. Публікації	06.12.2019 р.	
8	Попередній захист	10.12.2019 р.	
8	Захист МКР	12.12.2019 р.	

Дата видачі завдання "_02__" __09___2019 року

Студент _____ Калінчук Р. С.

Керівник магістерської кваліфікаційної роботи _____ Биков М.М.

ЗМІСТ

АНОТАЦІЯ

Магістерська кваліфікаційна робота присвячена розробці інтелектуальних методів і дослідженню моделей обробки текстових документів в системах автоматизації.

Розроблено математичну модель процесу розпізнавання тексту і визначено критерії оцінки його ефективності, представлено математичні основи визначення інформаційної структури тексту і його морфологічних характеристик, розроблено метод автоматичного виділення морфем в тексті.

Розроблені алгоритми і програмне забезпечення створення бази даних морфем української мови для систем автоматичного розпізнавання, необхідної для реалізації нової інформаційної технології введення і оброблення текстових документів, запропонованої авторами.

В економічній частині проведено розрахунок ефективності розробки.

Розробка дозволить поліпшити практичне застосування систем автоматичного розпізнавання образів в системах управління.

Summary

Master's qualification thesis is devoted to development of intellectual methods and to researching of the text documents processing models in the automation systems.

The mathematical model of process of recognition of the text is developed and criteria for evaluation of its efficiency is defined, the mathematical grounds of determination of information structure of the text and its morphological characteristics is presented, the method of automatic allocation of morphemes in the text is developed.

Algorithms and the software for creation of the morphemes database of Ukrainian language are developed for the systems of automatic recognition, necessary for realization of new information technology for input and processing of text documents, offered by authors.

In an economic part calculation of economic efficiency of development are executed.

Development will allow to improve practical using of automatic recognition of images in the automation systems.

ВСТУП

Проблема розпізнавання образів є однією з основних наукових проблем в області створення нових високоефективних засобів обчислювальної техніки, обробки інформації, автоматизації виробничих процесів і наукових досліджень.

Аналіз сучасного стану проблеми побудови ефективних стратегій розпізнавання показав, що її опрацюванню приділено недостатню увагу. Разом з тим, за думкою фахівців в області розпізнавання зображень, оптимізація процесу розпізнавання графічних образів можлива тільки за допомогою процедури „від початку до кінця", тобто одночасно за всіма елементами системи розпізнавання. Тому розробка ефективних методів і алгоритмів розпізнавання образів, зокрема розпізнавання символів тексту є на сьогодні досить актуальною проблемою, розв'язанню якої і присвячені дослідження, проведені в даній магістерській роботі [1].

Необхідність підвищення ефективності процесів автоматизації обробки текстових документів з метою прийняття оперативних рішень в автоматизованих виробничих системах ставить на порядок дня задачу розробки більш прогресивних технологій для розв'язання поставленої проблеми. Пошук та використання текстових документів з потрібною інформацією потребує їх зберігання в електронізованому вигляді. Технології таких фірм, як Adobe Acrobat і AT@T Labs дозволяють представляти тексти у pdf- та djv форматах і, таким чином, здійснювати їх електронізацію, проте для опрацювання таких документів з метою їх аналізу та інтерпретації потрібна людина-оператор. Використання традиційних технологій електронізації документів в текстових форматах, які дозволяють представити символи в ASCII кодах, і, таким чином, автоматизувати їх аналіз, передбачає посимвольне розпізнавання графічного зображення тексту за допомогою наявних програмних засобів (наприклад, FineReader). Однак такі технології в своїх історичних витоках орієнтовані на брак апаратних ресурсів (швидкодії і пам'яті), не враховують технічних можливостей сучасних обчислювальних систем і мікропроцесорних засобів, а також не використовують

мовних складових в інформаційній ієрархії текстового документа[2,3]. Тому тема магістерської роботи, присвячена розробці ефективних методів оброки текстових документів для електронізації , є актуальною.

Зв'язок роботи з науковими програмами, планами, темами.

Робота виконана згідно плану кафедральної науково-дослідної роботи № 52К4 "Методи прийняття рішень на основі принципів обчислювального інтелекту".

Мета роботи – підвищення ефективності процесу розпізнавання символів в системах автоматичного розпізнавання тексту.

Запропонована технологія для підвищення швидкості і надійності введення і розпізнавання передбачає використання поряд з параметричними складовими графічного опису тексту також мовних складових: лексичних, морфологічних і синтаксичних. Для реалізації даної технології необхідно розв'язати низку задач, а саме:

- розробка програмного забезпечення для створення бази даних морфем української мови;
- розробка програмного забезпечення для визначення статистик морфем – імовірності їх спостереження P_c і імовірності переходів між ними $P_{p;}$
- вибір ефективного опису еталонів морфем;
- розробка алгоритмів сегментації тексту на морфеми;
- адаптація алгоритмів розпізнавання на ПММ до розпізнавання морфем.

Об'єктом дослідження є процес розпізнавання символів.

Предметом дослідження є методи і алгоритми опису та розпізнавання морфем і символів.

Методи досліджень. Для розв'язання поставлених задач дослідження використовуються математичні методи моделювання складних систем; методи теорії ймовірностей і статистики; методи теорії розпізнавання; методи планування експерименту і обробки його результатів.

Наукова новизна отриманих результатів полягає в наступному :

- вперше вибрано критерій ефективності систем розпізнавання зорових образів, оснований на моделях потенційної і реальної систем. Він дозволяє керувати вибором оптимальних параметрів системи уже на етапі проектування, а не тільки оцінювати ефективність уже готових систем;
- вперше запропонована інформаційна технологія електронізації текстів, яка передбачає їх введення і опрацювання за допомогою розподіленої комп'ютерно-мікропроцесорної системи.

Практична цінність одержаних результатів полягає в можливості використання розроблених алгоритмів і програмного забезпечення для проектування ефективних систем розпізнавання образів модульного типу, які можуть розв'язувати більш складні задачі шляхом підключення нових модулів. Результати роботи використовуються також в навчальному процесі в дисциплінах "Інтелектуальні технології" та "Інтелектуальні засоби систем автоматики і управління" і при виконанні держбюджетних науково-дослідних робіт на кафедрі КСУ.

Достовірність теоретичних положень даної магістерської роботи визначається коректністю постановки задач, строгим доведенням аналітичних співвідношень і наукових положень шляхом використання адекватних математичних методів, відповідністю отриманих у роботі результатів уже відомим на сьогодні при їх порівнянні, та збіжністю результатів машинного експерименту з результатами математичного моделювання.

Апробація результатів роботи. Основні положення й результати досліджень доповідалися й обговорювалися на форумі молодих науковців ВНТУ (м. Вінниця, 2019).

Публікації. За тематикою дослідження опублікована робота в репозиторії ВНТУ.

1 ВАРІАНТНИЙ АНАЛІЗ МЕТОДІВ ОБРОБКИ ТЕКСТУ ТА ОБГРУНТУВАННЯ ВИБРАНОГО ВАРІАНТУ

1.1 Суть технічної проблеми, що виникла на сучасному етапі розвитку теорії розпізнавання текстів

Необхідність підвищення ефективності процесів автоматизації обробки текстових документів з метою прийняття оперативних рішень в автоматизованих виробничих системах ставить на порядок дня задачу розробки більш прогресивних, ніж існуючі на сьогодні, технологій для розв'язання поставленої проблеми. Пошук та використання текстових документів з потрібною інформацією потребує їх зберігання в електронізованому вигляді. Технології таких фірм, як Adobe Acrobat і AT@T Labs дозволяють представляти тексти у pdf- та djv форматах і, таким чином, здійснювати їх електронізацію, проте для опрацювання таких документів з метою їх аналізу та інтерпретації потрібна людина-оператор.

Електронізація текстових документів, представлених в графічних форматах, потребує представлення символів в ASCII кодах, що вимагає їх розпізнавння в процесі введення. Тому на сьогодні актуальною стає проблема розробки нових інформаційних технологій обробки текстових документів з метою підвищення надійності і швидкості їх автоматичного розпізнавання. В даній магістерській роботі пропонується нова інформаційна технологія переведення текстів з графічного формату в електроннний, в якій процеси введення і розпізнавання розподілені між мікропроцесорним пристроєм введення, що виконує попередній аналіз і введення графічного текстового документу, і комп'ютером, що виконує остаточне розпізнавання тексту і представляє його в ASCII коді. Такий підхід дозволяє підвищити ефективність електронізації тексту та зменшити вартість системи. Під час реалізації цієї технології однією з проблем, які потрібно розв´язати, є автоматизація процесу побудови бази даних морфем української

мови, які використовуються в даній технології в якості окремих елементів розпізнавання [1,2].

1.2 Існуючі способи вирішення проблеми

Пошук джерел інформації з вирішуваної проблеми і їх аналіз показали, що традиційні технології обробки текстових документів, представлених в графічному форматі, з метою їх електронізації, ґрунтуються на введенні графічного зображення, а потім його посимвольного розпізнавання в комп'ютері за допомогою розроблених програмних продуктів [3-13]. Процес розпізнавання передбачає виконання таких етапів: покращення зображення за допомогою фільтрації; попереднє перетворення до вигляду, зручного для виділення вибраних інформативних ознак; розподіл на окремі символи; описання виділених зображень символів в просторі ознак чи в структурному вигляді; застосування для розпізнавання алгоритмів, що відповідають обраному методу описання образів [14, 15, 16, 17, 18]. Даний підхід має переваги, які полягають у незначних затратах пам'яті, зумовлених невеликою кількістю еталонів символів. Серед недоліків основними є недостатня швидкодія, зумовлена значними обчислювальними затратами на процедури сегментації і розпізнавння символів тексту, та недостатня точність розпізнавання.

Більшість методів розпізнавання символів тексту на сьогодні використовують процедуру посимвольного розпізнавання. Така процедура дає прийнятні за точністю результати тільки для текстів з стилізованими шрифтами, отриманими методом друку. Однак її використання дає набагато гірші результати під час розпізнавання зашумлених друкованих текстів. Це викликано, в першу чергу, труднощами в реалізації процедури сегментації окремих символів, а також ігноруванням алгоритмом розпізнавання наявної лінгвістичної інформації. Під час розпізнавання рукописних символів ці недоліки підсилюються в декілька разів. В цьому випадку через злитне написання слів процедура сегментації символів стає

на порядок складнішою, а її часте застосування значно збільшує час розпізнавання. Крім того, велика різноманітність стилів написання рукописних текстів (підтвердженням чого є база даних рукописних символів MNIST [19], вимагає під час розпізнавання порівняння відокремленого символу з великою кількістю еталонів, або виконання процедури його перетворення до еталонного стилю написання.

Розглянемо основні етапи розпізнавання тексту методами, що використовуються на сьогодні.

Розпізнавання образів – це процес віднесення об’єкту до одного з попередньо заданих класів з урахуванням його властивостей та ознак.

Процедура розпізнавання образів є складним процесом, який вимагає комплексного підходу. Складність процесу розпізнавання образів полягає у тому, що зображення завжди деформується при його введенні в комп’ютер за допомогою скануючого пристрою, тому на першому кроці виникає задача попередньої обробки зображень. Будь - який об'єкт чи образ, що підлягає розпізнаванню і класифікації, має ряд розпізнавальних якостей чи ознак. Другим кроком усякого процесу розпізнавання, реалізується він обчислювальною машиною чи людиною, є аналіз задачі вибору розпізнавальних ознак і визначення способу їх виділення (виміру). Третім кроком процесу розпізнавання є саме класифікація образу за виділеними ознаками [15].

Структурна схема системи розпізнавання образів, яка реалізує усі три кроки, зображена на рисунку 1.1:

Рисунок 1.1 – Структурна схема системи розпізнавання образів

Блок попередньої обробки здійснює операції по поліпшенню зображення шляхом фільтрації завад у виді дрібних ізольованих цяток, ліквідації так званої бахроми, дрібних прогалин (пустот) усередині ліній - раковин і, якщо необхідно, операції утоншення ліній [17,18].

Кожний підданий попередній обробці об'єкт потім повинен бути поданий у виді послідовності інформативних ознак. Цей процес складається, по-перше, із сегментації і, по-друге, із виділення неподільних елементів. Іншими словами об'єкт розділяється на частини і кожна виділена частина, у свою чергу, ідентифікується щодо заданої множини неподільних елементів. Виконання цієї функції покладається на спеціальний блок, який називається блоком виділення ознак. Очевидно, що кількість ознак, необхідна для успішного вирішення деякої задачі розпізнавання, залежить від розподільних якостей обраних ознак. Задача вибору ознак ускладнюється звичайно тією обставиною, що найбільш важливі ознаки не завжди легко виміряти або, як виявляється в багатьох випадках, відповідні можливості виміру стримуються економічними факторами. Основною задачею класифікатора природно є задача класифікації, проте в залежності від конкретної предметної області на нього можуть покладатися і більш широкі функції, наприклад вказівка, до якого з образів ближче усього розпізнаваний об'єкт [6,7,11].

Процес формування зображення супроводжується його спотворенням за рахунок недосконалості технічних засобів і процесу формування, наявності шумів і перешкод. Передбачається, що формування зображення в разі потреби наступної обробки завершується його представленням у цифровій формі.

Попередня обробка служить для підвищення якості зображення перед його подальшим аналізом. Звичайно в неї включають наступні основні операції:

- корекцію рівнів яскравості по всій області зображення;
- усунення шумів;
- перетворення зображення.

Розглянемо коротко найбільш розповсюджені математичні методи

попередньої обробки зображень.

Бінарне представлення зображень. Найпростіший спосіб представлення зображень, що називається бінарним, вимагає задати усього два рівні функції яскравості. У багатьох випадках, особливо при роботі в реальному масштабі часу, такий підхід дозволяє ефективно вирішувати задачі аналізу, обробки, розпізнавання зображень. Крім того, існує великий клас графічних і текстових задач, у яких зображення по самій природі є бінарними. Особливо широко, бінарні зображення, використовуються в системах технічного зору промислових роботів.

Основною проблемою при бінарному представленні реального зображення є вибір порогу, що визначає розбивку функції яскравості на два рівні. Навіть на контрастних зображеннях у реальних умовах існує шумовий фон, виникають різного роду перешкоди, що маскують корисні об'єкти або формують помилкові об'єкти. Раціональним вибором порогу можна звести такі спотворення до мінімуму [18].

Поріг може бути постійним і адаптивним. У першому випадку він встановлюється заздалегідь у вигляді деякого граничного значення функції яскравості, що не залежить від властивостей аналізованого зображення, і може бути постійним для всього зображення, або заданим чином змінюватися по координатному полю

$$S_б(x,y)=\begin{cases}1 \quad при \quad S(x,y)\geq S_0(x,y);\\ \\ 0 \quad в \ інших \ випадках,\end{cases} \tag{1.1}$$

де S *(x, y)* – вихідне зображення;

Sy(x, y) – бінарне зображення;

So (x, y) – граничний рівень.

В другому випадку поріг формується в результаті деякої обробки початкового зображення

$$S_б(x,y)=\begin{cases}1 \quad при \quad S(x,y)\geq \Phi\left[S(x,y)\right];\\ \\ 0 \quad в \ інших \ випадках.\end{cases} \tag{1.2}$$

В якості функції Ф *[S (x, y)]*, що визначає вид обробки, використовуються різноманітні оператори. Найбільш розповсюдженими є наступні:

а) оператор вибору максимального значення функції яскравості початкового зображення;

б) оператор розподілу двох основних піків на гістограмі яскравості, що відповідають об'єктам і фону, що виділяються;

в) оператор згладжування, що робить усереднення функції яскравості в деякому просторовому ”вікні” Ω, що стоїть на заданій відстані (x_0, y_0) від точки аналізу;

г) оператор послідовного аналізу, у якому використовують два і більш порогових рівні; області, у яких функція яскравості перевищує максимальний поріг, відносяться до виділених об'єктів, не перевищує мінімального порога – до фону, а проміжні області класифікуються після одержання додаткової інформації з використанням найрізноманітніших критеріїв;

д) оператор кореляційного аналізу.

Параметричне представлення зображень. Під параметричним представленням зображення розуміють його опис за допомогою деяких ознак (параметрів). При використанні такого опису можна виділити два різні підходи [3,4,5].

Сутність першого підходу полягає в тім, що зображення прагнуть описати за допомогою функціонала щільності імовірності або його дискретних аналогів. При цьому всі ознаки реальних об'єктів зображення трансформуються через функцію яскравості в параметри щільності розподілу. Якщо вид щільності розподілу визначається фізичними думками за допомогою деякої математичної

моделі, то задачу можна вважати вирішеною. Для практичного використання знайденого представлення варто вказати тільки спосіб одержання оцінок параметрів щільності розподілу, однозначно зв'язаних з ознаками досліджуваних об'єктів.

Складнішою є справа, коли теоретично визначити вид щільності розподілу неможливо. Якщо ми не відмовляємося від опису зображення за допомогою щільності розподілу, то єдине рішення в цьому випадку – одержати оцінку самої щільності, використовуючи експериментальні дані. Для спрощення запису розглянемо оцінки одномірної щільності розподілу [9].

Нехай $x_1, x_2, ..., x_n$ – незалежні вибіркові значення випадкової величини *x', що* має функцію розподілу $F_x = P\{x' \le x\}$. Розташовуючи їх в порядку зростання і визначаючи відносне число вибіркових значень, не перевищуючи значення *x*, одержимо так називану емпіричну функцію розподілу

$$F_N(x) = \begin{cases} 0 \ - при \quad x \le x_1; \\ k/n \ - при \quad x_k \le x \le x_{k+1}, \ \ k = 1, ..., \ \ N-1; \\ 1 \ - при \quad x \ge x_N. \end{cases} \tag{1.3}$$

Вона являє собою східчасту функцію зі стрибками, рівними 1/*N* у кожній з *N* точок, так що похідна від неї, називається емпіричною щільністю розподілу, а її дисперсія рівна

$$W_N(x) = \frac{1}{N} \sum_{k=1}^{N} \delta(x - x_k), \tag{1.4}$$

де $\delta(x - x_k)$ — дельта-функція Дірака.

Сутність другого підходу до параметричного представлення зображень полягає у відмові від його повного статистичного опису за допомогою розподілу

ймовірностей функції яскравості і використанні для опису набору різних ознак з визначенням їх спільної щільності розподілу. Наприклад, для параметричного представлення зображення можуть бути використані наступні ознаки:

а) ознаки яскравості, у якості яких виступають такі величини, як яскравість в одній точці зображення або усереднена в деякій площі, спектральні складові випромінювання, координати кольору ;

б) гістограмні ознаки, сформовані на основі деякої обробки гістограми розподілу яскравостей елементів зображення, що відрізняється від приведеної вище тим, що вона обчислюється по одному зразку зображення, а не по безлічі його реалізацій

$$\Gamma(h) = N(h) / N_0 , \qquad (1.5)$$

де $N(h)$ – число елементів, яскравість яких перевищує рівень h;

N_0 – повне число елементів зображення.

Гістограма може будуватися по всьому зображенню або по окремим його фрагментам, а в якості гістограмних використовують наступні ознаки (їх може бути запропоновано багато): енергія, середнє значення, дисперсія, асиметрія, ексцес, ентропія, а також ті , що визначаються на основі двохмірної гістограми $\Gamma(g,h)$: автокореляція, коваріація, момент інерції, середня абсолютна різниця, “зворотна різниця”.

в) просторово-спектральні ознаки, що утворяться шляхом накладання на спектральний образ зображення областей різної форми (щілина, кільце, сектор) і наступного інтегрування по одній з координат у межах обраної області;

г) геометричні ознаки, у якості яких можуть бути використані описані вище найпростіші елементи зображення, а також різні контури, як контури яскравості, так і контури кольорів.

д) текстурні ознаки, утворені на основі обробки текстур статистично однорідних фрагментів зображення, що мають деяку просторову регулярність; у

якості текстурних можуть бути використані гістограмні ознаки [9], якщо попередньою обробкою вихідне зображення сегментовано на текстурні фрагменти. Інший спосіб формування текстурних ознак оснований на виміру просторових частот [10].

Фільтрація зображень. В обробці зображень застосовується множина різноманітних методів фільтрації. По способу реалізації їх можна розділити на дві групи – лінійні і нелінійні.

Лінійна фільтрація зображень. Широке розповсюдження в обробці зображень отримали лінійні фільтри. Це обумовлено наявністю добре розробленого математичного пристрою лінійної фільтрації, включаючи оптимізацію, методів проектування таких фільтрів (як аналогових, так і цифрових), а також достатньо високої ефективності їх практичного застосування.

Вінеровська фільтрація. В класичній постановці задача фільтрації випадкового процесу або випадкового поля формується наступним чином. Випадкове поле, що вміщує адитивну суміш шумів і сигналів поступає на вхід лінійного фільтра. Необхідно таким чином підібрати характеристику лінійного фільтру, щоб мінімізувати середньоквадратичну похибку виділення сигнальної складової

$$\varepsilon^2 = \left\langle \int_T \int_\Omega \{h(r,\rho,t,\tau)S(\rho,\tau)d\rho d\tau - m(r,t)\}^2 \right\rangle. \qquad (1.6)$$

Виконуючи статистичне усереднення і визначаючи імпульсну перехідну характеристику фільтра $h(\cdot)$, при якій середньоквадратична похибка мінімальна, отримаємо наступне рівняння:

$$\int_T \int_\Omega h(r,\rho_1,t,\tau_1)R_{ss}(\rho_1,\rho_2,\tau_1,\tau_2)d\rho_1 d\tau_1 = R_{ms}(r,\rho_2,t,\tau_2), \qquad (1.7)$$

де $R_{ss}(\rho_1,\rho_2,\tau_1,\tau_2)=\langle S(\rho_1,\tau_1)S(\rho_2,\tau_2)\rangle$ – кореляційна функція вхідного впливу;

$R_{ms}(r,\rho_2,t,\tau_2)=\langle m(r,t)S(\rho_2,\tau_2)\rangle$ – взаємна кореляційна функція вхідного впливу і сигнальної складової.

Звичайно сигнальна і шумова складові незалежні, так що кореляційна функція вхідного впливу рівна сумі сигнальної і шумової кореляційних функцій, а взаємна кореляційна функція співпадає з сигнальною кореляційною функцією.

Фільтрація плоских зображень просторових об’єктів. Необхідність врахування впливу завад пов'язана з можливістю появи системи спостереження сторонніх об’єктів (наприклад, об’єктів крапкового характеру) і наявністю шумів сприймаючих пристроїв. Вся методика врахування завад повинна бути адекватною реальній ситуації, яка прийнята концепції розпізнавання, використовуючи для цього ознаки. Наприклад, процедура оцінки впливу завад засобом введення деякого випадкового розкиду значень ознак в ряді випадків приводить до появи додаткових похибок, які важко піддаються обліку. Тому доцільний облік завад вже на рівні абсолютного опису зображень. При цьому процес введення шумів при моделюванні алгоритмів фільтрації (необхідність такого моделювання пов’язана з труднощами постановки відповідних реальних експериментів) може заключатися в наступному: аналізується кожний елемент матриці зображення і з раніше заданою ймовірністю ρ_n його стан змінюється на протилежний. Такий метод введення завад зручний для реалізації на ЕВМ і дозволяє описувати рівень шумів тільки одним параметром (величиною ρ_n, яка виражена в процентах). Якість фільтрації легко визначається коефіцієнтом $\rho = M_n / M_\phi$ (де M_n – число елементів кадрі, спотвореному завадами, не співпадаючими з відповідними елементами еталонного (вихідного) зображення; M_ϕ – число елементів відфільтрованого зображення, що не співпадає з елементами еталонного).

Анізотропна фільтрація. Найбільше застосування для просторової фільтрації зображень отримала анізотропна фільтрація. Дискретна інтерпретація останнього приводить до співвідношення

$$\widetilde{a}_{ij} = \Lambda\left[\sum_{v=-N_{a/2}}^{N_{a/2}}\sum_{\xi=-N_{a/2}}^{N_{a/2}} a_{i+v,j+\xi}\omega_{v\xi} - \eta\right], \qquad (1.8)$$

де $\widetilde{a}_{ij}$ – елемент матриці відфільтрованого зображення, яке знаходиться на перетині i-тої стрічки і j-го стовпчика ;

$a_{i+v,j+\xi}$ – елемент матриці зображення, спотвореного завадами, який розміщений на перетині $(i+v)$-ї стрічки і $(j+\xi)$- го стовпчика;

$\omega_{v\xi}$ – елемент апертури, який представляє собою матрицю розміром $N_a \times N_a$ (N_a, як правило, являється непарним числом), що знаходиться на перетині v-ї стрічки і ξ-го стовпчика;

η – поріг фільтрації, який являється константою;

Λ – порогова функція.

Для повної фільтрації матрицю зображення необхідно симетрично доповнити елементами, які дорівнюють нулю, так щоб результуючий розмір її став рівним $(N+N_a)\times(N+N_a)$ елементів (N – розмір зображення в елементах). При $\omega_{v\xi} = const$ має місце алгоритм усереднення. Крім того, якщо фільтрується багатоградаційне зображення, то необхідно ввести множину порогів (потужність множини рівна числу градацій яскравості) і присвоїти елементу $\widetilde{a}_{ij}$ значення, відповідне максимальному порогу.

Анізотропна фільтрація послаблює вплив окремих плям, які не відносяться до силуету об’єкта. Доведено, що вона забезпечує ефективну фільтрацію зображень, які спотворені адитивним нормальним шумом.

Виділення контурів зображень. Ця операція, іноді називається також операцією одержання графічного препарату або підкреслення границь [9],

дозволяє видозмінити зображення так, щоб контрастно виділити окремі області зображення і їх границі. Підкреслення здійснюють підвищенням яскравості зображення в цих точках або підфарбовуванням його у визначені кольори. Обробка зводиться до виділення перепадів яскравості у вихідному зображенні.

Як видно з аналізу існуючих методів електронізації текстових документів шляхом розпізнавання графічного зображення, вони вимагають великих часових затрат та не досить надійні. З лінгвістичної точки зору будь-який текстовий документ можна розглядати як деякий носій мовної інформації, що використовується для її передачі в тій чи іншій комунікативній системі [20-27]. З цієї точки зору зображення тексту опосередкованим чином відображає різні інформаційні складові, присутні комунікативному акті: прагматичний, семантичний, лексичний, морфологічний, сигматичний і афективний [21]. Виникає задача – в якій послідовності потрібно використовувати інформацію того чи іншого рівня в автоматизованому процесі введення і розпізнавання текстового документа, щоб отримати максимально можливу швидкість і мінімально можливі помилки і вартість [28]. Для розв’язання цього питання в даній роботі пропонується нова технологія обробки текстових документів, яка передбачає використовувати часткове розуміння тексту під час розпізнаванням графічних образів. Для цього вона використовує низку мовних складових інформації - лексичної, морфологічної, синтаксичної та інш. на етапі введення і розпізнавання поряд з графічним зображенням тексту. Ці виділені в графіччному зображенні складові дозволяють здійснити його часткове розуміння, а також оптимально розподілити процес обробки документа між пристроєм введення і комп’ютерною системою.

Для реалізації даної технології в роботі вирішуються задачі оцінки інформативності окремих ознак того чи іншого виду інформації, вибору критерію оцінки ефективності процедури електронізації тексту, розробки методів, агоритмів і програмного забезпечення автоматизації створення бази даних морфем української мови, які є елементами розпізнавання в запропонованому підході.

1.3 Задачі дослідження

Проблема автоматизованого ефективного ввводу та обробки текстової інформації є важливою і актуальною. Існуючі моделі та методи електронізації текстів не дозволяють розв’язати всі задачі в рамках зазначеної проблеми. Зокрема, існуючі моделі використовують лише графічну інформацію для опису зображень тексту, такий підхід перешкоджає отримати ефективну процедуру з точки зору швидкості і надійності, що викликає необхідність використання лінгвістичної інформації, присутньої текстовому документі. Недостатньо ефективно також працюють й існуючі алгоритми сегментації текстових даних, що особливо гостро, з точки зору надійності, відображається на якості обробки зашумлених зображень текстів та текстів, надрукованих нетипізованими шрифтами.

Метою роботи є підвищення якості автоматизованого введення та обробки текстової інформації. Для досягнення мети роботи ставляться наступні задачі дослідження:

- розробити метод вибору критеріїв ефективності та математичної моделі оптимізації процесу автоматизованого введення і обробки текстів в інформаційних системах;
- розробити структурні моделі інформаційного тезаурусу мови;
- розробити метод ознакового опису інформативних елементів тексту, що підвищують надійність та швидкість їх сегментації і розпізнавання;
- розробити інформаційну технології введення і обробки текстових документів для їх електронізації, шо включає розробку алгоритмів та програмного забезпечення для введення і обробки текстових докуметів;

1.4 Варіантний аналіз методів електронізації текстових документів

Формування адекватних критеріїв ефективності можна здійснити на основі рекомендацій, що відображують надбаний досвід системного проектування та дослідження операцій. Правильно вибрані критерії ефективності повинні задовольняти низці вимог, основними з яких є:

- відповідність цільовому призначенню системи;
- критичність до параметрів, що визначають якість функціонування системи;
- достатньо проста обчислюваність;
- нормованість;
- універсальність (можливість використання критерію для порівняльної оцінки будь-якої пари систем з деякого класу систем).

Особливістю систем автоматичного розпізнавання символів, як і інших складних систем, є необхідність їх оцінки по багатьох часткових показниках якості, з яких найбільш часто вживаними є наступні [29,30,31]:

а) E - точність розпізнавання;

б) t_r - час розпізнавання;

в) O - обсяг потрібної пам'яті;

г) τ_r - час приведення системи в готовність;

д) C - вартість системи.

Показники 2, 3, 4 з визначеними вагами входять в критерій вартості C, тому їх можна вилучити з списку часткових критеріїв. Таким чином, основними частковими критеріями для оцінки ефективності є точність розпізнавання та вартість. Використання їх утрудняється тими обставинами, що різні системи вирішують різні задачі і працюють в різних умовах. Це не дозволяє провести порівняльного аналізу якості систем автоматичного розпізнавання, що вирішують різні задачі розпізнавання [28]. Тому в роботі як загальносистемний критерій

використовується узагальнений функціонально-статистичний критерій, описаний в розділі 2.

1.5 Обґрунтування та вибір базового варіанту

Використання традиційних технологій електронізації документів в текстових форматах, які дозволяють представити символи в ASCII кодах, і, таким чином, автоматизувати їх аналіз, передбачає посимвольне розпізнавання графічного зображення тексту за допомогою наявних програмних засобів (наприклад, FineReader). Однак такі технології в своїх історичних витоках орієнтовані на брак апаратних ресурсів (швидкодії і пам’яті), не враховують технічних можливостей сучасних обчислювальних систем і мікропроцесорних засобів, а також не використовують мовних складових в інформаційній ієрархії текстового документа. Для уникнення даних недоліків в даній магістерській роботі пропонується інформаційна технологія електронізації текстів, яка передбачає їх введення і опрацювання за допомогою розподіленої комп’ютерно-мікропроцесорної системи [32,33,34].

Під час реалізації цієї технології однією з проблем, які потрібно розв´язати, є автоматизація процесу побудови бази даних морфем української мови, які використовуються в даній технології в якості окремих елементів розпізнавання.

Таблиця 1.1 – Порівняння технічних характеристик нового методу та аналога

Технічні показники	Новий метод	Аналог	Відношення
Точність, %	99,8	75	1,3
Надійність, %	99	70	1,4
Величина затрати часу, хв	5	90	18

Отже, з таблиці видно, що по всім технічним характеристикам новий розроблюваний метод має набагато кращі характеристики, що свідчить про доцільність його подальшої розробки та впровадження.

1.6 Прогноз величини попиту

Для виходу нового товару на ринок необхідно визначити попит на нього. Споживачами в такому випадку можна очікувати державні та приватні підприємства, галузі зв'язку та телекомунікації, промислові підприємства, підприємства транспорту, військові та оборонні об'єкти України, для яких важливим швидко та точно розпізнавати нестандартні символи. На Україні існує близько 600 таких закладів.

Встановимо приблизну кількість споживачів, які не купуватимуть нову розробку по різним причинам:

- приблизно 20% споживачів не купуватимуть через відсутність інформації про нове програмне забезпечення;
- ще приблизно 20% не купуватимуть через відсутність потреби ній;
- біля 10% закладів будуть користуватися старими методами через те, що звикли і не хочуть переходити на нові методи.

Тому, дану розробку можуть купити:

$$100\% - (20\%-20\%-10\%)=50\% \text{ (закладів)},$$

а саме:

$$600 \cdot 0{,}5=300 \text{ (закладів)}.$$

Середній термін, через який може вийти чи розробитись ще більш вдосконалене ПЗ становить 1-3 роки. Тому щорічно потенційним покупцям нових розробок можуть бути (так званий оптимістичний прогноз):

$$ОП= \frac{300}{1...3} = 100...300 \text{ (закладів).}$$

Нехай в даній географічній зоні, якій робимо прогноз, не всі заклади потребують дане ПЗ, а, наприклад, 50%. Окрім цього, не всі заклади, навіть якщо в них є потреба будуть його купувати, нехай це становитиме 20%. Тоді песимістичний обсяг ринку ПО програмного забезпечення буде:

$$ПО= (100...300) \cdot 0,5 \cdot 0,2 = 10...30 \text{ (шт.).}$$

Реалістичний прогноз РП може бути визначений як середнє арифметичне оптимістичного та песимістичного прогнозів, тобто:

$$РП= \frac{(100...300) + (10...30)}{2} = 55...165 \text{ (шт.).}$$

Таким чином, реалістичний прогноз попиту на розроблене ПЗ в даному регіоні України буде складати приблизно 150 дисків з ПЗ за рік.

1.7 Прогноз цінової політики на нову розробку

Вартість – один з найвагоміших факторів, який враховується при виборі програмного забезпечення. При прогнозуванні цінової політики на нове ПЗ будемо врахувати ряд таких факторів:

а) Стан конкуренції на ринку у проблемі розпізнавання мови.

Стан конкуренції по розпізнаванню мови на сьогоднішній день великий, тому що багато навчальних закладів, спеціальних фірм, підприємств займаються цією проблемою. Але процес розпізнавання мови дуже громіздкий та складний, тому цю проблему зазвичай вирішують по частинам. Але саме по виділенню

інформативних ознак конкурентів на Україні мало і тому ціну на новий метод і програмний продукт можна класти будь-яку, адже конкурентів немає.

б) Можливість покращення основних технічних показників у порівнянні з аналогами.

Оскільки згідно пункту 1.5 даного розділу всі технічні показники є кращими, ніж у аналога, тому вартість розробки можна суттєво підвищувати, адже робота з аналогом – це дуже громіздка робота і мало хто буде ним користуватися, якщо вийде новий метод і ПЗ.

в) Залежність величини попиту від ціни розробки.

Оскільки розроблене ПЗ буде потрібне усім закладам, підприємствам, фірмам, які займаються проблемою розпізнавання мови, а також враховуючи ціну аналогу – 4160 грн. То дане ПЗ є необхідним для них, оскільки ціна запропонованого мною програмного продукту є нижчою, тому попит на нього буде.

1.8 Визначення конкурентів

На сьогоднішній день про існування солідних українських фірм, які займаються створенням програмних продуктів на основі розпізнавання або синтезу мовлення невідомо. Існують або наукові заклади, які займаються дослідженнями в галузі розпізнавання та синтезу, або самотні розробники. З наукових закладів, що більш-менш серйозно займаються вказаними проблемами, можна виділити:

- відділ розпізнавання та синтезу звукових образів Міжнародного науково-навчального центру інформаційних технологій та систем (МННЦІТС), м. Київ, та Українську асоціацію з оброблення інформації та розпізнавання образів (УАСОІРО) при ньому.
- відділ розпізнавання мовних образів Державного інституту штучного інтелекту (ДІШІ).

Можливо, ці конкуренти теж розроблятимуть систему автоматичного виділення інформативних ознак, але поки що таких дій від них немає. Тому якщо вийде новий метод вирішення цієї проблеми, що розробляється в даній дипломній роботі, то навряд чи вони почнуть теж цією проблемою займатися, якщо вона вже буде вирішена. Тому дані конкуренти не заважають. А це означає, що попит на даний метод і ПЗ буде великим.

2 РОЗРОБКА ЕФЕКТИВНИХ МЕТОДІВ І МОДЕЛЕЙ ЕЛЕКТРОНІЗАЦІЇ ТЕКСТОВИХ ДОКУМЕНТІВ

2.1 Вибір критеріїв ефективності систем розпізнавання

Задача побудови оптимальної стратегії розпізнавання згідно загальносистемного підходу повинна пов'язуватися не тільки з умовою досягнення заданої точності розпізнавання, а й з ефективністю роботи системи в цілому. На сьогодні повна формалізація рішення вказаної задачі не є відомою для загального випадку і, мабуть, неможлива [30,31]. Процедуру побудови ефективної стратегії розпізнавання можна формалізувати тільки в тому разі, в якому неформальними методами попередньо вибрано апріорний алфавіт ознак образів, а також вибрано критерії для оцінки ефективності систем розпізнавання. В такому випадку формальна постановка задачі побудови оптимальної процедури розпізнавання може бути сформульована у вигляді задача пошуку оптимального по загальносистемному критерію дерева рішень, в якому на кожному кроці класифікації з апріорного алфавіту вибирається підмножина ознак, що максимально зменшує на досягнутому кроці ентропію про образ і збільшує швидкість класифікації.

Формування адекватних критеріїв ефективності можна здійснити на основі рекомендацій, що відображують надбаний досвід системного проектування та дослідження операцій [36-41]. Правильно вибрані критерії ефективності повинні відповідати таким вимогам:

- повинен враховувати ціль функціонування системи;
- бути критичним (чутливим) до досліджуваних параметрів системи;
- бути однозначним і просто обчислюваним;
- нормованість в заданому діапазоні значень (наприклад, від 0 до 1);
- можливість використання критерію для порівняння будь-якої пари систем заданого класу.

Особливістю систем автоматичного розпізнавання мови (САРМ) і систем оптичного розпізнавання тексту і символів (ОРС) , як і інших складних систем, є необхідність їх оцінки по багатьох часткових показниках якості, з яких найбільш часто вживаними є наступні:

а) E – точність розпізнавання;

б) t_r – час розпізнавання;

в) O – обсяг потрібної пам'яті;

г) τ_r – час приведення системи в готовність;

д) C – вартість системи.

Показники б), в), д) з заданими коефіцієнтами входять в критерій вартості С, тому їх можна не включати до списку часткових критеріїв. Таким чином, основними частковими критеріями для оцінки ефективності САР є точність розпізнавання та вартість. Використання їх унеможливлюється тими обставинами, що різні системи працюють в різних умовах і вирішують різні задачі. Це не дозволяє провести порівняльного аналізу якості САР, що вирішують різні задачі розпізнавання. Тому в роботі як загальносистемний критерій використовується узагальнений функціонально-статистичний критерій [42], модифікований належним чином до системи мовного спілкування шляхом належного вибору потенціальної і реальної САР [28]:

$$Э = \frac{Э_Р}{Э_П}\Bigg|_{E = E_\partial}, \qquad (2.1)$$

де $Э_Р$ і $Э_П$ – функціонально-статистичні критерії для реальної і потенціальної САР відповідно:

$$Э_Р = \frac{I_Р}{C_Р}, \qquad (2.2)$$

$$Э_{П} = \frac{I_{П}}{C_{П}}; \quad (2.3)$$

E_{∂} – задана в технічному завданні точність розпізнавання;

$I_{Р}, I_{П}$ – кількість інформації, яку дістає реальна і потенціальна системи відповідно;

$C_{Р}, C_{П}$ – вартість реальної і потенціальної систем відповідно.

В якості моделі потенційної системи вибрано модель біологічної системи розпізнавання слів і зображень людини, оскільки вона використовує найбільш ефективну систему ознак і оптимальні способи їх зображення, а також оптимальні процедури розпізнавання. Ця система є ідеальною ціллю функціонування декодера інформації з мовного сигналу чи тексту для заданого словника і рівня оточуючих завад. Кількість інформації, яку дістає потенціальна система розпізнавання, дорівнює:

$$I_{П} = H_{П}(W) - H_{П}(W / W^{*}), \quad (2.4)$$

де $H_{П}(W)$ – ентропія повідомлення до його передавання, зв'язана з апріорною ймовірністю окремого повідомлення $P(w_i)$ наступним виразом [43,44]:

$$H_{П}(W) = \sum_{i=1}^{|W|} P(w_i) * \log_2 P(w_i). \quad (2.5)$$

Апріорна ймовірність $P(w_i)$ визначається або статистичним шляхом, або з припущення рівної ймовірності повідомлень, тоді

$$P(w_i) = \frac{1}{|W|}, \quad (2.6)$$

де $|W|$ – кількість повідомлень в словнику W;

$\mathrm{H}_{\Pi}(W/W^{*})$ – ентропія повідомлення після його прийому, яка характеризує втрату інформації на одне повідомлення:

$$\mathrm{H}_{\Pi}(W/W^{*}) = -\sum_{w_i=W}\sum_{w_j=W} p(w_i/w_j) * p(w_j) * \log_2 \frac{p(w_i) * p(w_i/w_j)}{\sum\limits_{w_j} p(w_i/w_j) * p(w_i)}. \quad (2.7)$$

Величини $p(w_i/w_j)$, що характеризують подібність слів, можуть бути визначені експериментально для різних рівнів завад за результатами розпізнавання слів і символів людиною.

На етапі проектування за вартість C_{P} використовується величина, що пов'язана пропорційною залежністю зі складністю процесу розпізнавання:

$$C_{\mathrm{P}} = C_{\mathrm{X}} + C_{\mathrm{K}}, \quad (2.8)$$

де C_{X} – складність обчислення ознакового опису образів;

C_{K} - складність обчислень класифікації образів.

Критерій (2.1) відповідає перерахованим вище вимогам, яким повинні задовольняти показники якості роботи систем.

2.2 Математична модель процесу розпізнавання тексту

Формальна постановка задачі побудови оптимальної стратегії розпізнавання тексту може бути подана в наступному вигляді [29]:

$$\tilde{S}_{Gopt} = \arg u \max \Im(\tilde{S}_{Gi})\Big|_{\{\tilde{S}_{Gi} \in \tilde{S}_G, W_d, r_d, E_d\}}, \quad (2.9)$$

де $\tilde{S}_{Gi}$ – окрема стратегія розпізнавання в замкнутій відносно доступної інформації множині стратегій розпізнавання $\tilde{S}_G$;

W_d, r_d, E_d – відповідно алфавіт образів, рівень завад і точність розпізнавання апріорно відомі в поставленій задачі розпізнавання.

Оскільки обчислювальна складність C_p критерію ефективності (2.1) є функцією двох змінних - обчислювальних затрат на процедуру класифікації C_k та затрат на обчислення ознакового опису C_x, то розв'язання задачі оптимізації, сформульованої у вигляді (2.9)є досить складним. Спростити її можна шляхом декомпозиції цих змінних в критерії (2.1), зобразивши стратегію розпізнавання в вигляді покрокової процедури класифікації на дереві рішень [28,30].

Оскільки часто використовувані процедури статистичного розпізнавання в n-вимірному просторі ознак можна подати у вигляді дерева класифікації з одним кореневим вузлом, а всіма іншими – термінальними, то використання ієрархічної послідовно-паралельної стратегії розпізнавання в вигляді дерева рішень в оптимізаційній процедурі побудови ефективної стратегії розпізнавання образів не виключає узагальненого підходу. В такому дереві кількість кроків класифікації (гілок) дорівнює кількості термінальних вузлів (образів, що розпізнаються).

В системі розпізнавання тексту складність C_p є сумою складностей C_i кожного з ієрархічних рівнів розпізнавання, інформативність I_P є зростаючою функцією ймовірності правильного розпізнавання, тому оптимальуа стратегію отримують композицією алгоритмів, що максимізують на кожному з рівнів дерева відношення $\frac{I_i}{C_i}$. В оптимальній стратегії послідовність композиції алгоритмів повинна відповідати послідовності розміщення рівнів дерева класифікації, і на кожному рівні ознаки повинні вибиратися з умови їх мінімальної складності [30]:

$$\tilde{S}_{Gopt} = A_1(\tilde{I}(S^1)) \otimes A_2(\tilde{I}(S^2)) \otimes \cdots \otimes A_W(\tilde{I}(W\ )), \qquad (2.10)$$

$$\tilde{I}(S^i) = x^i_{opt}, x^i_{opt} = \arg u \min C_i(x^i_l), \quad (2.11)$$

де $\tilde{I}(W)$ – параметричний опис еталонів класів;

$C_i(x^i_l)$ – складність обчислення l-ї ознаки, яка використовується i-му рівні для опису елементів розпізнавання;

$A_h(\tilde{I}(S^h))$ – алгоритм попередньої класифікації образів тексту на групи, а $\tilde{I}(S^h))$ - ознаковий опис образів S^h, що класифікуються на даному рівні дерева.

Для того, щоб сумісно оптимізувати дерево класифікації відносно точності, швидкості і вартості, необхідно, як було досліджено в роботі [30], використовувати більш інформативні ознаки на верхніх рівнях дерева.

На мікропроцесорний пристрій введення тексту оптимальним розподілом процедури розпізнавання передається виконання тих функцій обробки графічного зображення , що лежать не нижче рівня h_p:

$$h_p = arg(C_{max}), \quad (2.12)$$

де h_p – той номер рівня дерева класифікації, на якому складність обчислень досягає граничних апаратних і обчислювальних ресурсів C_{max} мікропроцесорного пристрою.

Для створення інформаційної моделі текстового документу розпізнавання розглядається як процес передачі повідомлення, зашифрованого текстом вхідної мови, по каналу з шумами [44].

В вибраному підході одним із компонентів в даному підході є набір речень на правильного тексту (вхідний текст). Будемо вважати їх реченнями мови $L(G)$, яка генерується граматикою G. Шуми каналу відповідають огріхам в розпізнаванні. В такому випадку якість розпізнавання в даній моделі можна оцінити вибраним критерієм (2.1).

$$E = \frac{I_{MT}}{I_P}, \qquad (2.13)$$

де I_{MT} – інформативність розпізнаного САР тексту;

I_P – інформативність вхідного тексту.

Вони можуть бути визначені з таких виразів:

$$I_{MT} = H(L(G)) - H(L(G))_{MT}, \qquad (2.14)$$

$$I_p = H(L(G)) - H(L(G))_P, \qquad (2.15)$$

де $H(L(G))$ – ентропія мови;

$H(LG))_{MT}$ – ентропія тексту, розпізнаного системою автоматичного розпізнавання;

$H(L(G))_P$ – ентропія вхідного тексту.

Такий підхід, на відміну від запропонованого в роботі [45], дозволяє оцінити основний параметр якості розпізнавання – його інформативність. Для реалізації цього підходу необхідно визначити шляхи обчислення величин ентропії $H(L(G))$ і $H(LG))_{MT}$. З цією метою будемо розглядати мову $L(G)$ у вигляді діаграми переходів між станами без наявності в них петель. Діаграма переходів містить скінчений набір станів Q, скінчений словник V і функцію переходів δ, яка встановлює граматичні обмеження:

$$\delta : (Q \times V) \to Q, \qquad (2.16)$$

тобто кожен перехід на діаграмі має вигляд:

$$\delta(q_i, v) = q_j. \qquad (2.17)$$

При цьому стани $q_i \in Q$ вказують на порядок слів, а $v(q_{ij})$ – самі слова. Правильне речення w складається з міток всіх дуг, що проходять послідовно вздовж будь-якого шляху з початком в q_1 і кінцевим станом $q_l \in Z$, де Z – множина кінцевих станів.

Для характеристики діаграми переходів в якості визначальної кількісної величини використовується матриця зв'язності C, окремий елемент c_{ij} якої дорівнює кількості переходів (дуг) r від стану q_i до стану q_j:

$$c_{ij} = r, i, j = \overrightarrow{1, N}, \tag{2.18}$$

де r – число переходів з стану q_i в стан q_j.

Матриця зв'язності використовується для обчислення кількості речень N_k довжиною k в мові $L(G)$ з виразу:

$$N_k = \vec{e}_1 C^k \vec{f}^T, \tag{2.19}$$

де $\vec{e}_1$ представляє перші N компонент одиничного вектора;

$\vec{f}$ – N- компонентний вектор одиниць або нулів, i-та компонента якого дорівнює 1, якщо q_i є кінцевим станом.

Оскільки

$$\sum_{k=0}^{\infty} C^k = (I - C)^{-1}, \tag{2.20}$$

де I – одинична матриця розміром $N \times N$, то загальна кількість речень в мові $|L(G)|$ дорівнює

$$|L(G)| = \vec{e}_1 (I - C)^{-1} \vec{f}^{T} \tag{2.21}$$

і середня довжина речення:

$$|\bar{W}| = \frac{\sum_k k N_k}{|L(G)|}. \tag{2.22}$$

При умові $|L(G)| < \infty$ і рівноймовірності речень можна розрахувати N_k, $|L(G)|$, $|\bar{W}|$ і ентропію мови:

$$H(L(G)) = \frac{\log_2 |L(G)|}{|\bar{W}|}. \tag{2.23}$$

Оскільки здебільшого умова рівноймовірності речень не виконується, то для їх довільного імовірнісного розподілу максимальну ентропію отримують як:

$$H_{\max}(L(G)) = -\log_2(x_0), \tag{2.24}$$

де x_0 обрано з того розрахунку, щоб $1 - \sum_k N_k x_0^{\,k} = 0$.

Визначивши ентропію мови, можна визначити і ентропію тексту $H(LG))_{MT}$, отриманого після розпізнавання САР. Для цього система автоматичного розпізнавання тексту представляється у вигляді каналу передачі інформації з шумами (рисунок 2.1).

Для такого каналу основною характеристикою є втрата інформації на одне повідомлення (речення) $H(w/w^*)$, яка є мірою невизначеності розпізнавання

речення w в речення w^*. Звідси ентропію перекладу $H(LG))_{MT}$, розпізнаного САР, можна отримати з виразу [42]:

$$H(L(G))_{MT} = |L(G)|\, H(w/w^*), \tag{2.25}$$

тобто втрата інформації в розпізнаному тексті дорівнює добутку ентропії на одне речення на середню кількість речень в тексті:

$$\mathrm{H}(w/w^*) = -\sum P(w_i/w_i^*)\cdot \log P(w_i/w_i^*). \tag{2.26}$$

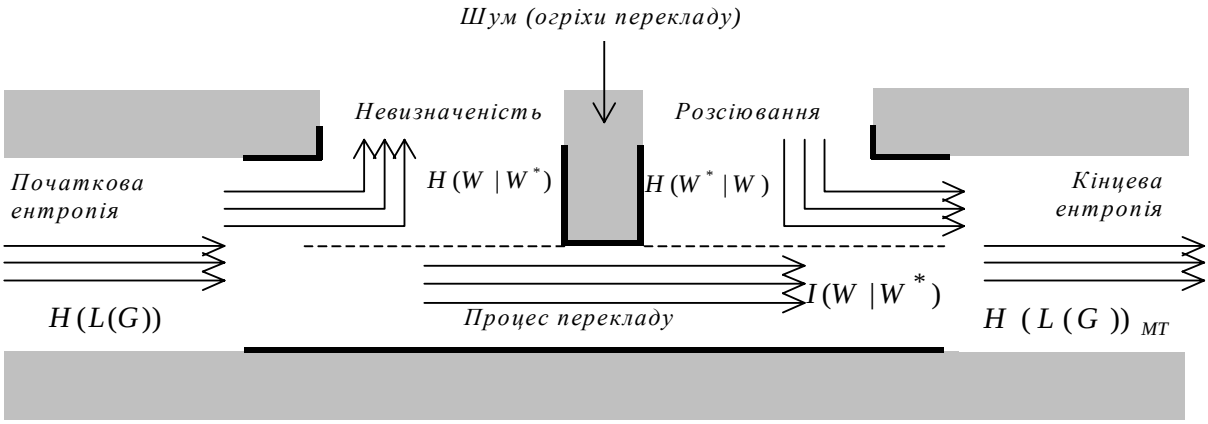


Рисунок 2.1 – Представлення процесу розпізнавання тексту каналом з шумами

Величини $p(w_i/w_i^*)$, що характеризують подібність речень, можуть бути визначені з формули [29]:

$$p(w_i/w_i^*) = \frac{\tilde{\mu}_i \cdot 10^{-d_{ii}^w}}{\sum_{k=1}^{|w|} \tilde{\mu}_k \cdot 10^{-d_{ii}^w}}, \tag{2.27}$$

де d_{ii}^w – модифікована відстань Левенштейна між i-им еталонним реченням і i-тим перекладеним реченнями тексту, яка визначається як сумарна кількість

пропусків, замін, вставок та змін порядку слів, яку потрібно зробити у перекладеному реченні, щоб отримати еталонне, а $\tilde{\mu}_k$ – середня величина відстані між реченнями.

2.3 Інформаційні характеристики текстового повідомлення

Комунікативна система, в якій передається зашифрована текстовим документом інформація, включає в загальному такі компоненти (рисунок 2.2).

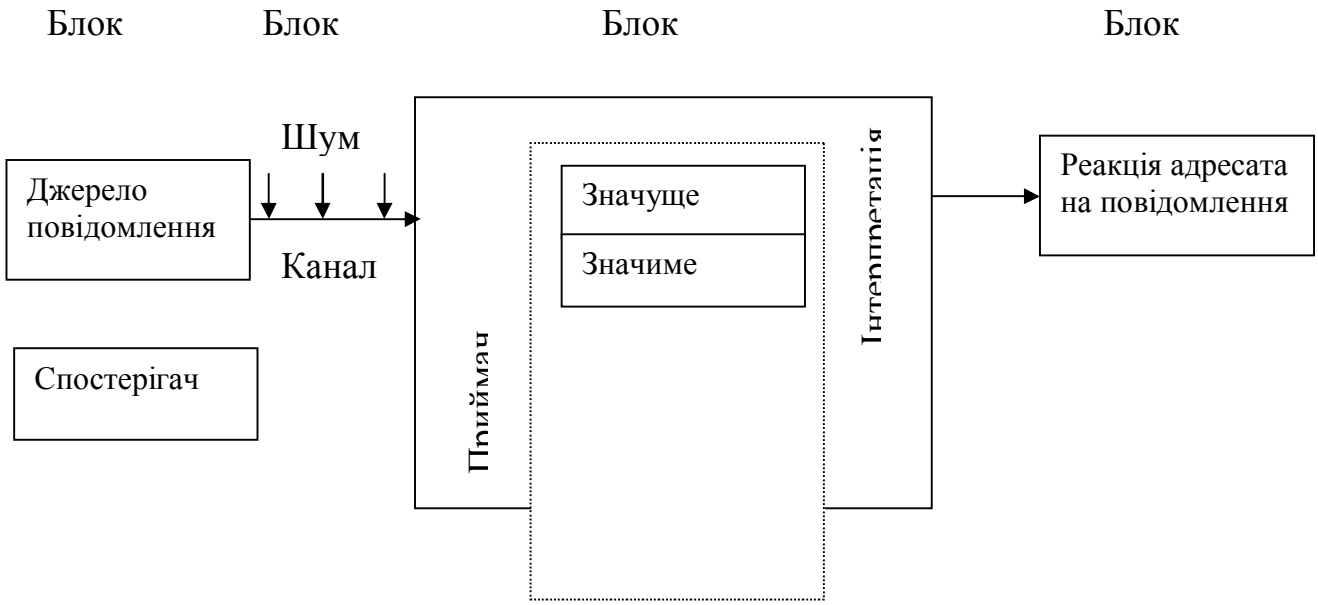


Рисунок 2.2 – Схема передачі інформації текстовим документом

Рисунок 2.2 складається з наступних елементів:

а) блок 1 – джерело повідомлення;

б) блок 2 – канал передачі з шумом;

в) блок 3 – знак, компонентами якого є:

- значуще (ім’я, носій інформації);
- значиме, яке містить денотат (відображення предмета зовнішнього світу), десигнат (поняття про предмет, зміст імені), контекст (комплекс чуттєво-оцінних, в тому числі естетичних відтінків значення);
- приймач повідомлення;
- інтерпретація повідомлення в приймачі, тобто готовність останнього реагувати на повідомлення;
- інтерпретація повідомлення в приймачі, тобто готовність останнього реагувати на повідомлення;
- контекст, тобто оточення, в якому здійснюється інтерпретація повідомлення;

г) блок 4 – реалізація реакції адресата на розпізнане і проінтерпретоване повідомлення;

д) блок 5 – зовнішній спостерігач (мета спостерігач).

Наявність компонентів 1-5 є обов’язковою умовою для здійснення інформаційного процесу. Наприклад, за умови відсутності знака або інтерпретаційної реакції приймача вплив джерела на приймач може мати тільки енергетичний характер; відсутність каналу зв’язку взагалі знімає питання про взаємодію джерела і приймача. Що ж стосується контексту і зовнішнього спостерігача, то їх присутність взагалі не є обов’язковою під час реалізації інформаційного процесу передачі повідомлення текстовим документом.

Інформація, з якою мають справу під час розв’язання проблем, пов’язаних з обробкою текстових документів (наприклад, їх розпізнаванням чи розумінням), в залежності від того, до яких аспектів знака відноситься інформаційна міра, поділяється на такі види:

а) прагматична інформація, що визначає цінність змісту повідомлення для досягнення приймачем повідомлення поставленої ним в даний момент

мети, і з точки зору вибору того рішення, яке є найкращим для досягнення цієї мети [29];

б) семантична інформація, яка оцінює кількісну характеристику змісту в повідомленні відносно тезаурусу приймача;

в) сигматична (лексична) інформація, що характеризує здатність елементів повідомлення поєднуватися між собою;

г) синтаксична інформація, визначає невизначеність елементів повідомлення зі зміною ступеня обізнаності приймача і оцінюється тими обмеженнями, які накладаються на комбінаторні властивості і частоту знаків [5];

д) морфологічна інформація, яка визначає взаємозв'язок і відношення між установленими лінгвістикою окремими морфами слова;

е) графічна інформація, що міститься в зображенні символів, з яких складаються слова тексту.

Задача розробника технології електронізації тексту полягає, в першу чергу, в розробці процедури, яка б визначала послідовність і кількість використання інформації того чи іншого із названих ієрархічних рівнів [29]. Для цього необхідно оцінити інформативність різних складових текстового документа, що підлягає розпізнаванню і розумінню. Дана оцінка виконується в наступних розділах шляхом проведення статистичних дослідів над репрезентативними наборами текстів.

2.4 Статистичні характеристики текстових образів на лексичному рівні

Експеримент по вгадуванню дає можливість оцінити осмислену інформацію, що знаходиться в повідомленні. Ще раз нагадуємо, що ця осмислена інформація кількісно оцінюється не шеннонінвською мірою інформації цього ж повідомлення, а різницею значень ентропії, що падають на наступну за даним

повідомленням ділянку тексту (одне значення отримане за умови, що зміст повідомлення невідомо, а друге – що цей зміст уже відомо угадувачу) [13,22,24].

Тепер подивимося, що являє собою оцінювана в такий спосіб інформація. Для цього звернемося до умов підбору текстового матеріалу й вибору вгадувача.

Для вгадування використовуються тексти, що містять істині й осмислені пропозиції. Тому інформація *I* має семантичну природу. Наш вгадувач не тільки ідеально володіє системою й нормою мови, але є за умовою знавцем того спеціального предмета, що трактується в текстах, який угадується. Тому інформація, що отримується ним з тексту, оптимальним чином використовується для вгадування наступних ділянок тексту. Звідси слідує, що інформації *I* властивий і прагматичний аспект. Таким чином, оцінювана за допомогою величини *I* осмислена інформація є семантико-прагматичною та сигматичною інформацією. Ми будемо називати її смисловою інформацією [25].

Як уже говорилося, кількість інформації в тезаурусі й тексті не залишається незмінним.

Починаючи вгадувати текст, зразковий інформант оперує тезаурусом, що містить задану смислову інформацію. При цьому невизначеність початкової ділянки (у нашому випадку початкової букви) для нашого вгадувача дорівнює H_1. Довідавшись першу букву тексту, вгадувач збільшує кількість інформації в тезаурусі, переводячи його в стан I_1, угадування другого, третього й т.д. ділянок тексту ще більше насичує тезаурус.

У міру нагромадження тезаурусом відомостей про текст невизначеність при вгадуванні букв послідовно зменшується так, що

$$H_1 > H_2 > \ldots > H_{n-1} > H_n > \ldots > H_\infty. \qquad (2.28)$$

Пам'ятаючи, що $H=I$, вираз (2.28) можна переписати у вигляді:

$$I_1 > I_2 > \ldots > I_{n-1} > I_n > \ldots > I_\infty. \qquad (2.29)$$

Останній член нерівності (2.29) кількісно дорівнює тій інформації, що отримує вгадувач при відгадуванні ділянки (букви), як завгодно далеко віддаленого від початку тексту. Величину I_∞ ми будемо називати гранично синтаксичною інформацією зв'язного тексту.

Гранична інформація в ідеальній схемі тексту завжди буде більше нуля. Усякий текст, будучи утворений зі складних знаків (слів, словосполучень, пропозицій), що володіють практично необмеженою комбінаторною здатністю, має кілька продовжень або, інакше кажучи, завжди має невизначеність вибору. Навіть у тих випадках, коли даний крок конкретного тексту передбачає єдино можливе продовження, завжди знайдуться наступні кроки, які дадуть кілька можливих продовжень. Якщо ж розглядати деяку сукупність текстів, на основі якої будується усереднена схема, то поряд з текстами, що дають на даному кроці достовірне продовження, завжди виявляться інші тексти, які дадуть на цьому кроці кілька продовжень.

Величину I_∞ можна розглядати як сумарну оцінку статистичної інформації, одержуваної ідеальним носієм для даної мови або його різновиду від однієї букви зв’язного тексту, - букви, що перебуває досить далеко від початку тексту (у цьому випадку ідеальний носій уже враховує контекстні зв'язки повідомлення) [21].

Розглянемо ланцюги окремих значень I_n (відповідно $\overline{I}_n$ або $\underline{I}_n$), що оцінюють величини тої інформації, яку отримує вгадувач з тексту. Ці значення виявляють тенденцію до убування залежно від росту значень *n*. Представимо кожну із цих ланцюгів не як послідовності дискретних значень ентропії, а як неперервні криві (точніше, як неперервні функції неперервного аргументу). Якщо відволіктися на час від неперіодичних коливань, відносячи їх за рахунок розкиду, то ці ланцюги значень можуть бути апроксимовані в першому наближенні теоретичної показової кривої (експонентою) виду:

$$I_\xi = \left(I_0 - I_\infty\right)e^{-s\xi} + I_\infty, \qquad (2.30)$$

де I – верхні або нижній теоретичні межі інформації в даній ділянці тексту;

ξ – безперервний аргумент функції, що заміняє дискретні величини n;

$I_0=H_0$ – інформація алфавіту, а I_∞ – гранична інформація мови або його різновиду, величина якій служить асимптотою крива I_ξ;

e – основа натуральних логарифмів;

s – розраховується спеціально для кожної кривої контекстний коефіцієнт.

Замінивши величину $I_\xi = I_n$ в формулі (11) правою частиною виразу (2.29) і зробивши деякі спрощувальні перетворення, одержимо загальний вираз контекстної зв'язаності в даній ділянці тексту:

$$K_\xi = \left(I_0 - I_\infty\right)\left(1 - e^{s\xi}\right). \qquad (2.31)$$

По суті, для кожної мови або стилю ми маємо по дві формули що до виразів (2.30) і (2.31). Одну для $\underline{I}_\xi$, (відповідно $\overline{K}_\xi$), іншу – для $\overline{I}_\xi\left(\underline{K}_\xi\right)$ або $\overline{I}'_\xi\left(\underline{K}'_\xi\right)$. Фіксуємо в тексті, що угадується ділянку $\xi = A$. Тоді площа, обмежена кривою I_ξ, віссю OA, ординатами OI і AI ($\xi = A$), представляє загальну кількість синтаксичної інформації I_{OA}, отриманої зразковим вгадувачем з декодованої ним ділянки тексту OA (рисунок 2.3).

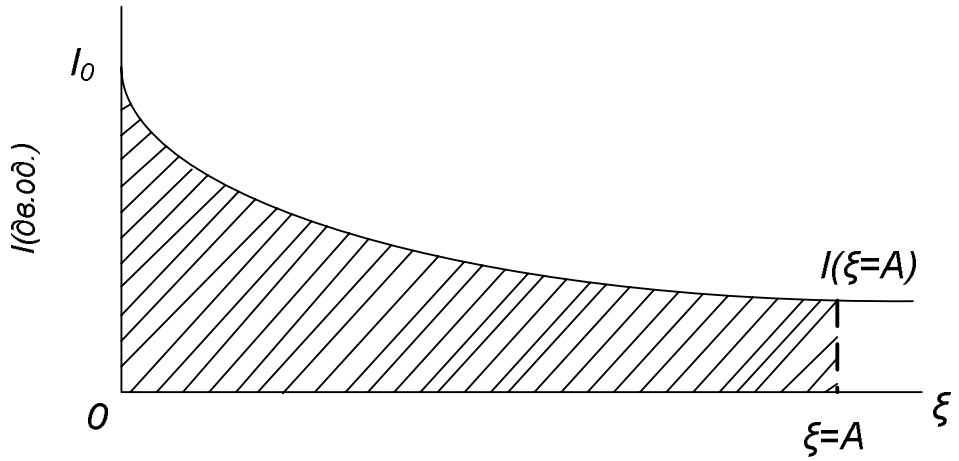


Рисунок 2.3 – Загальна кількість синтаксичної інформації, яка отримується з тексту на ділянці OA

Розрахунок цієї інформації можна здійснити за допомогою інтегрального виразу:

$$I_{OA} = \int_{0}^{\xi=A} I_{\xi} d\xi = \int_{0}^{\xi=A} \left|((I_0 - I_\infty)e^{-s\xi} + I_\infty)d\xi\right| = I_\infty A + \frac{I_0 - I_\infty}{S}\left(1 - e^{-sA}\right). \qquad (2.32)$$

Так, наприклад, якщо наш зразковий інформант прочитав або відгадав український діловий текст довжиною в 100 букв, то він витяг з нього, рахуючи по верхній границі інформації,

$$I_{OA} = 0.83 \cdot 100 + \frac{5 - 0.83}{0.24}\left(1 - e^{-0.24 \cdot 100}\right) = 100.8(\textit{біт})$$

синтаксичної інформації.

Звертаючись до загальної інтерпретації тільки що виведених залежностей, можна стверджувати, що крива I_{ξ} показує хід отримання з тексту статистичної й значеннєвої інформації. Крива K_{ξ}, відбиває динаміку взаємодії «механізмів» тезауруса вгадувача й інформації, що витягається з тексту [22,23].

У виразах (2.31) і (2.32) використовується контекстний коефіцієнт *s*, що виступає як показник швидкості зміни величин I_{ξ} і K_{ξ}. Чим більше величина *s*, тим скоріше йде збільшення значень K_{ξ}, (відповідно зменшення I_{ξ}). Інакше кажучи, коефіцієнт s є показником темпу росту контекстних зв'язків. Характерно, що найбільшу величину *s* дають тексти ділового стилю, у яких завдяки використанню великої кількості сталих словосполучень і обмеженого кола лексики контекстні зв'язки мовних одиниць установлюються швидше, ніж у текстах інших стилів (рисунок 2.4).

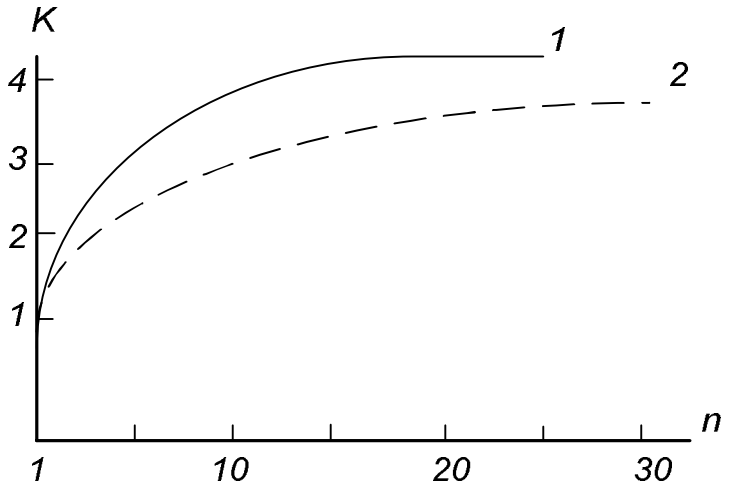


Рисунок 2.4 – Ріст контекстної обумовленості в ділових текстах і в змішаній вибірці

текстів російської мови:

1 – діловий текст;

2 – змішана вибірка

При побудові алгоритмів автоматичного аналізу тексту для нас важливими є не стільки сумарні дані, скільки роздільні оцінки інформаційної ваги окремих слів, лексико-математичних зв'язків між ними, морфології і т.п.

Таблиця 2.1 – Ділянка матриці, що використовується для розрахунку послівної схеми тексту (угадування початку українського тексту по скороченій програмі)

№ тексту	Стовбці											
	1	2	3	4	5	6	7	8	9	10	11	12
	Перше слово					пробіл	Друге слово					пробіл
	1-а буква	2-а буква	Середнє слово	Передостання буква	Остання буква		1-а буква	2-а буква	Середнє слово	Передостання буква	Остання буква	
1	a -	c 1	e 1	s 1	t 0	Δ 1	f 2	a 2	- -	p 1	t 0	Δ 1
2	Î -	- -	- -	- -	n 1	Δ 2	p 2	r 1	eze 1	n 1	t 1	Δ 1
3	v 1	- -	- -	- -	- -	Δ 2	a 1	- -	- -	- -	m 1	Δ 1

У зв'язку із цим спробуємо побудувати таку усереднену інформаційну схему, яка б відбивала розбиття тексту на слова (цю схему ми будемо називати лексичною схемою тексту).

Щоб побудувати таку схему, необхідно, по-перше, визначити середні довжини першого, другого й т.д. слів у текстах даної мови або стилю, по-друге, одержати схеми розподілу інформації для першого, другого, третього й т.д. слів усередненого тексту. Перше завдання вирішується звичайним статистичним шляхом, друга вимагає особливого групування того матеріалу, що був отриманий з експерименту по вгадуванню літер. Принципи цього групування ілюструє таблиця 2.1, що представляє собою ділянка матриці, по якій розраховувалася лексична схема українського тексту. У першому стовпці зібрані перші букви слів з кожного тексту. У другому стовпці вказуються другі букви перших слів за умови, що ці слова мають довжину не менш трьох букв. Для одно- і двохбуквених слів у другому стовпці ставиться прочерк. У четвертий стовпець попадають передостанні букви слів довжиною в чотири, п'ять і т.д. букв. Для коротких слів у четвертому стовпці ставиться прочерк. У п'ятому стовпці містяться останні букви слів, що мають довжину дві й більше букви. У шостий стовпець записуються всі пробіли між першим і другим словами. При кожній букві й пробілі вказується кількість спроб, що знадобилися для їхнього вгадування. У третій стовпець заносяться середини довгих слів без вказівки кількості спроб. У такий же спосіб розписуються друге й наступне слова тексту.

Аналогічна побудову має матриця для розрахунку лексичної схеми тексту, що відгадується по скороченій програмі. Її відмінність від першої матриці складається лише в тім, що вгадування букви із двох **і** більше спроб відзначається одним цифровим символом «2» (таблиця 2.2).

Таблиця 2.2 – Ділянка матриці, що використовується для розрахунку послівної схеми тексту (угадування українського тексту по повній програмі)

№ тексту	Стовбці											
	1	2	3	4	5	6	7	8	9	10	11	12
	Перше слово					пробіл	Друге слово					пробіл
	1-а	2-а	Середн	Передостання	Остання буква		1-а	2-а	Середн	Передостання	Остання буква	
1	м 8	н 3	ог	и 1	е 1	Δ 0	д 5	е 3	лег	т 1	ы 0	Δ 0
2	д 9	- -	-	- -	0 3	Δ 2	в 6	е 2	че	р 1	а 1	Δ 0
3	я 11	- -	-	- -	- -	Δ 1	п 1	е 2	-	-	л 1	Δ 1

Описана матриця містить розподіли числа спроб, що знадобилися для вгадування перших, других, передостанніх і останніх букв першого, другого, третього й т.д. слів тексту, а також аналогічні розподіли щодо пробілів між словами.

Обробка цих розподілів по формулах дає верхні й нижні оцінки інформації (ентропії) для перших двох і для останніх двох букв слів нашого усередненого тексту, а також оцінки інформації на пробілах між цими словами. Інформація, що падає на середини слів (порівн. стовбці 3 і 9) оцінюється як середнє арифметичне від інформації, що доводяться на першу, другу, передостанню й останню букви усередненого слова. У підсумку ми одержуємо схему розподілу інформації в тексті, що враховує його членування на слова. Отримана шляхом обробки результатів угадування по повній програмі схема російського тексту довжиною в 15 слів показана на рисунку 2.5.

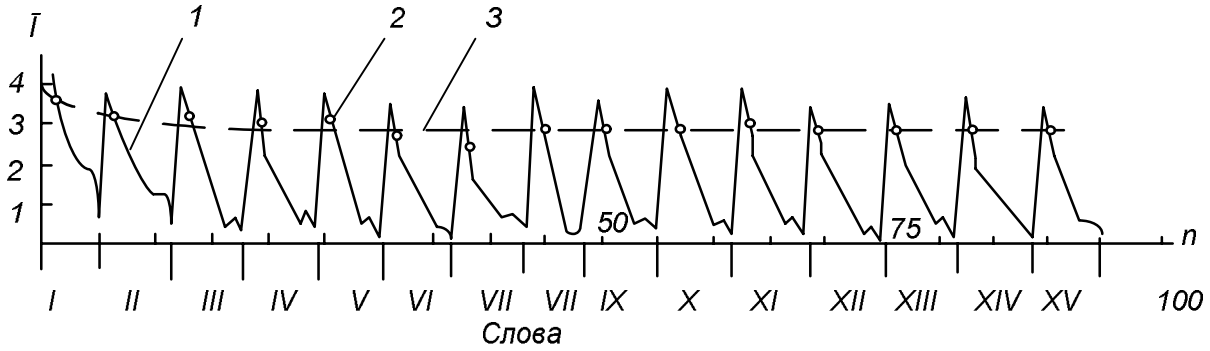


Рисунок 2.5 – Лексична схема усередненого російського тексту:

1 – полігон розподілу інформації з верхньої границі з урахуванням достовірних продовжень;

2 – значення $\frac{\bar{I}_1^c + \bar{I}_2^c}{2}$;

3 – графік значень $\frac{\bar{I}_1^c + \bar{I}_2^c}{2}$

Як показують дані дослідження мов, розподіл у тексті статистичної інформації має квантовий характер. Початок слів несуть максимуми інформації, у той час як середини й особливо наступні за ними пробіли виявляються або мало інформативними, або взагалі надлишковими.

Що стосується кінцевих букв, то вони несуть невелику кількість інформації.

Проаналізуємо, як змінюється розподіл інформації залежно від зміни довжини слова.

Розподіл інформації в коротких (до чотирьох букв), середніх (від п'яти до семи букв) словах, з одного боку, і довгих словах (від восьми букв і більше) – з іншого, має різний характер. Короткі й середні слова, як правило, дають монотонне убування інформації від початку слова до його кінця. Убування це звичайно відбувається гладко, і полігони (графічні схеми) цих слів мають компактний вигляд. Такий розподіл пов'язаний з тим, що серед коротких і довгих слів досить часто вживаються незмінні форми, які в підсумовуючій схемі згладжують максимуми інформації на кінцевих афіксах змінюваних слів.

У довгих словах (в першу чергу у позатекстових) полігони приймають поступово так звану *U*-образну форму; максимуми інформації зосереджені тут на початку й наприкінці словоформи, букви ж, що перебувають у середній частині словоформи, несуть найменшу інформацію. *U*-образний характер розподілу інформації в довгих словоформах відбиває певні ймовірносно-статистичні й лінгвістичні закономірності. Присутність максимуму інформації на початку слова відбиває той факт, що початкові букви письмового слова містять найбільшу кількість невизначеності у виборах.

2.5 Методи опису образів тексту на морфологічному рівні

У випадках, коли потрібно визначити розподіл інформації в слові на складовому й морфемному рівнях, результати вгадування групуються таким чином, щоб одержати розподіли частот спроб для вгадування першої й останньої букв складу або морфеми. Ця процедура, аналогічна перегрупуванню експериментального матеріалу при побудові лексичної схеми тексту, дає можливість не тільки визначити загальну кількість інформації, що припадає на склад або морфему, але дозволяє також чисельно оцінити інформацію на складових і морфемних границях, з'ясовуючи при цьому особливості складового й морфемного членування слова.

При побудові побуквених розподілів пробіл вважається останньою буквою слова. У складових схемах пробіл не враховується (як відомо, пробіл не входить до складу фонетичного складу). У морфемних схемах ураховуються лише ті пробіли, які є функцією нульових морфем [24].

Варто мати на увазі, що дослідження інформаційного членування слова може бути успішно здійснено лише за умови, що передуюче розрахункам виділення складів і морфем усередині слова проводиться на основі строгої й послідовної лінгвістичної процедури [25].

Побуквенне, складове і морфемне розподілення інформації в словах певної довжини відносяться лише до окремих випадків, не дають повного вигляду про загальний розподіл інформації й мало що говорять про загальні принципи складового й морфемного членування слова. Тому виникає необхідність побудувати узагальнюючу інформаційну схему слова на всіх трьох рівнях.

З огляду на емпіричні ймовірності появи слів певної довжини, можна звести їх побуквені схеми в узагальнюючу схему розподілу інформації в слові. Використовується два види такого узагальнення.

По-перше, може бути побудована схема, що відповідає середній довжині слова в даній мові. При побудові цієї схеми букви, або, точніше, інформації, що падають на ці букви, групуються так, як це робилося при розрахунку лексичної схеми тексту. Інформації ($I_n^{(c)}$), що припадають на першу, другу, передостанню, останню букви й пропуск в узагальнюючій схемі, одержуються як суми зважених інформацій, що припадають на відповідні букви в окремих схемах. Іншими словами

$$\overline{I}_n^{(c)} = \sum P_\lambda I_n^{(c)}, \qquad (2.33)$$

де P_λ – емпірична ймовірність слова певної довжини (λ);

$I_n^{(c)}$- інформація, що припадає на n-у букву в окремій схемі. Кількість інформації, що міститься в буквах, які перебувають в інтервалі між другою й передостанньою буквами усередненої схеми, виходить як середнє арифметичне першої, другої, передостанньої й останньої букв слова, помножене на кількість букв у цьому інтервалі [27].

Таблиця 2.3 – Розподіл інформації з $\bar{I}$ в узагальнюючій схемі українського текстового слова, що відповідає середній довжині слова

λ	P_λ	1-я буква $\bar{I}$	P_λ’	2-я буква $\bar{I}$	Середні букви (1,37 букви) $\bar{I}$	P_λ”	Передостання буква $\bar{I}$	P_λ”’	Остання буква $\bar{I}$	Пропуск $\bar{I}$
2	0,11	3,10	—	—		—	—	—		0,84
3	0,11	2,98	—	—		—	—	0,12	1,65	0,62
4	0,10	3,63	0,13	1,83	—	—	—	0,11	1,04	0,31
5	0,08	3,52	0,10	2,05		0,12	0,61	0,10	0,47	0,05
6	0,13	3,66	0,16	1,46		0,19	0,47	0,14	0,39	0,06
7	0,11	3,61	0,14	1,88		0,16	0,30	0,12	0,25	0,05
8	0,11	3,49	0,14	2,33		0,16	0,29	0,12	0,20	0,00
9	0,09	3,38	0,12	1,80		0,13	0,07	0,10	0,20	0,00
10	0,05	2,82	0,07	1,85		0,08	0,25	0,06	0,11	0,00
11	0,04	3,59	0,05	2,25		0,06	0,24	0,05	0,00	0,00
12	0,07	3,50	0,09	1,90		0,10	0,26	0,08	0,15	0,05
$\overline{I}_{\Pi}^{(c)}$		3,45		1,90	1,75		0,33		0,51	0,21

Розрахунок узагальнюючої схеми українського текстового слова по $\bar{I}$ даний у таблиці 2.3, а на рисунку 2.6 представлено її малюнок.

По-друге, може бути побудована узагальнююча схема, на яку не накладаються обмеження, зв'язані із середньою довжиною слова в текстах даної мови. Оскільки границя довжини слова нічим не обмежена, то довжина узагальнюючої схеми в цьому випадку прагне до нескінченності. Однак практично виявляється, що дуже довгі слова зустрічаються в мові порівняно рідко. Так, наприклад, у російській мові слова довжиною більше одинадцяти букв, а у французькій і англійській мовах більше десяти букв становлять у середньому не більше 5-7% загального числа слів. Тому доцільно обмежити розглянуту схему деякою межею. В українській мові, наприклад, ця схема доведена до дванадцятої букви (11 букв+пробіл).

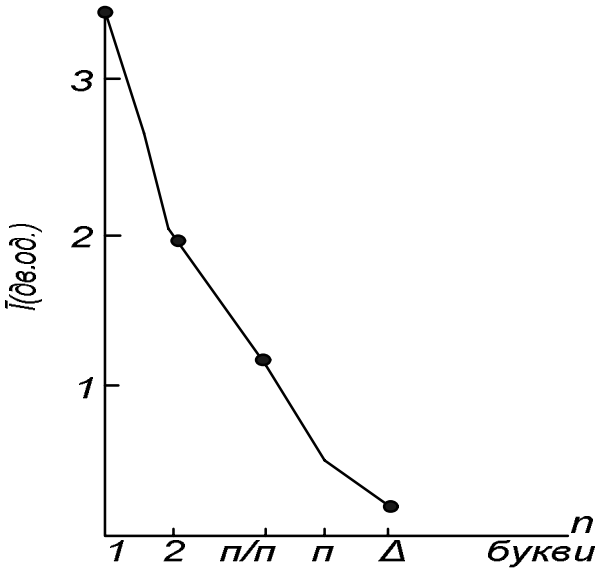


Рисунок 2.6 – Узагальнююча схема українського текстового слова, яка відповідає його середній довжині, побудована по $\bar{l}$

Узагальнюючі схеми слів будуються також на складовому й морфемному рівнях. Нижче як зразок приводяться таблиці 2.4, що містять дані для побудови узагальнюючих складових і морфемних схем українського позатекстового й текстового слова. Креслення цих схем представлені на рисунку 2.7.

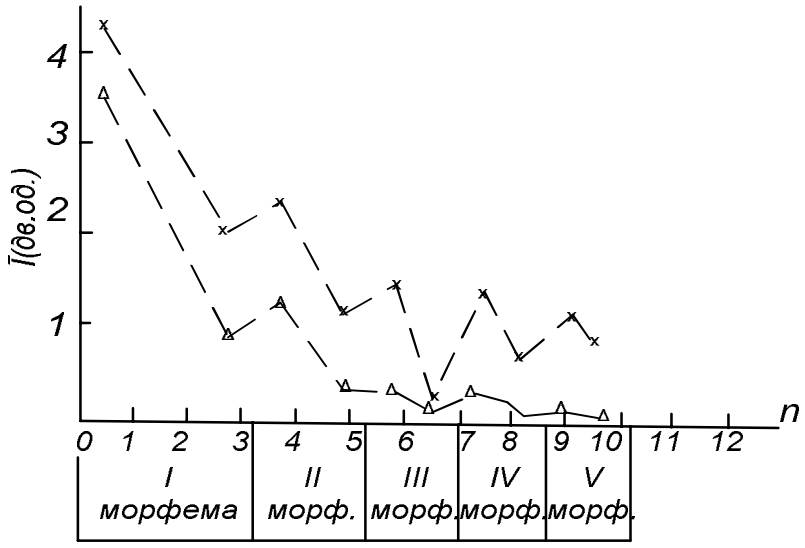


Рисунок 2.7 – Загальні морфемна схема розподілу інформації російської мови: 1 – позатекстове слово; 2 – текстове слово

Таблиця 2.4 – Морфемна схема української мови за інформативністю $\bar{I}$

Довжина слова в складах	Вірогідність появи слова даної дов-жини поза текстом	Інформація, в дв.од.									
		Морфеми									
		I		II		III		IV		V	
		Перша літера	Остання буква	Перша літера	Остання буква	Перша літера	Остан-ня буква	Перша літера	Остання буква	Пер-ша літера	Остання буква
1	0.260	4.22	1.87								
2	0.417	4.22	2.28	2.24	1.05						
3	0.165	4.22	1.80	2.30	1.96	1.72	0.73				
4	0.118	4.22	2.02	2.93	0.99	1.31	0.00	1,18	0,56		
5 і більше	0,040	4.22	2,17	2,84	0,39	1,49	0,00	1,17	0,80	1, 11	0,79
Загальна морфемна схема слова поза текстом		4.22	2,03	2,39	1.19	1,54	0,35	1,39	0,63	1, 11	0,79
Загальна морфемна схема текстового слова		3,49	0,88	1,27	0.31	0,30	0,07	0,36	0,00	0, 11	0.00
Середня довжина морфеми в буквах		3,24		2,07		1,70		1,70		1,44	

Морфемні схеми позатекстових слів у всіх мовах характеризуються послідовним чергуванням максимумів і мінімумів інформації. Максимуми інформації падають на першу, а мінімуми на останню букву морфеми. Іншими словами, на всьому протязі позатекстового письмового слова виявляються чітко

позначеними границі між морфемами. Ці границі збігаються з межею розподілу між останньою буквою попередньої морфеми (мінімум інформації) і першою буквою наступної морфеми (максимум інформації). Якщо звернутися до текстових слів, то виявиться, що вони також виявляють морфемне членування, хоча воно й виражено тут значно слабкіше.

Отже, можна стверджувати, що з погляду синтаксичної інформації слову в європейських мовах властива чітко виражена морфемна структура, що придушує складове членування слова. Такий стан справа зберігається у всіх тих мовах, в яких склад не збігається з морфемою.

Взаємодія різних механізмів тезаурусу еталонного угадувала визначається тим фактом, що морфологічна структура слова придушує синтагматичну систему фігур (тобто складів). Відомо, що мову можна промоделювати складним марковським процесом лінійного проходження різних елементів мови - як фігур, так і знаків-символів. Таким чином, ймовірнісно-статистичні закономірності визначають сполучуваність фігур і взаємодіють із імовірністю сполучуваності символів (морфем, синтагм, слів, лексичних сполучень і т.і.).

2.6 Висновки

В даному розділі вибрано критерій ефективності систем розпізнавання текстових образів, який оснований на моделях потенційної і реальної систем. Він дозволяє керувати вибором оптимальних параметрів системи уже на етапі проектування, а не тільки оцінювати ефективність уже готових систем.

В якості процедури розпізнавання символів тексту під час його електронізації запропоновано використовувати послідовно-паралельну стратегію розпізнавання у вигляді дерева рішень, яка дозволяє підвищити точність і швидкість розпізнавання, а також використовувати ієрархічно-модульний принцип побудови програмних засобів для задач різної складності розпізнавання.

З метою визначення послідовності класифікації графем тексту в вузлах дерева рішень розроблено методи оцінки інформативності окремих складових тексту і ефективності розпізнавання тексту в цілому.

Проведений в розділах 2.3–2.5 інформаційний аналіз показав, що найбільш значимою інформацією для розпізнавання і розуміння текстового документу володіють такі його елементи, як графічні особливості напису окремих слів, графеми визначених літер, морфем, довжина складів і окремих літер, статичні і перехідні ймовірності вказаних елементів. Тому в наступному розділі розробляються алгоритми і програмне забезпечення для побудови бази даних, в якій буде зберігатися інформація про вказані характеристики складових тексту.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Варіантний аналіз програмних платформ

Розроблене в магістерській дипломній роботі програмне забезпечення реалізоване на базі платформи Microsoft.NET Framework 2.0, мовою програмування був обраний C# [46] .

Microsoft.NET – програмна технологія, для створення як звичайних програм, так і веб-додатків (як платформа для розробок уперше запропонована фірмою Microsoft).

Однієї з основних ідей Microsoft.NET є сумісність різних служб, написаних на різних мовах. Кожна бібліотека (складання) в .NET має відомості про свою версію, що дозволяє усунути можливі конфлікти між різними версіями зборок.

Платформа Microsoft .NET надає:

– стійке загальномовне середовище виконання CLR (Common Language Runtime), що входить до складу даної платформи;

– засоби розробки додатків на кожній з багатьох мов програмування, підтримуваних платформою .NET;

– лежачу в основі відкритої моделі програмування величезну бібліотеку класів .NET Framework. Ці класи містять багаторазово використовуваний код. Вони доступні в будь-якій мові програмування, підтримуваному платформою .NET;

– підтримку мережної інфраструктури, побудованої на верхньому шарі стандартів Internet, внаслідок чого забезпечується високий рівень взаємодії між додатками;

– підтримку нового промислового стандарту, а саме технології Web-Служб. Технологія Web-Служб надає новий механізм створення розподілених додатків. По суті, вона є поширенням технології створення додатків на базі компонентів і на сферу Internet;

– модель безпеки, що програмісти можуть легко використовувати у своїх додатках;

– потужні інструментальні засоби розробки [47].

Платформа NET має також кілька важливих характерних рис, а саме:

– каркас .NET Framework;

– загальномовне середовище виконання CLR (Common Language Runtime);

– можливість розробки додатка на багатьох мовах програмування, підтримуваних платформою .NET;

– інструментальні засоби розробки додатків.

У каркасі .NET Framework представлено більше 2500 класів, що містять повторно використовуваний код. Ці класи доступні в будь-якій мові програмування, що підтримується платформою. Бібліотека класів .NET Framework є розширюваною На основі вже існуючих базових класів можна створити нові похідні класи, причому похідні класи можуть бути реалізовані на зовсім іншій мові програмування [46].

До складу бібліотеки класів .NET Framework, входять класи, які використовуються при розробці Windows-Додатків, Web-додатку, а також додатку з базами даних. У бібліотеці класів .NET Framework є також класи, що забезпечують взаємодію з мовою XML, з моделлю компонентних об'єктів Microsoft (COM) і з будь-якою платформою, що підтримує інтерфейс 32-розрядних Windows-Додатку (Win32 API).

Класи бібліотеки .NET Framework створюють каркас (інфраструктуру) додатка й надають системні служби .NET-Додаткам.

C# – мова програмування, яка включає в себе об’єктно-орієнтовані і аспектно-орієнтовані концепції. Розроблений в 1998-2001 роках як основна мова розробки додатків для платформи Microsoft.NET. Він найбільше підходить для розробки додатків для цієї платформи. Компілятор C# входить у стандартну установку самої .NET.

C# відноситься до родини мов з C-подібним синтаксом, з них його синтаксис найбільш близький до C++ і Java. Перейнявши багато чого зручного від своїх попередників – мов C++, Java, Delphi, Модула і Smalltalk –C#, опираючись на практику їх використання, виключає деякі моделі, які зарекомендували себе як проблематичні при розробці програмних систем: так, C# не підтримує великої кількості наслідування класів (на відміно від C++).

Тому C# на базі платформи Microsoft.NET Framework є найбільш універсальною та зручною мовою програмування для створення розробленого в магістерській дипломній роботі програмного продукту [32].

3.2 Розробка структури програмного забезпечення

Структура розробленого програмного забезпечення представлена на рисунку 3.1.

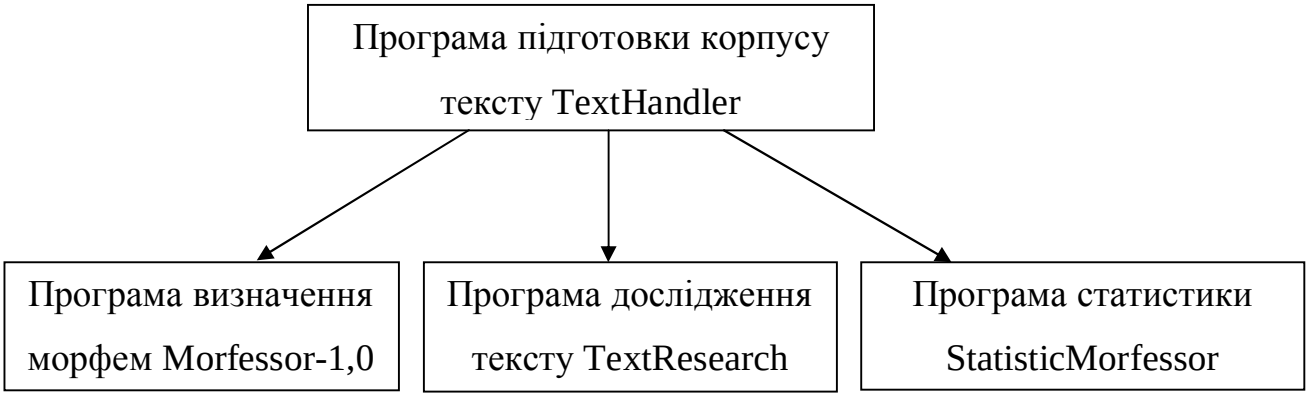


Рисунок 3.1 – Структура програмного забезпечення для побудови бази даних

Обґрунтуємо дану структуру. На вхід програми подається текст розміром більше 900000 слів (UAMORFTEXT.txt). Потім за допомогою підпрограми Morfessor 1.0.exe створюється база даних морфем української мови. Далі за допомогою підпрограми TextResearching.exe вхідний текст досліджується на його

інформаційні елементи. Підпрограма MorfessorStatistic.exe формує двомірний масив ймовірностей переходів між морфемами української мови.

3.3 Розробка алгоритмів та програм опису створення бази даних морфем української мови

Опишемо кожну з реалізованих функцій та програм.

Для використання корпусу тексту в алгоритмах роботи з морфемами української мови, з нього потрібно вилучити нелітерні символи та розбити даний текст на окремі слова (кожне слово в окремому рядку).

На вхід алгоритму – підготовка корпусу тексту до тестування (рисунок 3.2) – подається текстовий файл з текстом для тестування, на виході алгоритму отримуємо масив окремих слів, що в подальшому використовується алгоритмами пошуку та аналізу морфем української мови.

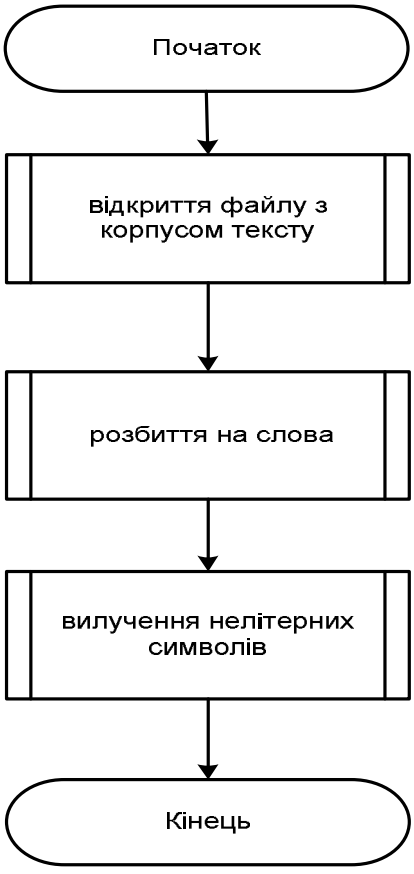


Рисунок 3.2 – Схема алгоритму підготовки корпусу тексту до тестування

На рисунку 3.3 представлена схема алгоритму пошуку морфем української мови.

Рисунок 3.3 – Схема алгоритму пошуку морфем української мови

Даний алгоритм в автоматизованому режимі з великого корпусу тексту визначає масив морфем української мови. Він побудований на статистичних n-грамах та використовує скрипт (мова Perl) пошуку морфем англійської мови розроблений вченими Креутсом М. та Леджусом К [20].

Отримана в результаті доопрацювання програма „Морфесор 1.0” дозволила повністю автоматизувати роботу згаданої вище програми для проведення морфологічного аналізу (розбиття слів на елементарні частини – префікси, суфікси, корені і т.д.) та навчити його працювати з українськими та російськими символами, закодованими за допомогою відповідних кодувальних сторінок (“windows-1251”, “UTF8” та “ASCII”). Також дана програма дозволяє перевірити адекватність утвореної лінгвістичної моделі тексту.

На рисунку 3.4 представлена схема алгоритму визначення статистичних ймовірностей появи морфем в корпусі тексту, який визначає статистичні ймовірності появи морфем в корпусі тексту.

На вхід алгоритму подається масив морфем визначених за допомогою схеми алгоритму представленому на рисунку 3.3 та великий корпус тексту (обсяг 7 млн. слів). Далі алгоритм виконує підготовку тексту до тестування за алгоритмом, який представлений на рисунку 3.2.

Наступним кроком є пошук статистичних ймовірностей появи і-ої морфеми в даному корпусі тексту. Для цього масив вхідних слів взятих з тестового корпусу тексту проганяється по кожному слову і, якщо і-та морфема є в даному слові то лічильник і-ої ймовірності збільшується на одиницю. Завершальним етапом є формування бази даних знайдених в алгоритмі ймовірностей для кожної і-ої морфеми.

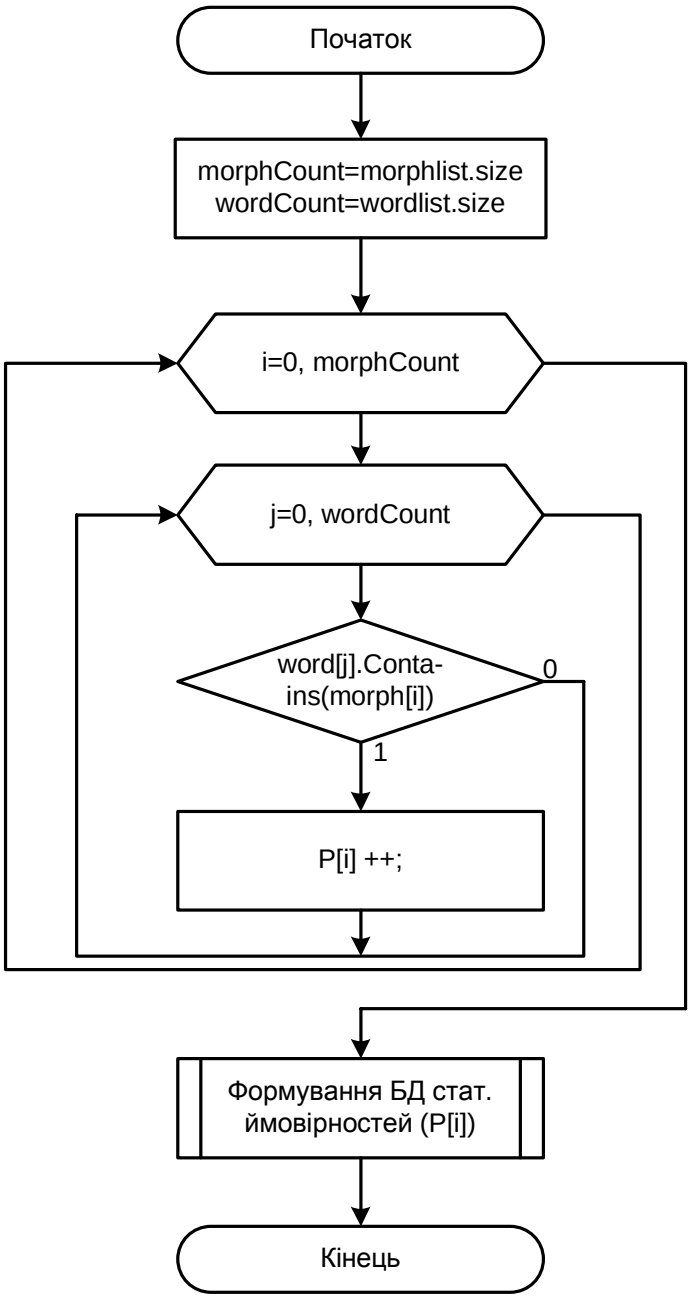


Рисунок 3.4 – Схема алгоритму визначення статистичних ймовірностей появи морфем

Дуже важливим етапом ефективної системи введення, обробки та розпізнавання текстової інформації побудованої на морфологічному аналізі є визначення ймовірностей переходів між морфемами. Цей етап реалізує алгоритм визначення ймовірностей переходів між морфемами корпусу тексту (рисунок 3.5)

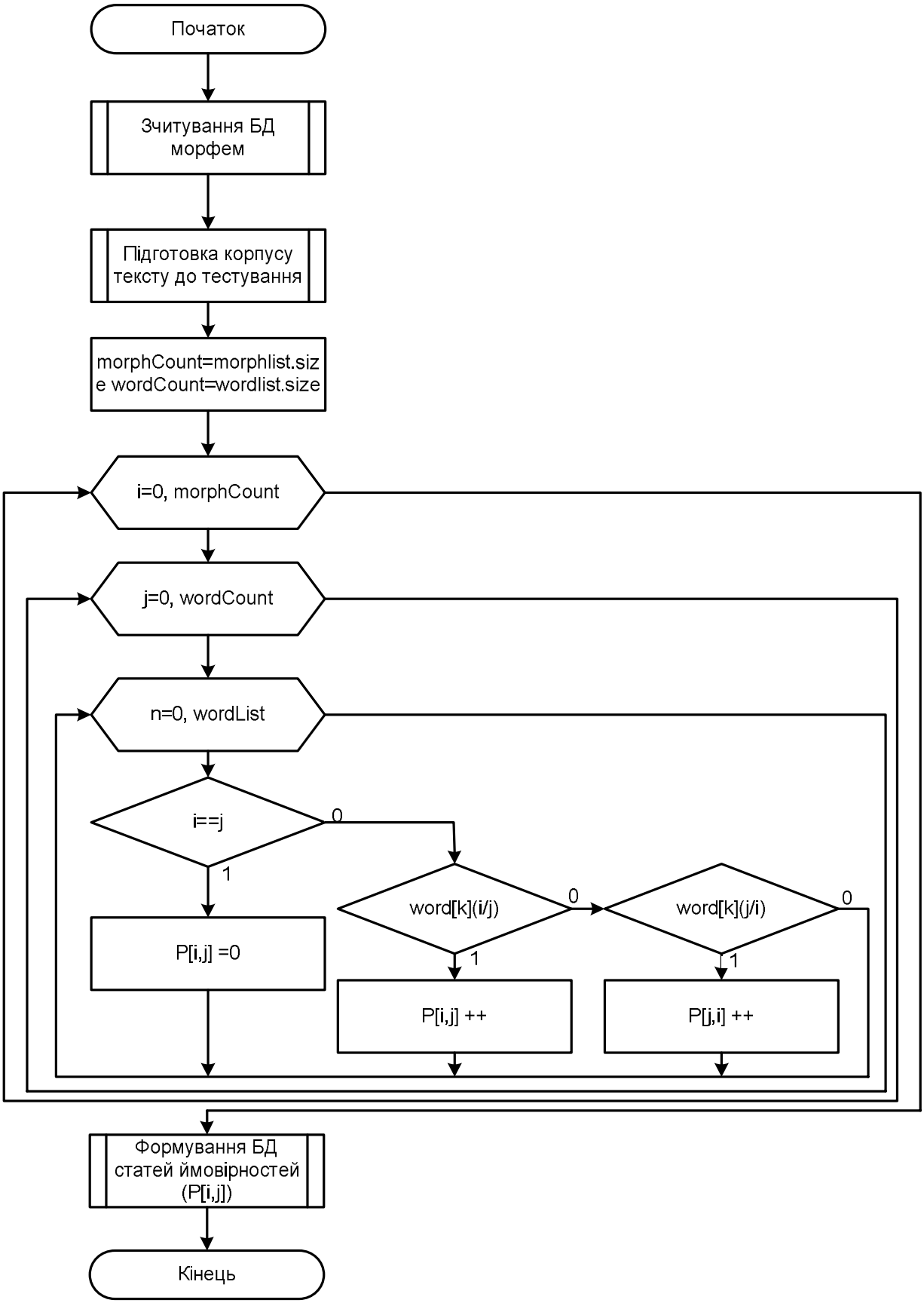


Рисунок 3.5 – Схема алгоритму визначення ймовірностей переходів між морфемами корпусу тексту

Суть даного алгоритму полягає в пошуку ймовірності переходу між і-ою та j-ою морфемами, для цього алгоритм проганяє корпус тексту по кожному слову і, якщо містить і-у та j-у морфему слідуючі одна за одною, то лічильник ймовірностей P_{ij} збільшується на одиницю.

В результаті роботи алгоритму отримується двомірний масив ймовірностей переходів між морфемами української мови (таблиця 3.1).

Таблиця 3.1 – Двомірний масив ймовірностей переходів між морфемами

0	1	2	3	…	i-1	i
1		P_{12}	P_{13}	$P_{1...}$	P_{1i-1}	P_{1i}
2	P_{21}		P_{23}	$P_{2...}$	P_{2i-1}	P_{2i}
3	P_{31}	P_{32}		$P_{3...}$	P_{3i-1}	P_{3i}
…	$P_{...1}$	$P_{...2}$	$P_{...3}$		$P_{... i-1}$	$P_{...i}$
j-1	P_{j-11}	P_{j-12}	P_{j-13}	$P_{j-1...}$		P_{j-1i}
j	P_{j1}	P_{j2}	P_{j3}	P_{j4}	P_{ji-1}	

Характерною ознакою сегментації слова на літери/морфеми є залежність довжини слова від кількості складів.

Суть алгоритму визначення залежності довжини слова від кількості складів (рисунок 3.7) полягає у визначенні класів довжин слів залежних від кількості складів (кількість складів визначається кількістю голосних).

На вхід алгоритму поступає великий корпус тексту, далі алгоритм проганяє даний корпус по кожному слову та визначає його довжину в пікселях та кількість складів.

Результатом роботи алгоритму є база даних залежностей кількості складів слова від його довжини (рисунок 3.6).

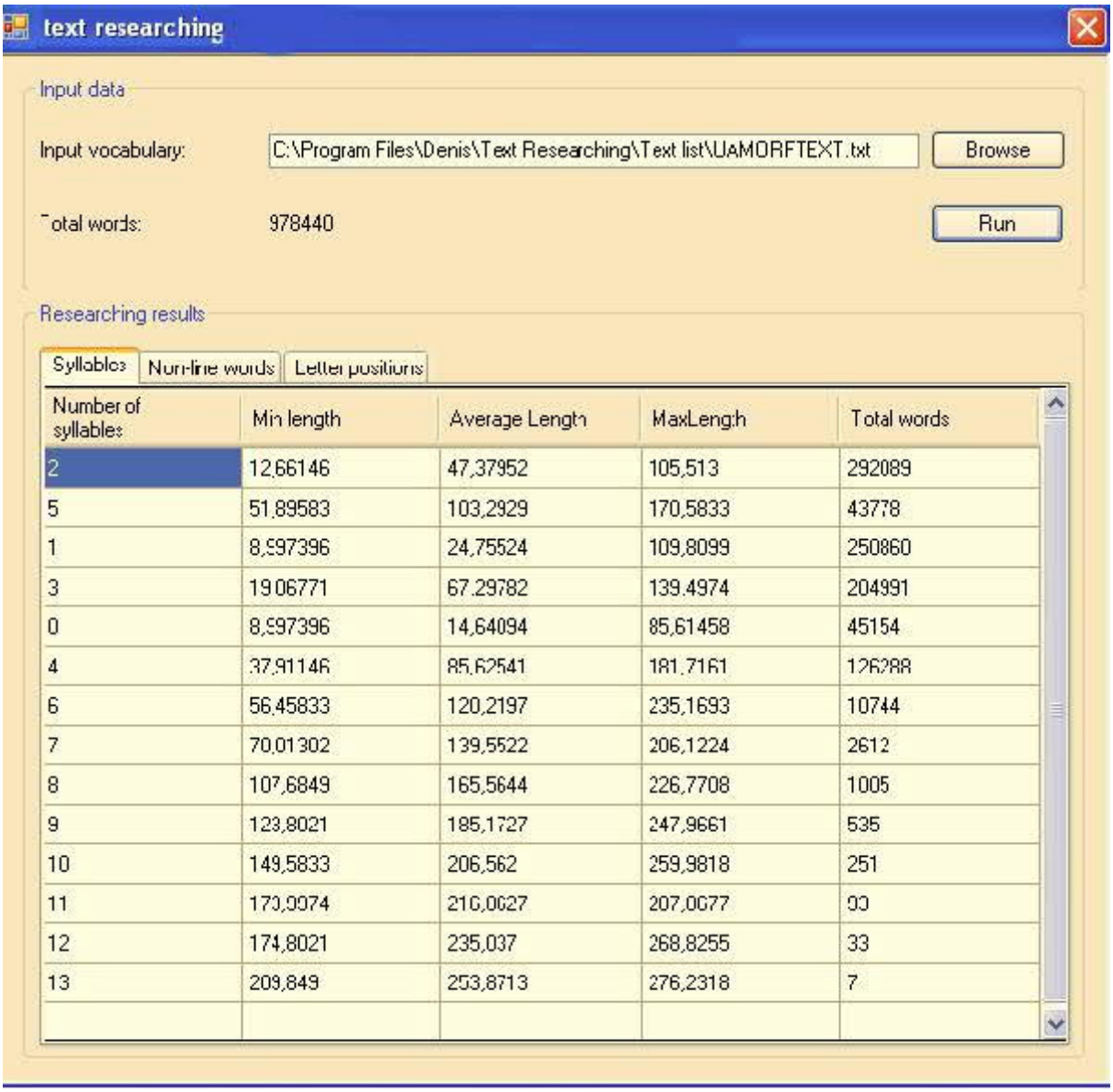


Рисунок 3.6 – База даних залежності кількості складів слова від його довжини

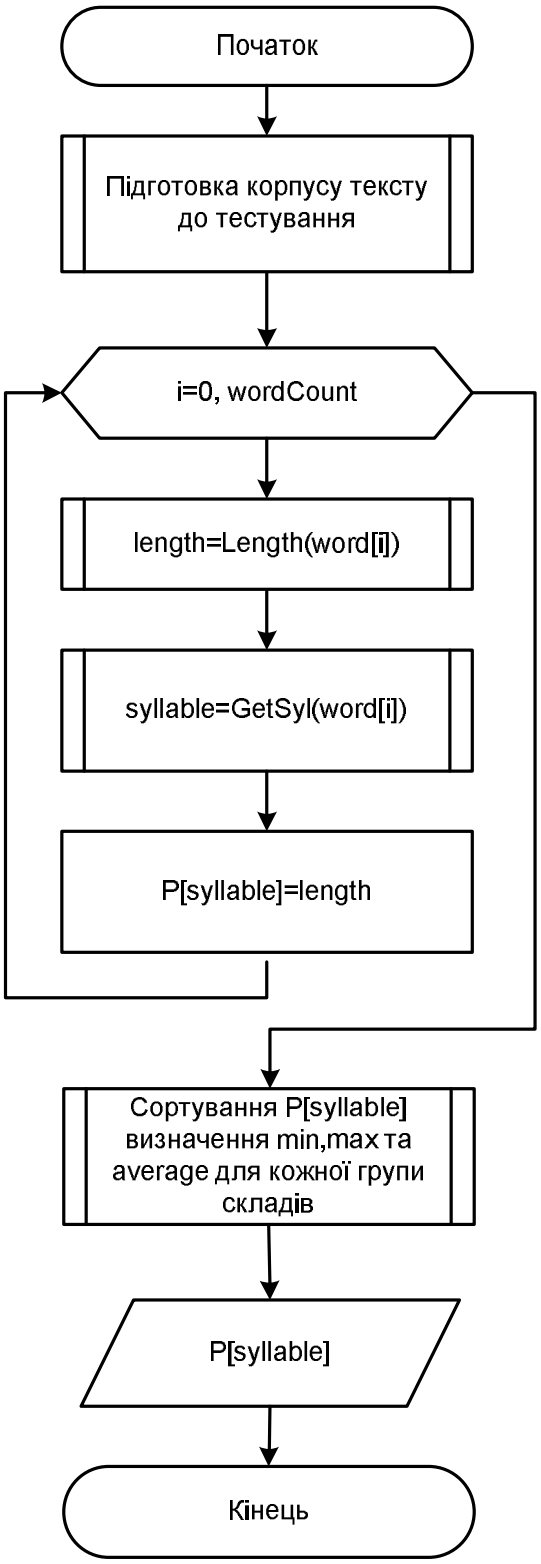


Рисунок 3.7 – Схема алгоритму визначення залежності довжини слова від кількості складів

Ще одним важливим етапом ефективної системи введення, обробки та розпізнавання текстової інформації побудованої на морфологічному аналізі є

визначення статистичних ймовірностей появи літер, що мають надстрокові та підстрокові ознаки (рисунок 3.8).

Суть цього алгоритму полягає в тому, що на його вхід подається підготовлений текст до тестування, з нього формується масив надстрокових та під строкових літер (в українській мові таких літер 10).

Потім даний корпус тексту проганяється алгоритмом по кожному слову і, якщо слово містить надстрокову чи під строкову ознаку то лічильник ймовірності P_j збільшується на одиницю. На виході формується база даних появи літер, що мають надстрокові та підстрокові ознаки (рисунок 3.9).

Останній алгоритм визначає місцезнаходження літери, яка містить надстрокової та під строкової ознаки в тексті (рисунок 3.10).

На його вхід подається підготовлений корпус тексту до тестування. Потім формується масив masLiter – масив літер з надстроковими та підстроковими ознаками. Після чого текст проганяється алгоритмом і формуються проміжки початку(begin=[0, middle)), середини(middle=length/2) і кінця(end=(middle, length]) слова. Якщо літера з надстроковою чи підстроковою ознакою попадає в один із цих проміжків то відповідні ймовірності збільшуються на одиницю і формується база даних знаходження букви на початку, середині чи кінця слова (рисунок 3.11).

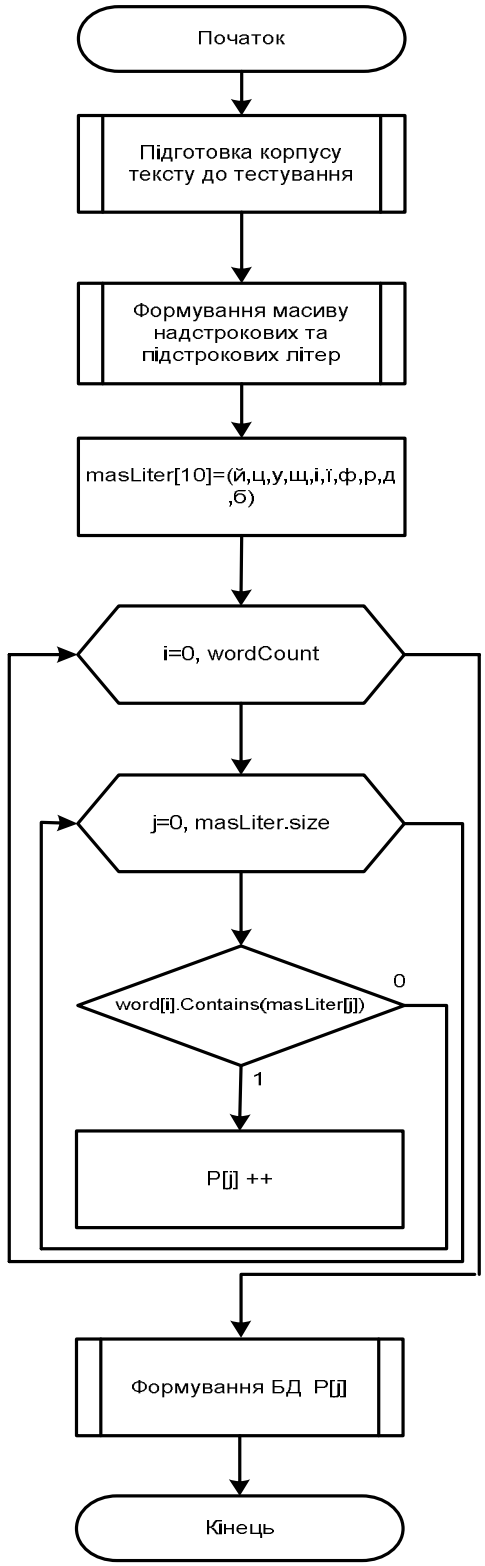


Рисунок 3.8 – Схема алгоритму визначення статистичних ймовірностей появи літер, що мають надрядкові та підрядкові ознаки

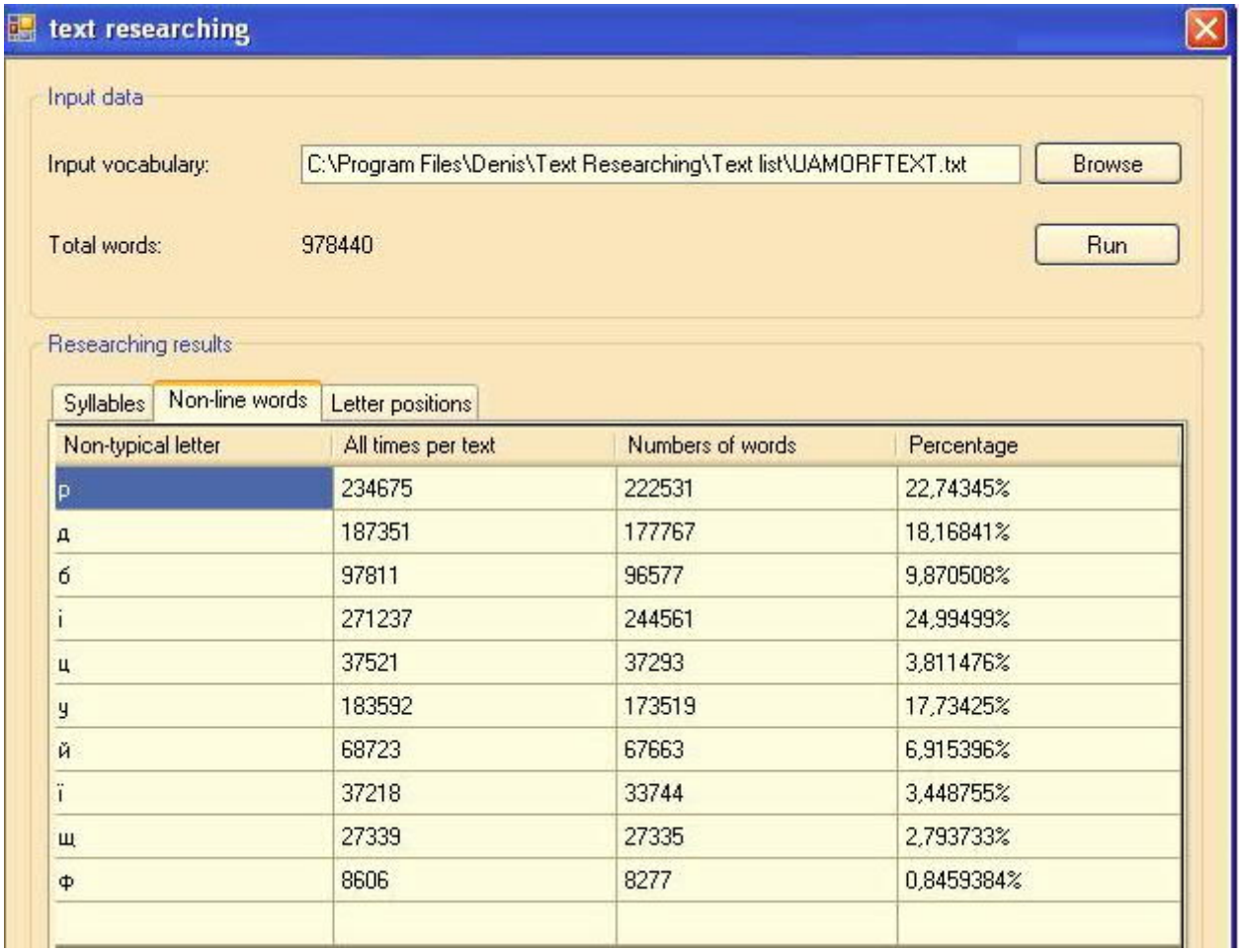


Рисунок 3.9 – База даних появи літер, що мають надрядкові та підрядкові ознаки

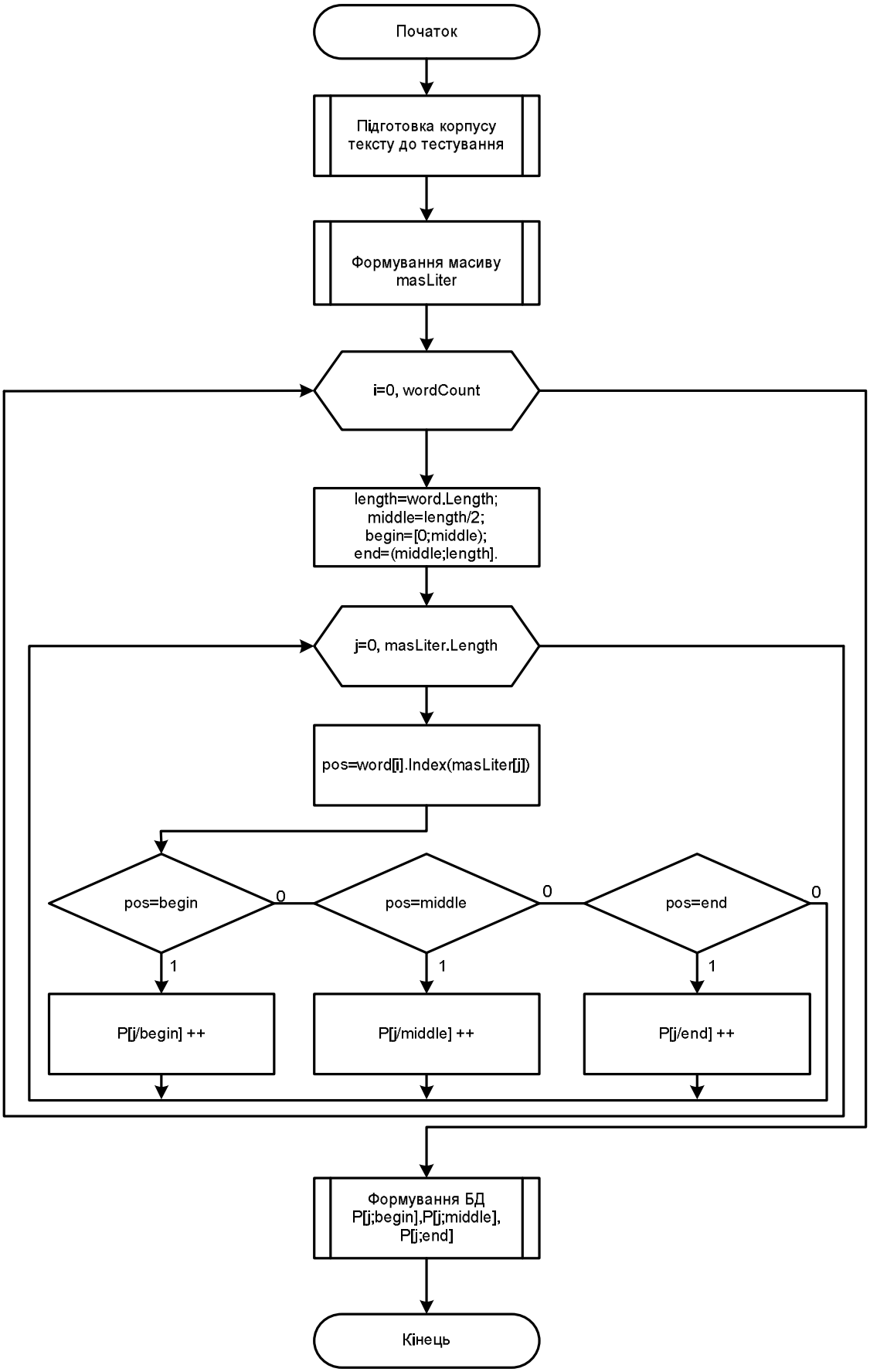


Рисунок 3.10 – Схема алгоритму визначення місцезнаходження літери в слові, яка має надрядкові та підрядкові ознаки

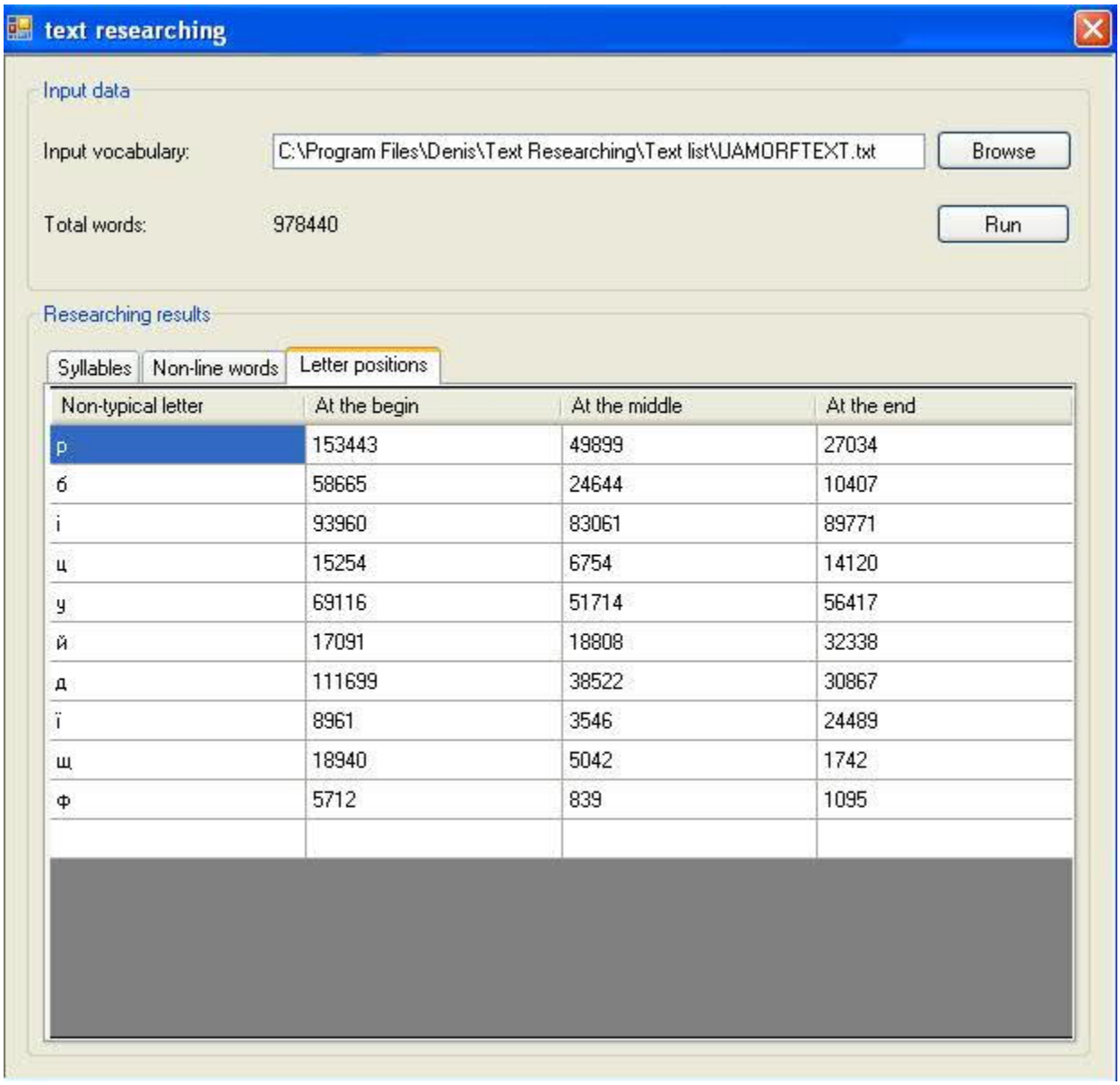


Рисунок 3.11 – База даних знаходження літери на початку, середині чи кінця

слова, яка має надрядкові чи підрядкові ознаки

Лістинг основних модулів програми приведено в додатку Б.

3.4 Розробка структури вхідних, вихідних та проміжних даних

Початкові дані, які подаються на вхід програми – україномовний текст без формул, абревіатур і скорочень обсягом не менше 60000 слів, тип шрифта – друкований заданого стилю.

Текст завантажується в підпрограму підготовки корпусу тексту до тестування. Де з нього видаляються нелітерні символи. Після чого він поступає на вхід підпрограми пошуку морфем української мови. Потім визначаються статистичні ймовірності появи і-ої морфеми в даному корпусі тексту.

Вихідними даними є бази даних залежності кількості складів слова від його довжини; появи літер, що мають надстрокові та підстрокові ознаки в слові; знаходження літери на початку, середині чи кінці слова, яка має надстрокові чи підстрокові ознаки.

Також кінцевим результатом виконання програми є сформована база даних морфем української мови.

Опишемо всі дані, які виводять всі функції та програми.

Підпрограма Morfessor 1.0.exe використовує наступну функцію DataHandler, яка підготовлює вхідний текст до тестування (вилучає з нього нелітерні символи). А також функцію RunMorfessorScript, яка формує з фідформатованого тексту базу даних морфем української мови.

TextResearching.exe використовує функцію TextHandler, яка підготовлює текст для дослідження і досліджує інформаційні елементи тексту (залежність кількості складів слова від його довжини; поява літер, що мають надстрокові та підстрокові ознаки; знаходження літери на початку, середині чи кінця слова, яка має надстрокові чи під строкові ознаки).

MorfessorStatistic.exe використовує функцію StatisticManager, яка визначає перехідні і статичні властивості морфем української мови

3.5 Керівництво оператора

Розроблений програмний продукт розміщується на диску у вигляді інсталяційного пакету і має назву setup.exe з розміром 31 МБ.

Для початку роботи з програмним продуктом оператор повинен:

- запустити файл setup.exe;
- вказати місце куди потрібно розпакувати пакет (за умовчанням він розпакується на жорсткий диск C:\Program Files\Denis);

Після чого програма готова до використання.

Для того, щоб сформувати базу даних морфем української мови, оператору потрібно запустити файл Morfessor 1.0.exe. Після чого з'явиться графічний інтерфейс програми, де користувач повинен вказати директорію для збереження та ім'я результуючого файлу. Програма автоматично призначає ту саму директорію для результуючого файлу, де знаходиться вхідний файл. Також програма генерує ім'я результуючого файлу, шляхом додавання до імені вхідного файлу службового слова «Results» та поточної дати.

Для того, щоб дослідити вхідний текст на його інформаційні елементи під час розпізнавання потрібно запустити файл TextResearching.exe. Після чого з'явиться графічний інтерфейс де користувач повинен вказати директорію та ім'я результуючого файлу. Потім натиснути на кнопку "Run" розпочавши виконання програми. В результаті з'являться три таблиці, перша з яких означає досліджень інформативності ознаки "довжина слова"; друга – досліджень інформативності ознаки літер, що мають надстрокові та підстрокові символи; третя – досліджень інформативності ознаки знаходження літери на початку, середині чи кінці слова, яка має надстрокові чи під строкові ознаки.

Запустивши файл MorfessorStatistic.exe, перед користувачем з'явиться графічний інтерфейс де він повинен вказати файлове ім'я списку морфем, а також – тексту і куди потрібно зберегти результати дослідження. Після чого оператор отримає двомірний масив ймовірностей переходів між морфемами української мови.

Вихід з програми здійснюється натискання лівою кнопкою миші на значку “Закрити”, який розташований в верхньому правому куті інтерфейсу програми.

3.6 Проведення тестування та аналіз результатів

Враховуючи невеликі системні вимоги обраного програмного забезпечення (C#) можна виділити наступні вимоги до апаратної частини: тестування проводилось на комп’ютері Celeron 1,7 МГц та об’ємом ОЗУ 512

Для початку роботи з розробленим програмним продуктом потрібно підготовити корпус тексту до тестування. Тому спочатку ми запускаємо файл Morfessor.exe, який проводить автоматизовану підготовку довільного тексту до вигляду, придатного для морфологічного аналізу. Дана програма без втручання користувача:

- вилучає із вхідного тексту усі символи, що не є літерами;
- записує кожне окреме слово у новий рядок текстового файлу.

Графічний інтерфейс програми “Морфесор 1.0” показаний на рисунку 3.12.

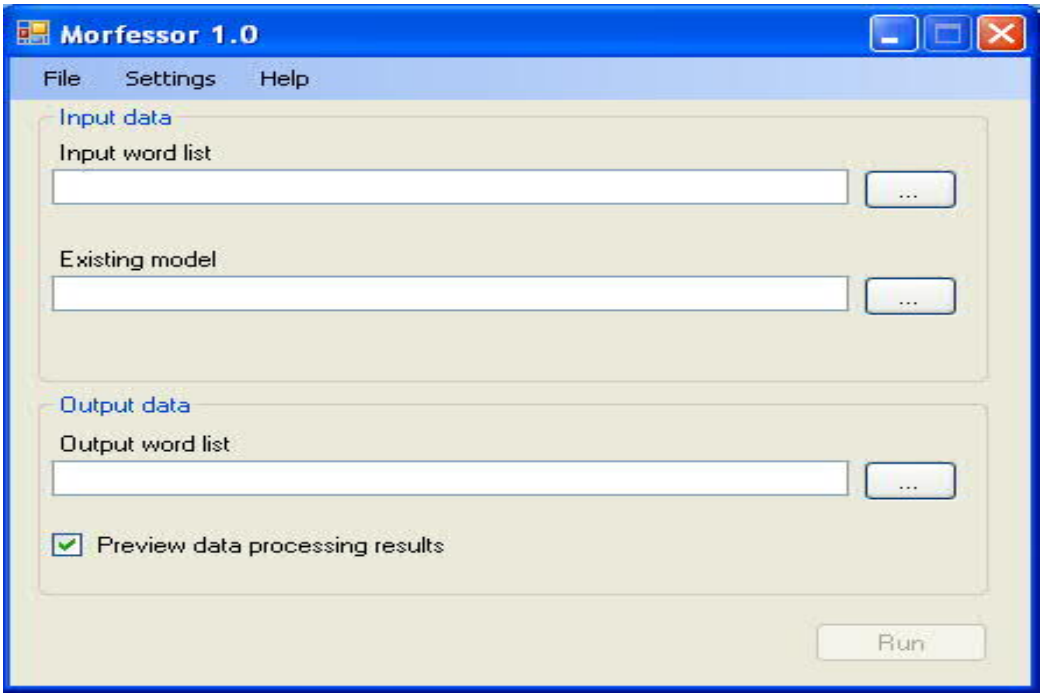


Рисунок 3.12 – Графічний інтерфейс програми “Морфесор 1.0”

У головному вікні програми користувач має вказати ім’я текстового файлу, який необхідно сегментувати на морфеми. Також користувач повинен вказати директорію для збереження та ім’я результуючого файлу. Програма автоматично призначає ту саму директорію для результуючого файлу, де знаходиться вхідний файл. Також програма генерує ім’я результуючого файлу, шляхом додавання до імені вхідного файлу службового слова «Results» та поточної дати.

Відповідне вікно інтерфейсу представлено на рисунку 3.13.

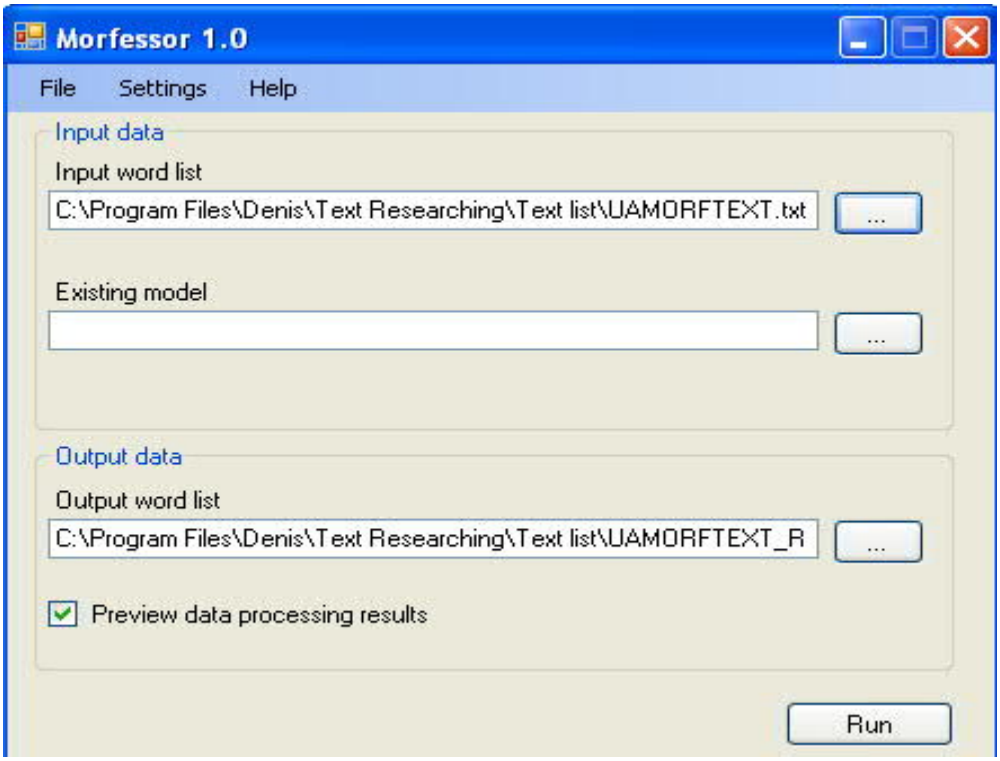


Рисунок 3.13 – Сегментація текстового файлу на морфеми

Як тільки модель морфемної сегментації навчена на деякому наборі даних, вона може використовуватися для сегментування нових словоформ. У цьому режимі сегментації програми “Морфесор” не відбувається навчання моделі. Кожне слово на вході сегментується на морфеми за допомогою алгоритму Вітербі, що знаходить самі схожі сегменти слова в послідовності морфем, що є в існуючій моделі. (для того, щоб забезпечити можливість існування як мінімум однієї сегментації, кожен окремий символ слова, що не існує, як морфема, може розглядатися як морфема з дуже малою імовірністю). Користувач може ввести

ім’я файлу, що містить існуючу модель (друге вікно графічного інтерфейсу на рис. 3.13).

Для дослідження був сформований текст із статистично достатньої вибірки слів розміром більше 900 000 слів UAMORFTEXT.txt. На виході отримаємо відформатований текст pUAMORFTEXT.txt.

Приклад роботи процедури підготовки вхідного тексту до морфологічного аналізу:

а) вхідний текст

Ігор Дзюба
Помаранчеві серця
Добро всі вулиці заполонило,
Воно перемагае зло.
В містах любов замайоріла.
Навколо сніг, але в душі тепло.

б) відформатований текст

Ігор
Дзюба
Помаранчеві
серця
Добро
всі
вулиці
заполонило
Воно
перемагае
зло
В
містах
любов
замайоріла
Навколо
сніг
але
в
душі
тепло

Після того як ми підготували вхідний текст до тестування, запускаємо файл TextResearching.exe, який визначає ймовірність спостереження P_c морфем в тексті і імовірності переходів між ними P_p. Відповідне вікно інтерфейсу представлено на рисунку 3.14.

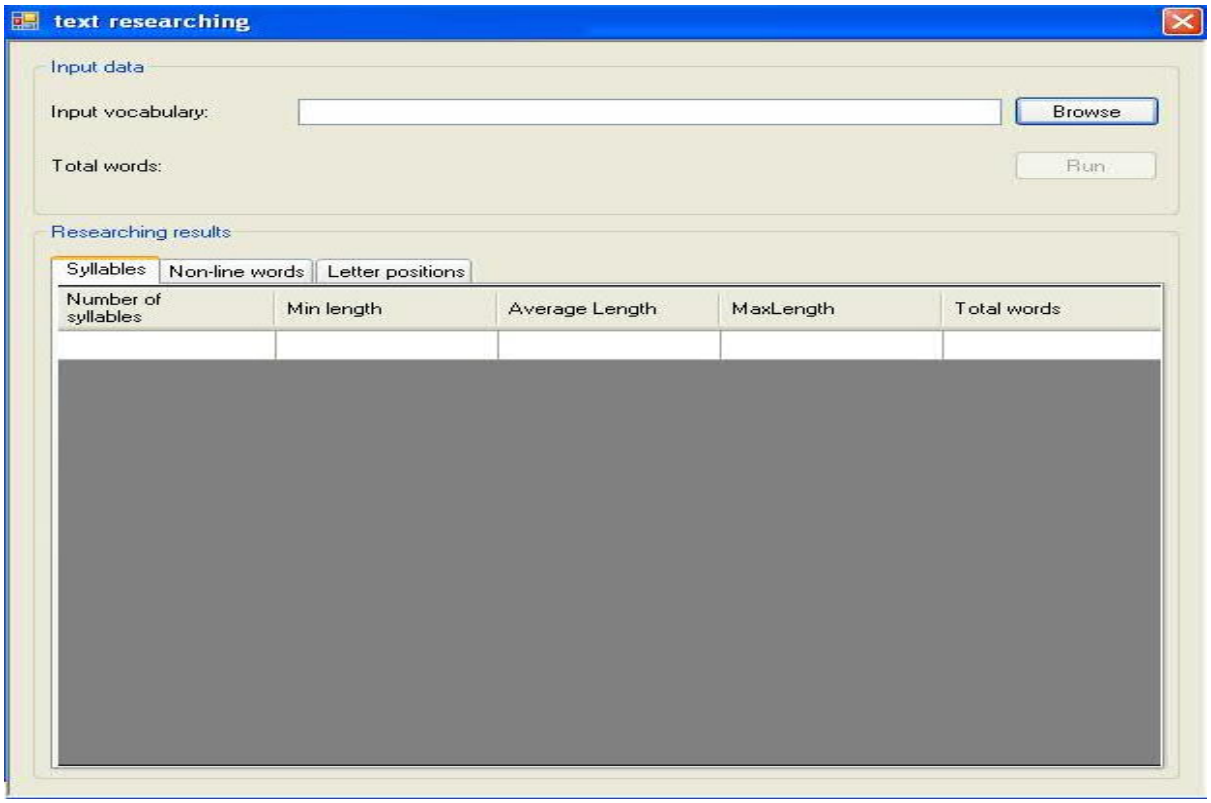


Рисунок 3.14 – Графічний інтерфейс файлу TextResearching.exe

Тепер завантажуємо відформатований текстовий файл, для оцінки інформативності ознак текстових образів. І отримаємо наступні бази даних (рисунки 3.15 -3.17):

- база даних залежності кількості складів слова від його довжини;
- база даних появи літер, що мають надстрокові та підстрокові ознаки;
- база даних знаходження букви на початку, середині чи кінця слова;

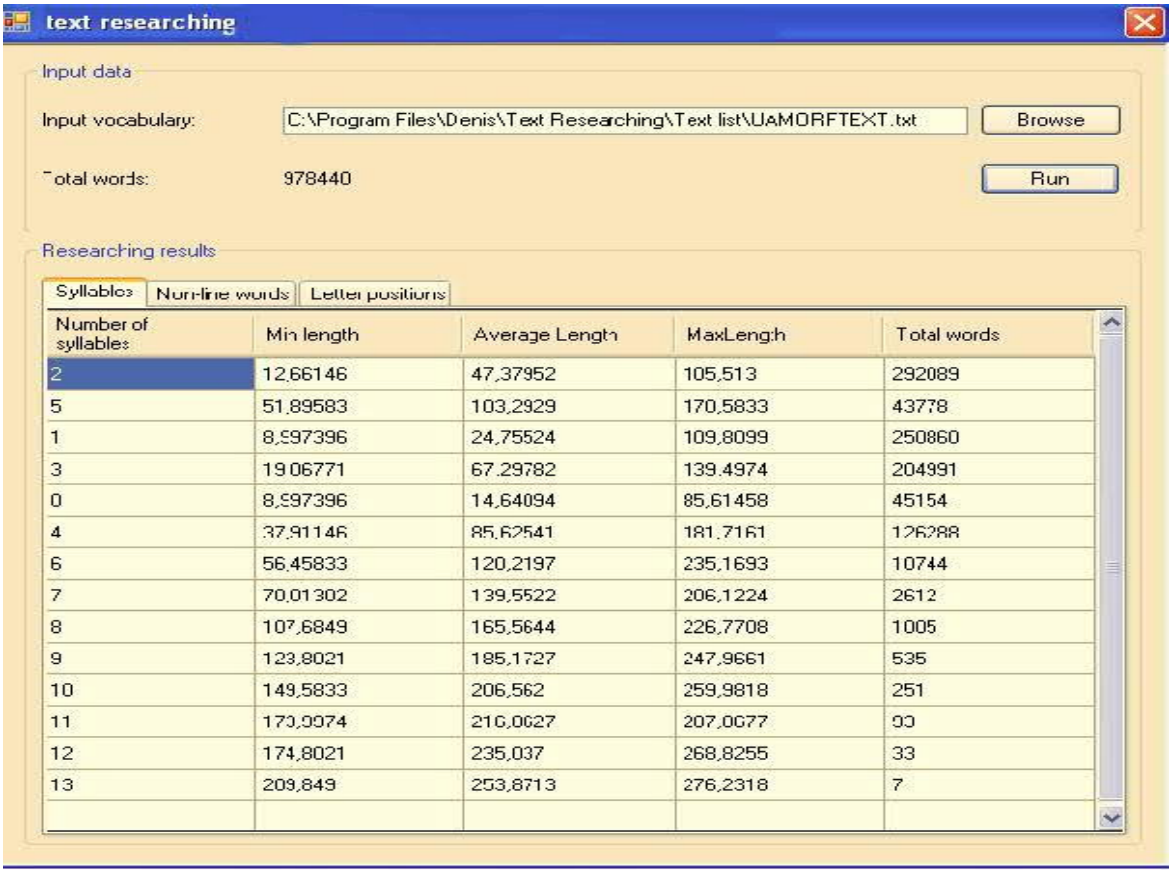


Рисунок 3.15 – База даних залежності кількості складів слова від його довжини

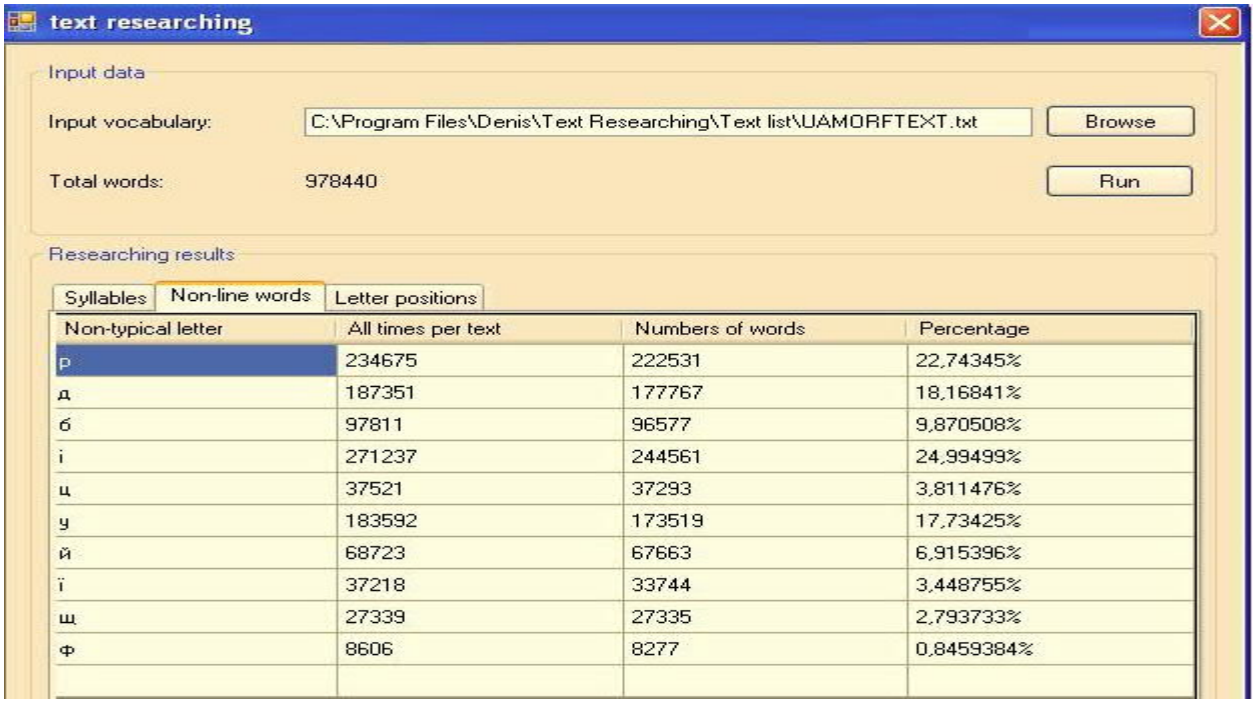


Рисунок 3.16 – База даних появи літер в тексті , що мають надстрокові та підстрокові ознаки

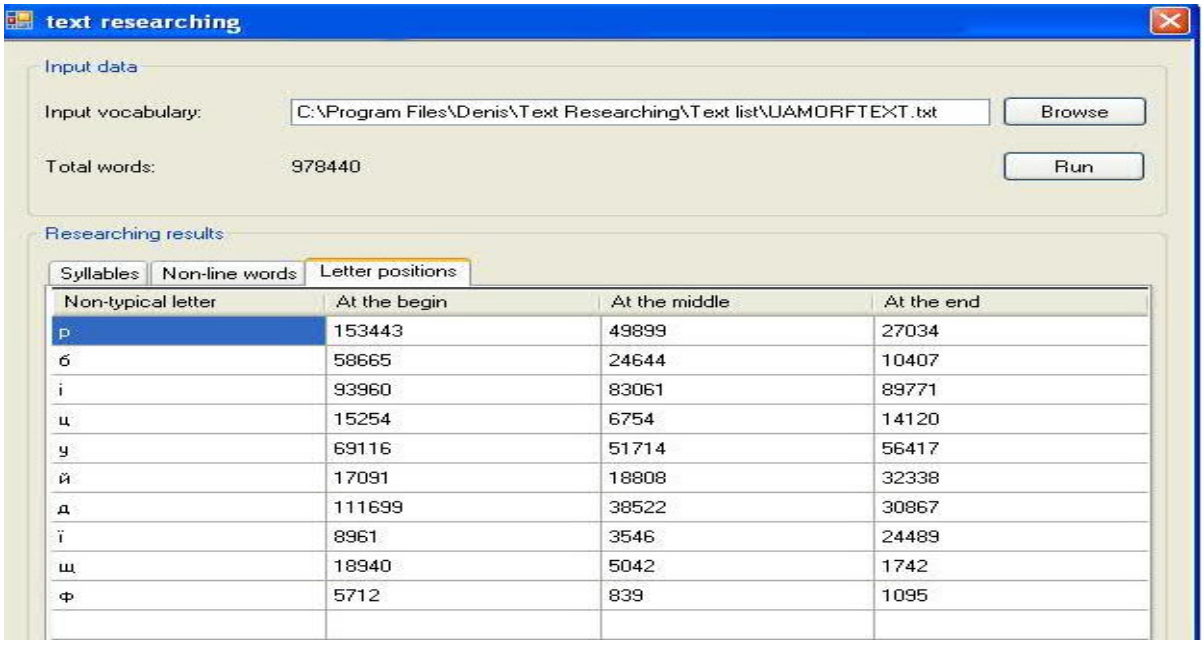


Рисунок 3.17 – База даних знаходження букви на початку, середині чи кінця слова

3.7 Висновки

В даному розділі розроблено програмне забезпечення для оцінка інформативності ознак текстових образів. Така оцінка проводилася з метою визначення ефективності застосування ознак в вузлах дерева рішень запропонованої ієрархічної процедури розпізнавання. В якості графічних образів (графем) тексту, які можуть бути використані для попереднього розпізнавання, було запропоновано використовувати на лексичному рівні графеми слів і морфем з тих міркувань, що перші можна легко сегментувати в зображенні, а другі представляють собою скінченну множину стійких до спотворень змістовних одиниць інформації. Для дослідження в якості ознак були вибрані довжини слів і надрядкові і підрядкові особливості слів і морфем, що задаються графікою написання окремих слів і морфем. Результати цих досліджень показали, що в окремих випадках дані ознаки можуть звузити пошук альтернатив для етапу розпізнавання графічних зображень в 3-5 разів.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки методів та дослідження моделей обробки текстових документів в інтелектуальних системах автоматизації.

Для проведення технологічного аудиту було залучено 3-х незалежних експертів Вінницького національного технічного університет, кафедри комп'ютерних систем та управління: Ковтун В. В. – к.т.н., доцент, доцент кафедри, Ковалюк О.О. – к.т.н., доцент, доцент кафедри, Юхимчук М.С. – к.т.н., доцент, доцент кафедри. За допомогою таблиці 4.1 за п'ятибальною шкалою використовуючи 12 критеріїв оцінки комерційного потенціалу розробки експерти надали свої оцінки.

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів

Продовження табл. 4.1

5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Таблиця 4.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

В таблиці 4.3 наведено результати оцінювання експертами комерційного потенціалу розробки.

Таблиця 4.3 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	Ковтун В.В.	Ковалюк О.О.	Юхимчук М.С.
	Бали, виставлені експертами:		
1	2	2	1
2	3	2	3
3	1	2	1
4	4	5	5
5	1	1	1
6	4	5	4
7	0	0	0
8	5	5	5
9	4	5	4
10	5	4	5
11	5	5	5
12	1	2	1
Сума балів	$СБ_1$=35	$СБ_2$=38	$СБ_3$=35
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_1^3 СБ_i}{3} = \frac{35+38+35}{3} = 37$		

Середньоарифметична сума балів, розрахована на основі висновків експертів склала 37, що згідно таблиці 4.2 вважається, що рівень комерційного потенціалу розробки є вище середнього.

Дана програма може бути використана в системах розпізнавання образів модульного типу, які можуть розв’язувати більш складні задачі шляхом підключення нових модулів.

Порівняємо програму, яка розробляється з аналогами, які існують на ринку. В якості аналога для розробки було обрано програмний засіб Fine Reader.

Основними недоліками аналога є:такі технології в своїх історичних витоках орієнтовані на брак апаратних ресурсів (швидкодії і пам’яті), не враховують технічних можливостей сучасних обчислювальних систем і мікропроцесорних засобів.

Також до недоліків можна віднести: не використовують мовних складових в інформаційній ієрархії текстового документа

У розробцідля уникнення даних недоліків пропонується інформаційна технологія електронізації текстів, яка передбачає їх введення і опрацювання за допомогою розподіленої комп’ютерно-мікропроцесорної системи.

В таблиці 4. 4 наведено порівняння аналога і нової розробки.

Таблиця 4.4 – Порівняння технічних характеристик нового методу та аналога

Технічні показники	Новий метод	Аналог	Відношення
Точність, %	99,8	75	1,3
Надійність, %	99	70	1,4
Величина затрати часу, хв	5	90	18

Отже, з таблиці видно, що по всім технічним характеристикам новий розроблюваний метод має набагато кращі характеристики, що свідчить про доцільність його подальшої розробки та впровадження.

Проведемо оцінку якості продукції, яка є найефективнішим засобом забезпечення вимог споживачів та порівняємо її з аналогом.

Визначимо відносні одиничні показники якості по кожному параметру за формулами (4.1) та (4.2) і занесемо їх у відповідну колонку табл. 4.5.

$$q_i = \frac{P_{Hi}}{P_{Бi}} \tag{4.1}$$

або

$$q_i = \frac{P_{Бi}}{P_{Hi}} \tag{4.2}$$

де P_{Hi}, $P_{Бi}$ – числові значення i-го параметру відповідно нового і базового виробів.

Таблиця 4.5 – Основні параметри нової розробки та товару-конкурента

Показник	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (товар-конкурент)	Новий (інноваційне рішення)		
1	2	3	4	5
Точність, %	75	99,8	1,3	45%
Надійність, %	70	99	1,4	10%
Величина затрати часу, хв..	90	5	18	45%

$$q_1 = \frac{99,8}{75} = 1,3;$$

$$q_2 = \frac{99}{70} = 1,4;$$

$$q_3 = \frac{90}{5} = 18.$$

Відносний рівень якості нової розробки визначаємо за формулою:

$$К_{я.в.} = \sum_{i=1}^{n} q_i \cdot \alpha_i , \quad (4.3)$$

$$К_{я.в.} = 1,3 \cdot 0,45 + 1,4 \cdot 0,1 + 18 \cdot 0,45 = 8,83$$

Відносний коефіцієнт показника якості нової розробки більший одиниці, отже нова розробка якісніший базового товару-конкурента.

Наступним кроком є визначення конкурентоспроможності товару. Конкурентоспроможність товару є головною умовою конкурентоспроможності підприємства на ринку і важливою основою прибутковості його діяльності.

Однією із умов вибору товару споживачем є збіг основних ринкових характеристик виробу з умовними характеристиками конкретної потреби покупця. Такими характеристиками найчастіше вважають нормативні та технічні параметри, а також ціну придбання та вартість споживання товару.

Приблизна ціна нового товару складе 1700 грн. Занесемо ці та інші показники (взяті з попередніх розрахунків) до табл. 4.6.

Таблиця 4.6 – Нормативні, технічні та економічні параметри інноваційного рішення і товару-виробника

Показники	Варіанти	
	Базовий (товар-конкурент)	Новий (інноваційне рішення)
1	2	3
1. Нормативно-технічні показники		
Точність, %	75	99,8
Надійність, %	70	99
Величина затрати часу, хв..	90	5
2. Економічні показники		
Ціна придбання, грн	2800	1700

Загальний показник конкурентоспроможності інноваційного рішення (К) з урахуванням вищезазначених груп показників можна визначити за формулою:

$$K = \frac{I_{m.n.}}{I_{e.n.}}, \tag{4.4}$$

де $I_{m.n.}$ – індекс технічних параметрів; $I_{e.n.}$ – індекс економічних параметрів.

Індекс технічних параметрів є відносним рівнем якості інноваційного рішення. Індекс економічних параметрів визначається за формулою (4.5)

$$I_{e.n.} = \frac{\sum_{i=1}^{n} P_{Hei}}{\sum_{i=1}^{n} P_{Бei}}, \tag{4.5}$$

де P_{Hei}, $P_{Бei}$ – економічні параметри (ціна придбання та споживання товару) відповідно нового та базового товарів.

$$I_{e.п.} = \frac{1700}{2800} = 0{,}60;$$

$$K = \frac{8{,}83}{0{,}60} = 14{,}71.$$

Зважаючи на розрахунки, можна зробити висновок, що нова розробка буде конкурентоспроможніше, ніж конкурентний товар.

4.2 Прогнозування витрат на виконання науково-дослідної роботи

1. Основна заробітна плата – винагорода за виконану роботу відповідно до встановлених норм праці. Вона встановлюється у вигляді тарифних ставок (окладів) і відрядних розцінок для робітників та посадових окладів для службовців. Стаття «Основна заробітна плата робітників» містить витрати на виплату основної заробітної плати робітникам, зайнятим виробництвом продукції.

Основна заробітна плата кожного із розробників (дослідників) *З* розраховується за формулою:

$$З = \frac{M}{T_P} \cdot t \ ,[грн.] \ , \tag{4.6}$$

де *М* – місячний посадовий оклад конкретного розробника.

T_P – число робочих днів, $T_р = 22$;

t – число днів роботи розробника.

Розрахунки основної заробітної плати зведемо в таблицю 4.7:

Таблиця 4.7 – Розрахунок основної заробітної плати розробників

Найменування посади	Місячний посадовий оклад,грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	9000	409	5	2159
Програміст	6700	304	30	9136
Всього				11295

2. До статті «Додаткова заробітна плата» відносяться витрати на виплату виробничому персоналу підприємства додаткової заробітної плати за працю понад установлені норми, заохочувальні виплати за поточну виробничу діяльність, компенсаційні виплати тощо $З_д = 11\% \cdot (10341) = 1137,5(грн.)$.. Звичайно, ці витрати встановлюються у відсотках до основної заробітної плати на підставі відповідних розрахунків на підприємстві:

$$З_д = 11\% \cdot З_\Sigma \ , \tag{4.7}$$

$$З_д = 11\% \cdot (11295) = 1242,4(грн.) \ .$$

3. Витати на соціальні заходи виникають внаслідок здійснення обов’язкової сплати єдиного внеску на загальнообов’язкове державне соціальне страхування.

Відрахування на соціальні заходи здійснюється від суми всіх витрат на оплату праці робітників, зайнятих безпосередньо виробництвом продукції:

$$B_{сз} = (З_{\Sigma} + З_{д}) \cdot \frac{\beta}{100\%}, \quad (4.8)$$

де в – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, %.

З 1.01.2016 року ставка єдиного внеску на загальнообов'язкове державне соціальне страхування встановлена залежно від класу професійного ризику виробництва і для бюджетної сфери β=22,0% .

$$В_{сз} = (11295 + 1242,4) \cdot \frac{22,0\%}{100\%} = 2758,2\,(грн.)$$

4. У спрощеному вигляді амортизаційні відрахування у загальному можуть бути розраховані за формулою:

$$A = \frac{Ц \cdot Т}{Ткор \cdot 12} \quad [грн], \quad (4.9)$$

де Ц – балансова вартість даного виду обладнання (приміщень), грн.;

$Т_{кор}$ – час користування;

Т – термін використання обладнання (приміщень), цілі місяці.

Згідно пункта 137.3.3 Податкового кодекса амортизація нараховується на основні засоби вартістю понад 2500 грн.

$$А = \frac{12000 \cdot 1}{2 \cdot 12} = 500 \text{ грн.}$$

5. Норма витрат матеріалу – це плановий показник, який визначає максимально допустимі затрати відповідних ресурсів на виробництво одиниці продукції в умовах певного рівня техніки і організації виробництва.

Витрати на матеріали М, що були використані під час виконання даного етапу роботи, розраховуються по кожному виду матеріалів за формулою:

$$М = \sum_{1}^{n} Н_i \cdot Ц_i \cdot К_i - \sum_{1}^{n} В_i \cdot Ц_в \text{ грн.,} \quad (4.10)$$

де $Н_i$ – витрати матеріалу *і*-го найменування, кг;

$Ц_i$ – вартість матеріалу *і*-го найменування, грн./кг.;

$К_i$ – коефіцієнт транспортних витрат, $К_i$ = (1,1…1,15);

$В_i$ – маса відходів матеріалу *і*-го найменування, кг;

$Ц_в$ – ціна відходів матеріалу *і*-го найменування, грн/кг;

n – кількість видів матеріалів.

Інформацію про використані матеріали подамо у вигляді табл. 4.8.

Таблиця 4.8 – Матеріали, що використані на розробку

Найменування матеріалу	Ціна за одиницю, грн.	Витрачено	Вартість витраченого матеріалу, грн.
Флешка	120	1	120
Папір	75	1	75
Ручка	10	1	10
CD диск	10	1	10
Всього			215
З врахуванням коефіцієнта транспортування			236,5

6. До статті «Паливо та енергія на технологічні цілі» відносяться витрати на всі види палива й енергії, що безпосередньо використовуються у процесі виробництва продукції. У даному випадку будемо враховувати лише витрати на електроенергію. Витрати на енергію визначаються на основі витрат на одиницю продукції та тарифів на енергію за допомогою залежності:

$$В_е = В \cdot П \cdot Ф \cdot К_п, \quad (4.11)$$

де В – вартість 1 кВт енергії, грн. В = 8,44 грн/кВт*год;

П – установлена потужність обладнання, кВт. При паяні використовується паяльник потужність П =400 Вт або П = 0,4 кВт;

Ф – фактична кількість годин роботи обладнання, год. Ф = 100 год;

$К_п$ – коефіцієнт використання потужності, $К_п$ = 0,65.

$В_е = 8,44 \cdot 0,4 \cdot 100 \cdot 0,65 = 219,44 (грн)$.

7. Інші витрати $B_{ін}$ охоплюють: витрати на управління організацією, оплата службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо.

Інші витрати $B_{ін}$ можна прийняти як (100…300)% від суми основної заробітної плати розробників та робітників, які виконували дану МКНР, тобто:

$$B_{ін} = (1..3) \cdot (З + З_р). \tag{4.12}$$

$$B_{ін} = 1 \cdot (11295) = 11295 (грн.)$$

Сума всіх попередніх статей витрат дає витрати, які безпосередньо стосуються даного розділу МКНР

В = 11295+1242,4+2525,3+500+236,5+219,44+11295=27313,6 грн.

Загальна вартість всієї МКНР визначається за формулою:

$$B_{заг} = \frac{B}{\alpha} \tag{4.13}$$

$$B_{заг} = \frac{27313,6}{1} = 27313,6 (грн.)$$

Прогнозування загальних втрат ЗВ на виконання та впровадження результатів виконаної МКНР здійснюється за формулою:

$$ЗВ = \frac{B}{\beta}, \tag{4.14}$$

де $\beta -$ коефіцієнт, який характеризує стадію виконання даної НДР.

Оскільки, робота знаходиться на стадії розробки дослідного зразка, то коефіцієнт β = 0,9.

Звідси:

$$ЗВ = \frac{27313,6}{0,9} = 30348 (грн.).$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

У даному підрозділі кількісно спрогнозуємо, яку вигоду, зиск можна отримати у майбутньому від впровадження результатів виконаної наукової роботи. Розрахуємо збільшення чистого прибутку підприємства $ДП_i$, для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, за формулою

$$\Delta П_i = \sum_{1}^{n} \left(\Delta Ц_o \cdot N + Ц_o \cdot \Delta N \right)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\upsilon}{100} \right) \qquad (4.15)$$

де $\Delta Ц_0$ – покращення основного оціночного показника від впровадження результатів розробки у даному році.

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки:

Цо – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки:

л – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт *л* = 0,8333.

р – коефіцієнт, який враховує рентабельність продукту. р = 0,25;

х – ставка податку на прибуток. У 2019 році – 18%.

Припустимо, що при впроваджені результатів наукової розробки покращується якість, що дозволяє підвищити ціну його реалізації на 300 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року на 100 шт., протягом другого року – на 130 шт., протягом третього року на

170 шт. Реалізація продукції до впровадження розробки складала 1 шт., а її ціна 1700 грн. Розрахуємо прибуток, яке отримає підприємство протягом трьох років.

$$\Delta П_1 = \left[300 \cdot 1 + (1700 + 300) \cdot 100\right] \cdot 0,833 \cdot 0,25 \cdot \left(1 + \frac{18}{100}\right) = 37736,2 (\text{грн.})$$

$$\Delta П_2 = \left[300 \cdot 1 + (1700 + 300) \cdot (100 + 130)\right] \cdot 0,833 \cdot 0,25 \cdot \left(1 + \frac{18}{100}\right) = 81191 (\text{грн.})$$

$$\Delta П_3 = \left[300 \cdot 1 + (1700 + 300) \cdot (100 + 130 + 170)\right] \cdot 0,833 \cdot 0,25 \cdot \left(1 + \frac{18}{100}\right) = 122651 (\text{грн.})$$

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Розрахуємо основні показники, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності. Теперішню вартість інвестицій PV, що вкладаються в наукову розробку приймемо рівну загальним витратам PV = ЗВ $= 30348$ грн.

Розрахуємо абсолютну ефективність вкладених інвестицій $E_{абс}$ згідно наступної формули:

$$E_{абс} = (ПП - PV) \tag{4.16}$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство від реалізації результатів наукової розробки, грн;

$$ПП = \sum_{1}^{T} \frac{\Delta П_i}{(1 + \tau)^t}, \tag{4.17}$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДЦКР, грн;

Т – період часу, протягом якою виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,2;

t – період часу (в роках).

$$ПП = \frac{37736,2}{(1+0,2)^1} + \frac{81191}{(1+0,2)^2} + \frac{122651}{(1+0,2)^3} = 151791,93(грн.).$$

$$E_{абс} = (151791,93 - 30348) = 121443,93(грн.).$$

Оскільки $E_{абс} > 0$ то вкладання коштів на виконання та впровадження результатів НДДКР може бути доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_в$. Для цього користуються формулою:

$$E_в = \sqrt[T_ж]{1 + \frac{E_{абс}}{PV}} - 1, \qquad (4.20)$$

$T_ж$ – життєвий цикл наукової розробки, роки.

$$E_в = \sqrt[3]{1 + \frac{121443,93}{30348}} - 1 = 1.2 = 120\%$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \qquad (4.18)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні d= (0,14…0,2);

f – показник, що характеризує ризикованість вкладень; зазвичай, величина f = (0,05...0,1).

$$\tau_{min} = 0,18 + 0,05 = 0,23.$$

Так як $E_в > \tau_{\min}$ то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_в} \tag{4.19}$$

$$T_{ок} = \frac{1}{1,2} = 0,83 \text{ (роки)}$$

Так як $T_{ок} \leq$ 3...5-ти років, то фінансування даної наукової розробки в принципі є доцільним.

4.5 Висновки до економічного розділу

В економічній частині було оцінено економічний потенціал розробки методів та дослідження моделей обробки текстових документів в інтелектуальних системах автоматизації.

. Наведено порівняння нової розробки з аналогом, яке показало, що новий виріб буде набагато краще за аналог по технічним і економічним параметрам.

Оцінка якості і конкурентоспроможності показали, що нова розробка є більш якісною і конкорунтноспроможнішою ніж аналоги, які існують на ринку.

Прогнозування витрат на виконання науково-дослідної роботи по кожній з статей витрат складе 27313,6 грн. Загальна ж величина витрат на виконання та впровадження результатів даної НДР буде складати 30348 грн.

Вкладені інвестиції в даний проект окупляться через 0,83 роки при прогнозованому прибутку 122651 грн. за три роки.

ВИСНОВКИ

Магістерська кваліфікаційна робота присвячена розробці інтелектуальних методів і дослідженню моделей обробки текстових документів в системах автоматизації.

Проведений у вступній частині аналіз показав, що розпізнавання тексту є на сьогодні дуже актуальною темою, над якою працюють науковці та інженери багатьох країн світу. При цьому використовуються різні підходи до розпізнавання зображень та виділення інформативних ознак.

В першому розділі проведено варіантний аналіз методів обробки тексту та обгрунтування оптимального варіанту вирішення задачі, поставленої в технічному завданні.

В другому розділі розроблено структурну схему системи та математичну модель процесу розпізнавання тексту і визначено критерії оцінки його ефективності, представлено математичні основи визначення інформаційної структури тексту і його морфологічних характеристик, розроблено метод автоматичного виділення морфем в тексті. Проведено дослідження низки лексичних і морфологічних ознак, яке виявило їх високу інформативність для розпізнавання мовної і графічної інформації, що міститься в зображенні тексту. Проведений інформаційний аналіз показав, що найбільш значимою інформацією для розпізнавання і розуміння текстового документу володіють такі його елементи, як графічні особливості напису окремих слів, графеми визначених літер, морфем, довжина складів і окремих літер, статичні і перехідні ймовірності вказаних елементів. Запропоновано використовувати послідовно-паралельну стратегію розпізнавання у вигляді дерева рішень, яка дозволяє підвищити точність і швидкість розпізнавання, а також використовувати ієрархічно-модульний принцип побудови програмних засобів для задач різної складності розпізнавання.

В третьому розділі роботи розроблено алгоритми та програмне забезпечення для створення бази даних морфем української мови, алгоритм

визначення ймовірностей переходів між морфемами корпусу тексту, а також алгоритми дослідження інформативності ознак в тексті. Для попереднього розпізнавання графічних елементів (графем) текстового документу, було запропоновано використовувати на лексичному рівні графеми слів і морфем, оскільки перші можна нескладно виділити в зображенні, а другі входять до скінченної множини стійких до спотворень змістовних складових інформації. В якості ознак досліджувалися такі параметри, як довжини слів і надрядкові і підрядкові особливості слів і морфем, що задаються графікою написання в них окремих літер. За тестування програмного забезпечення було встановлено, що в багатьох випадках використання таких ознак звужує пошук альтернатив в процедурі розпізнавання в 3-5 разів.

Розроблене в магістерській дипломній роботі програмне забезпечення реалізоване на базі платформи Microsoft.NET Framework 2.0, за мову програмування був обраний C# .

В економічній частині доведено, що розроблений програмний продукт є економічно вигідним і тому має великі перспективи до широкого розповсюдження.

СПИСОК ЛІТЕРАТУРИ

1. Биков М.М., Калінчук Р.С. Методи підвищення ефективності процесу розпіз навання тексту / М.М. Биков, Р.С. Калінчук. -

2. Биков М. М. Використання інтелектуальних методів в розпізнаванні символів / М. М. Биков, Д. Є. Балховський, А. Раїмі // Інформаційні технології та комп'ютерна інженерія. – 2007. – № 2 (9). – С. 121 – 125.

3. Методи розпізнавання тексту.- [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/методи розпізнавання тексту.

4. Репік С. І., Штогріна О. С. Методи розпізнавання тексту / С.І. Репік, О.С. Штогріна // Збірник матеріалів Міжнародної науково-технічної конференції «ПЕРСПЕКТИВИ ТЕЛЕКОМУНІКАЦІЙ», [S.l.], nov. 2016. – [Електронний ресурс]. Режим доступу: <http://conferenc.its.kpi.ua/proc/article/view/71101>.

5. Forsyth D. A. Computer Vision: A Modern Approach / D. A. Forsyth, J. Ponce. – Pearson Education, Inc., 2011. – 792 p.

6. Ту Д., Гонсалес Р. Принципы распознавания образов. – М.: Мир, 1978. – С. 411.

7. П. Г. Катыс, Г. П. Катыс. Автоматическая обработка и интерпретация изображений. - МИРЭА – М.: Частина I, 1998. – 210 с.

8. Федотов Н.Г. Теорія ознак розпізнавання зображень, заснована на стохастической геометрії / Н.Г. Федотов // Тези доповідей на 3-їй конф-ии "Розпізнавання образів і аналіз зображень: нові інформаційні технології (РОАИ-97)". – Нижній Новгород, РАНЕЙ. - Нижегородський державний університет ім. Н.И. Лобачевского, 1997. – С.278-282.

9. Федотов Н. Г. Методи стохастической геометрії в розпізнаванні образів / Н.Г. Федотов. - М: Радіо і Зв'язок, 1990. – 250 с.

10. Афонасенко, А. В. Обзор методов распознавания структурированных символов / А.В. Афонасенко, А.И. Елизаров // Доклады Томского государственного университета систем управления и радиоэлектроники. —2008. - Вып. 2(18). -Ч.1.-С.83-88.

11. Фу К. Структурные методы в распознавании образов / К. Фу. – М.: Мир, 1976. – С. 284.

12. Wang H., Bengio S.The MNIST Database of Handwritten upper– case letters. – [Електронний ресурс] – Режим доступу: htttp://www.idiap.ch/home.html.

13. Потапов А.С. Распознавание образов и машинное восприятие: Общий подход на основе принципа минимальной длины описания / А.С. Потапов - СПб.: Политехника, 2007. — 548 с.

14. В.О. Козел. Методи та етапи автоматичного розпізнавання тексту / Козел В.О. // Вісник Черкаського університету. - Випуск 172. Серія прикладна математика. Інформатика С.75-86.

15. Анисимов Б.В., Курганов В.Д., Злобин В.К.– Распознавание и цифровая обработка изображений. – М. – 1983. – С.35– 68.

16. Суясов Д. И. Выделение структурных признаков изображений символов на основе клеточных автоматов с метками / Д. И. Суясов // Информационно-управляющие системы, 2010. – № 4. – С. 39–45. – [Электронный ресурс]. - Режим доступа: http://cyberleninka.ru/article/n/vydelenie-strukturnyh-priznakovizobrazheniy-simvolov-na-osnove-kletochnyh-avtomatov-smetkami

17. Trier S. D., Jain A.K., Taxt T. Feature extraction methods for character recognition – A Survey. // Pattern recognition, 1996. – Vol. 29, No. 4. – P. 641– 642.

18. Кульчицька І.О. Особливості алгоритмів бінаризації зображень документів/ І.О.Кульчицька, О.В. Тимченко // Зб. наук. пр. ІПМЕ НАН України.— Вип.68. — К.:2013. — С.141–149.

19. К. Лекун, К. Берджес. MNIST рукописные цифры базы данных. – [Електронний ресурс]. - Режим доступу: http://yann.lecun.com/exdb/mnist/

20. Пиотровский Р. Г. Текст машина, человек / Р.Г. Пиотровский — Ленинград: Наука”, 1975. – 326 с.

21. Пиотровский Р.Г. Математическая лингвистика / Р.Г. Пиотровский, К.Б. Бектаев, А.А. Пиотровская. – М.: Высшая школа, 1977. – 384 с.

22. Пиотровский Р.Г. Инженерная лингвистика и теория язика / Р.Г. Пиотровский. – Л.: Наука, 1980. – 260 с.

23. Баранов А.Н. Введение в прикладную лингвистику / А.Н. Баранов. – М.: Высшая школа, 2001. – 326 с.

24. Алексеев П.М. Статистическая лексикография (типология, составление и применение частотных словарей / П.М. Баранов. – Л. Наука, 1975. – 179 с.

25. Апресян Ю.Д. Лексическая семантика / Ю.Д. Апресян. – М.:, Высшая школа 1974. – 245с.

26. Арутюнова Н.Д. Типы языковых значений: Оценка. Событие. Факт / Н.Д. Арутюнов – М.: Высшая школа, 1988. – 214 с.

27. Белоногов Г.Г., Кузнецов Б.А. Языковые средства автоматизированных информационных систем / Г.Г. Белоногов, Б.А. Кузнецов. – М.:Наука, 1983. – 288 с.

28. Bykov N.M., Kuzmin I.V., Yakovenko A.I. Development of effective strategy of pattern recognition // Proceedings of SPIE, 2000, Vol.4425, pp. 76– 82.

29. Быков Н.М., Агеев А.С. Модель потенциальной системы для распознавания речи в СЧМ. – В кн.: Исследование и проектирование систем “человек– машина” / Н.М. Быков, А.С. Агеев. – Киев: ИК АНН УССР, 1986. – С. 57– 61.

30. Биков М.М., Грищук Т.В. Розробка методів оцінки ефективності автоматизованих систем розпізнавання мови // Вісник Технологічного університету Поділля – Хмельницький, ТУП, 2003. – №3, том 1. – С. 122– 125.

31. Бутенков С.А., Семерий О.С. Оптимизационный метод распознаванния изображений с помощью аналитических моделей в параллельных системах / С.А. Бутенков, О.С. Семарий // Сб. трудов Междун. конф. “Интеллектуальные многопроцессорные системи-99” – Таганрог, 1999. – 250 с.

32. Биков М.М. Оцінка інформативності графем тексту / М. М. Биков, Д. Є. Балховський, В. В. Ковтун // - Вісник Вінницького політехнічного інституту, 2009, № 2, с. 95 – 98.

33. Ефективна технологія електронізації документів в автоматизованих інформаційно-пошукових системах / Д.Є. Балховський, Т.В. Грищук, М.М. Биков

// Оптико-електронні інформаційно-енергетичні технології. — 2008. — № 2 (16). — С. 9-13.

34. Биков М.М. Оцінка інформативності графем тексту / М. М. Биков, Д. Є. Балховський, В. В. Ковтун // - Вісник Вінницького політехнічного інституту, 2009, № 2, с. 95 – 98.

35. Козловський В.О. Техніко економічне обґрунтування та економічні розрахунки в дипломних проектах та роботах. Навчальний посібник / В.О. Козловський. - Вінниця: ВДТУ, 2003. – 75 с.

36. Ковалевский В.А. Методы оптимальных решений в распознавании изображений. – М.: Наука, 1976. – 328с.

37. Быков Н.М., Данилюк Ю.С., Ильницкий Н.П. - Анализ эффективности средств распознавания в системах принятия решений.- / В кн.: Эффективность и надежность автоматизированных систем принятия решений.-Киев: ИК АН УССР, 1982.- С. 24 - 29.

38. Чумаков Н.М., Серебряный Е.И. Оценка эффективности сложных технических устройств / Н.М. Чумаков, Е.И. Серебряный. - М.: Советское радио,1982.-192 с.

39. Ackoff R.L. Scientific Method. Optimizing Applied Research Decisions / R.L. Ackoff. – New Yourk, 1962. – P. 150.

40. Вентцель Е.С. Исследование операций: задачи, принципы, методология / Е.С. Вентцель. - 2-е изд. — М.: Наука, 1988. — 208 с.

41. Основы моделирования сложных систем : учебник для вузов / под ред. И. В. Кузьмина. – К. : Вища школа, 1981. – 359 с. – Киев.: Вища школа, 1981. – 360 с.

42. Кузьмін І.В. Основи теорії інформації і кодування: [Підручник] / І.В.Кузьмін, І.В.Троцишин, А. І. Кузьмін,В. О. Кедрус,В. Р. Любчик; За ред.Іван Васильович Кузьмін.–3–тє вид.–Хмельницький : ХНУ, 2009.–373 с.

43. Вентцель Е.С. Теория вероятностей / Е.С. Вентцель. - 7-е изд. — Москва: Высшая школа, 2001. — 575 с.

44. Бикова К.М., Гаврищук В.Л., Биков М.М. Аналіз сучасного стану проблеми машинного перекладу мови // Матеріали IV міжнародної конференції студентів та молодих вчених.- С.27.- Київ: Національний авіаційний університет.- 2004.- 416 с.

45. Creutz M., Lagus K. Unsupervised Morpheme Segmentation and Morphology Induction from Text Corpora Using Morfessor 1.0. [Електронний ресурс]. – Режим доступу: http://www.cis.hut.fi/mcreutz/papers/Creutz04tr.pdf.

46. Кристиан Нейгел, Билл Ивьен, Джей Глинн, Карли Уотсон, Морган Скиннер C# 2005 и платформа .NET 3.0 для профессионалов. – М.: «Диалектика», 2007. – 956 с.

47. Эндрю Троелсен. Язык программирования C# 2005 (Си Шарп) и платформа .NET 2.0. – 3– е изд. / Э. Троелсен. – М.: «Вильямс», 2007. – С. 1168.

ДОДАТКИ

Додаток А

(обов’язковий)

ВНТУ

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

Розробка методів та дослідження моделей обробки текстових документів в інтелектуальних системах автоматизації

Студент групи 2АКІТ-18м Калінчук Р.С.

“___” _____ 2019 р.

Керівник к.т.н., доцент Биков М.М.

“___” _____ 2019 р.

Вінниця 2019

1. Назва та галузь застосування

1.1. Назва – Розробка методів та дослідження моделей обробки текстових документів в інтелектуальних системах автоматизації.

1.2. Галузь застосування – системи управління і автоматики, інформаційні технології.

2. Підстава для проведення розробки.

Тема магістерської дипломної роботи затверджена наказом по ВНТУ № _254_ від “_2_” _10_ 2019 р.

3. Мета та призначення розробки.

Метою магістерської кваліфікаційної роботи є підвищення ефективності процесу розпізнавання текстових документів.

4. Вихідні дані для проведення розробки.

Магістерська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

- Биков М.М., Балховський Д.Є. .Лінгвістико – математичний метод опису символів / М.М. Биков, Д.Є. Балховський. – Кн.: Матеріали IV Міжнародної конференції студентів і молодих вчених. – К.: НАУ, 2004. – С.26
- Bykov N.M., Kuzmin I.V., Yakovenko A.I. Development of effective strategy of pattern recognition // Proceedings of SPIE, 2000, Vol.4425, pp. 76– 82.
- Пиотровский Р.Г. Инженерная лингвистика и теория язика / Р.Г. Пиотровский. – Л.: Наука, 1980. – 260 с.
- Анисимов Б.В., Курганов В.Д., Злобин В.К. Распознавание и цифровая обработка зображений / Б.В. Анисимов, В.Д. Курган, В.К. Злобин. – М. – 1983. – С.35– 68.

5. Вимоги до розробки.

5.1. Перелік головних функцій:

– завантаження файлу з текстовим документом
– морфологічний аналіз морфем української мови, створення бази даних морфем української мови;
– створення бази даних статистичних та перехідних характеристик морфем української мови;
– оцінка інформативності графічник характеристик тексту.

5.2. Основні технічні вимоги до розробки.

5.2.1. Вимоги до програмної платформи:

- WINDOWS 7\8;
- Matlab 7.0;
- Microsoft Visual Studio.

5.2.2. Умови експлуатації системи:

- робота на стандартних ПЕОМ в приміщеннях зі стандартними умовами;
- можливість цілодобового функціонування системи;
- текст програмного забезпечення системи є цілком закритим.

6. Економічні показники

До економічних показників входять:

– термін окупності не більше _0.83_ років;
– розмір капіталовкладень не більше _30348_ грн..;
– економічний ефект не менше _122651_ грн..;
– рентабельність не менше _18_ %;
– інші економічні переваги у порівнянні з аналогами.

7. Стадії та етапи розробки.

7.1 Пояснювальна записка:

1	Вступ, варіантний аналіз методів обробки тексту та обґрунтування обраного варіанту.	02.10.2019 р.
2	Постановка задачі і розробка технічного завдання	04.10.2019 р.
3	Розробка і дослідження ефективних методів і	20.10.2019 р.

моделей електронізації текстових документів

4	Розробка алгоритмів та програмного забезпечення	02.11.2019 р.
5	Проведення експериментальних досліджень.	15.11.2019 р.
6	Підготовка економічної частини	22.11.2019 р.
7	Оформлення пояснювальної записки, графічного матеріалу і презентації	30.11.2019 р.
6	Апробація результатів дослідження. Публікації.	06.12.2019 р.
9	Попередній захист	07.12.2019 р.
10	Захист МКР	12.12.2019 р.

7.2 Графічні матеріали:

– структурна схема системи: «02 » грудня 2019 р.

– схеми алгоритмів: «04 » грудня 2019 р.

– Графічні інтерфейси програм: «05 » грудня 2019 р.

– плакати демонстраційні: «05 » грудня 2019 р.

8. Порядок контролю і приймання.

8.1. Хід виконання магістерської кваліфікаційної роботи контролюється керівником роботи, консультантами з економічної частини. Рубіжний контроль провести до. «06» грудня 2019 р

8.2. Атестація проекту здійснюється на попередньому захисті. Попередній захист магістерської кваліфікаційної роботи провести до 07» грудня 2019 р.

8.3. Підсумкове рішення щодо оцінки якості виконання магістерської кваліфікаційної роботи приймається на засіданні ДЕК. Захист магістерської кваліфікаційної роботи провести «12» грудня 2019 р.

Додаток Б

Лістинги процедур моделей автоматизованої обробки тексту

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Drawing;
using System.IO;
using System.Text;

namespace fontsize
{
	/// <summary>
	/// Class that executes preparing of input text for next using
	/// </summary>
	public class TextHandler
	{
		#region Class members

		private static TextHandler m_Instance = null;

		private string m_fileName = string.Empty;

		private Graphics m_graphics = null;

		#endregion

		#region Initialization/Constructors

		/// <summary>
		/// Static Constructor
		/// </summary>
		static TextHandler()
		{
		}

		/// <summary>
		/// Constructor
		/// </summary>
```

```
        protected TextHandler(string fileName, Graphics g)
        {
              m_fileName = fileName;
              m_graphics = g;
        }

        #endregion

        #region Properties

        /// <summary>
        /// Gets an single instance od <c>TextHandler</c> object.
        /// Create should be called before</summary>
        public static TextHandler Instance
        {
              get { return m_Instance; }
              set { m_Instance = value; }
        }

        #endregion

        #region Static Methods

        /// <summary>
        ///    Creates    a    singltone    instance    of    <c>DataHandler</c>
object</summary>
        /// <param name="args">
        /// Morfessor Arguments to optimize data processing</param>
        public static TextHandler Create(string fileName, Graphics g)
        {
              m_Instance = new TextHandler(fileName, g);
              return m_Instance;
        }

        #endregion

        #region Methods

        /// <summary>
        /// Prepares text for researching
        /// </summary>
        public string[] Prepare(Encoding encoding)
```

```
        {
            if (m_Instance == null)
                throw new ArgumentNullException("m_Instance");

            if (string.IsNullOrEmpty(m_fileName))
                throw new ArgumentNullException("m_fileName");

            return GetWords(m_fileName, encoding);
        }


        public void Run(
            ref Dictionary<int, List<float>> syllablesTestResults,
            ref Dictionary<char, int> nonLineLettersTestResults,
            ref Dictionary<char, int> nonLineWordsTestResults
            )
        {
            List<ResearchingArgs> syllablesTestList = new
List<ResearchingArgs>();
            List<ResearchingArgs> nonLineWordsTestList = new
List<ResearchingArgs>();

            string[] words = Prepare((Encoding.UTF8));

            foreach (string word in words)
            {
                //Syllables test
                string trimmedWord = word.Trim();
                int syllables = GetWordSyllables(trimmedWord);
                float length = GetWordLentgh(trimmedWord);
                syllablesTestList.Add(new ResearchingArgs(syllables,
length));

                //Non-line words test
                foreach (char letter in word.ToLower())
                {
                    if (IsNonLineLetter(letter))
                    {
                        if
(nonLineLettersTestResults.ContainsKey(letter) == false)
                            nonLineLettersTestResults.Add(letter, 0);
                        nonLineLettersTestResults[letter]++;
```

```
                        }
                    }

                    foreach (char letter in NonLineLetters)
                    {
                        if (word.ToLower().Contains(letter.ToString()))
                        {
                            if (nonLineWordsTestResults.ContainsKey(letter)
== false)
                                nonLineWordsTestResults.Add(letter, 0);
                            nonLineWordsTestResults[letter]++;
                        }
                    }
                }

                syllablesTestResults                                    =
ProcessSyllablesTestResults(syllablesTestList);
            }

            #endregion

            #region Implementation

            private                 Dictionary<int,                 List<float>>
ProcessSyllablesTestResults(List<ResearchingArgs> resultList)
            {
                Dictionary<int,    List<float>>    resultDictionary    =    new
Dictionary<int, List<float>>();
                foreach (ResearchingArgs arg in resultList)
                {
                    int syllables = arg.WordSyllables;
                    if (resultDictionary.ContainsKey(syllables) == false)
                        resultDictionary.Add(syllables, new List<float>());
                    resultDictionary[syllables].Add(arg.WordLength);
                }
                return resultDictionary;
            }

            public static float EvaluateAverage(List<float> array)
            {
                float sum = 0;
                foreach (float arg in array)
```

```
                  sum += arg;
            return sum / array.Count;
      }

      private string AddQuotas(string line)
      {
            return "\"" + line + "\"";
      }

      private char[] GetSeparators()
      {
            byte[] bytes = new byte[byte.MaxValue];
            for (byte i = 0; i < byte.MaxValue; i++)
                  bytes.SetValue(i, i);
            string        str        =        Encoding.GetEncoding("windows-
1251").GetString(bytes);

            ArrayList array = new ArrayList();
            foreach (char c in str)
                  if (Char.IsLetter(c) == false)
                        array.Add(c);

            return (char[])array.ToArray(typeof(char));
      }

      private string[] SplitLine(string Line)
      {
            char[] charSeparators = GetSeparators();
            return                                  Line.Split(charSeparators,
StringSplitOptions.RemoveEmptyEntries);
      }

      private string[] GetWords(string fileName, Encoding encoding)
      {
            byte[] inData;
            using    (FileStream    file    =    new    FileStream(fileName,
FileMode.Open))
            {
                  BinaryReader bRead = new BinaryReader(file);
                  inData = bRead.ReadBytes((int)file.Length);
            }
            string inString = encoding.GetString(inData);
```

```
            return SplitLine(inString);
        }

        private void WriteWordsToFile(string fileName, string[] words)
        {
            using (StreamWriter sw = new StreamWriter(fileName, false))
            {
                foreach (string word in words)
                {
                    sw.WriteLine(word);
                }
            }
        }

        private float GetWordLentgh(string word)
        {
            FontFamily fontFamily = new FontFamily("Arial");
            Font font = new Font(fontFamily, 16, FontStyle.Regular,
GraphicsUnit.Pixel);

            return m_graphics.MeasureString(word, font).Width;
        }

        public int GetWordSyllables(string word)
        {
            if (string.IsNullOrEmpty(word))
                throw new ArgumentNullException("word");

            int syllables = 0;
            foreach (char letter in word.ToLower())
                if (IsOpenLetter(letter))
                    syllables++;

            return syllables;
        }

        private bool IsOpenLetter(char letter)
        {
            char[] openLetters = { 'а', 'е', 'є', 'и', 'і', 'ї', 'о', 'у',
'ю', 'я', 'a', 'e', 'i', 'o'};
```

```
            foreach (char openLetter in openLetters)
                  if (letter == openLetter)
                        return true;

            return false;
      }

      char[] NonLineLetters = { 'й', 'ц', 'у', 'щ', 'і', 'ї', 'ф', 'р', 'д',
'б' };

      private bool IsNonLineLetter(char letter)
      {
            foreach (char nonLineLetter in NonLineLetters)
                  if (letter == nonLineLetter)
                        return true;

            return false;
      }

      #endregion
   }
                                          }
```

Додаток В

Лістинг процедур методів сегментації тексту на інформативні елементи

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.IO;
using System.Runtime.InteropServices;
using System.Text;
using System.Windows.Forms;
using fontsize;

namespace TextResearching
{
    public class WordFormResearchingController
    {
        #region Fields

        private string fileName;
        private TextHandler textHandler;
        private List<string> wordList;
        private Graphics m_graphics;
        private PictureBox pictureBox;
        private Dictionary<int, int> wordsDictionary;
        #endregion

        #region Constants

        private const string dirNonFormatted = "c:\\Images\\NonFormatted";
        private const string dirFormatted = "c:\\Images\\Formatted";

        #endregion

        #region Constructor

        public    WordFormResearchingController(string    fileName,    PictureBox
pictureBox, Graphics graphics)
        {
```

```
            if (string.IsNullOrEmpty(fileName))
                throw new ArgumentNullException("fileName");

            this.fileName = fileName;
            this.pictureBox = pictureBox;
            m_graphics = graphics;
        }

        #endregion

        #region P/Invoke

        [DllImport("user32.dll")]
        private static extern int SendMessage(IntPtr hwnd, int msg, IntPtr hdc,
int flags);

        #endregion

        #region Public Methods

        public void Run()
        {
            Initialize();
            CreateWordFormArgs();
            CreateWordImages();
            FindWordsBySyllables();
        }

        #endregion

        #region Implementation

        private void Initialize()
        {
            textHandler = TextHandler.Create(fileName, null);
            wordList = new List<string>(textHandler.Prepare(Encoding.UTF8));
        }

        private void Draw(string word)
        {
            if (string.IsNullOrEmpty(word))
                return;
```

```
            string fontName = "Arial";
            int size = 20;

            FontFamily fontFamily = new FontFamily(fontName);
            Font font = new Font(fontFamily, size, FontStyle.Regular,
GraphicsUnit.Pixel);
            PointF wordSize = new PointF();
            wordSize.X = m_graphics.MeasureString(word, font).Width;
            wordSize.Y = m_graphics.MeasureString(word, font).Height;
            Bitmap bitmap = new Bitmap(Convert.ToInt32(wordSize.X),
Convert.ToInt32(wordSize.Y));
            pictureBox.Image = bitmap;
            m_graphics = Graphics.FromImage(pictureBox.Image);
            SolidBrush solidBrush = new SolidBrush(Color.Black);
            pictureBox.Size = new Size(Convert.ToInt32(wordSize.X),
Convert.ToInt32(wordSize.Y));
            pictureBox.BackColor = Color.White;
            m_graphics.DrawString(word, font, solidBrush, 0, 0);
        }

        private void Save(string fileName)
        {
            IntPtr hdc = m_graphics.GetHdc();
            try
            {
                SendMessage(pictureBox.Handle, 0x0317, hdc, 0x36);
            }
            finally
            {
                m_graphics.ReleaseHdc();
            }
            if (System.IO.Directory.Exists(dirNonFormatted) == false)
                System.IO.Directory.CreateDirectory(dirNonFormatted);
            pictureBox.Image.Save(Path.Combine(dirNonFormatted, fileName),
System.Drawing.Imaging.ImageFormat.Gif);
        }

        private void Shape(string imgPath, out List<int> XPixels, out List<int>
YPixels)
        {
            Bitmap srcImage = new Bitmap(imgPath);
            Bitmap img = ConvertToBinary(srcImage, 0.5);
```

```
List<Point> points = new List<Point>();
for (int i = 0; i < img.Width; i++)
{
    for (int j = 0; j < img.Height; j++)
    {
        if (img.GetPixel(i, j).Name == "ff000000")
        {
            points.Add(new Point(i, j));
        }
    }
}
XPixels = new List<int>();
YPixels = new List<int>();
foreach (Point point in points)
{
    XPixels.Add(point.X);
    YPixels.Add(point.Y);
}
XPixels.Sort();
YPixels.Sort();
int x1 = XPixels[0];
int x2 = XPixels[XPixels.Count - 1];
int y1 = YPixels[0];
int y2 = YPixels[YPixels.Count - 1];

Bitmap dstImage = new Bitmap(x2 - x1 + 1, y2 - y1 + 1);
for (int i = 0; i < (x2 - x1 + 1); i++)
{
    for (int j = 0; j < (y2 - y1 + 1); j++)
    {
        dstImage.SetPixel(i, j, img.GetPixel(i + x1, j + y1));
    }
}

points.Clear();
for (int k = 0; k < dstImage.Width; k++)
{
    //Draw top line
    for (int m = 0; m < 2; m++)
    {
        dstImage.SetPixel(k, m, Color.Black);
```

```
                if (img.GetPixel(k, m).Name == "ff000000")
                {
                    points.Add(new Point(k, m));
                }
            }
            //Draw average line
            int average = (y2 - y1) / 2;
            for (int m = average - 1; m < average + 1; m++)
            {
                dstImage.SetPixel(k, m, Color.Black);
                if (img.GetPixel(k, m).Name == "ff000000")
                {
                    points.Add(new Point(k, m));
                }
            }
            //Draw bottom line
            for (int m = dstImage.Height - 2; m < dstImage.Height; m++)
            {
                dstImage.SetPixel(k, m, Color.Black);
                if (img.GetPixel(k, m).Name == "ff000000")
                {
                    points.Add(new Point(k, m));
                }
            }
        }
        XPixels.Clear();
        YPixels.Clear();
        foreach (Point point in points)
        {
            XPixels.Add(point.X);
            YPixels.Add(point.Y);
        }
        if (!Directory.Exists(dirFormatted))
            Directory.CreateDirectory(dirFormatted);
        dstImage.Save(Path.Combine(dirFormatted,    Path.GetFileName(imgPath)),
ImageFormat.Gif);
        //MessageBox.Show("Done!");
    }

    public Bitmap ConvertToBinary(Bitmap srcImg, double threshold)
    {
        Bitmap dstImg = new Bitmap(srcImg.Width, srcImg.Height);
```

```
            for (int i = 0; i < srcImg.Width; i++)
            {
                for (int j = 0; j < srcImg.Height; j++)
                {
                    //double  brightness  =  GetPixelBrightness(srcImg.GetPixel(i,
j));
                    double brightness = srcImg.GetPixel(i, j).GetBrightness();
                    Color color = Color.FromArgb(0, 0, 0);
                    if (brightness > threshold)
                        color = Color.FromArgb(255, 255, 255);
                    dstImg.SetPixel(i, j, color);
                }
            }
            //dstImg.Save("C:\\test3.bmp", ImageFormat.Bmp);
            return dstImg;
        }
        private void FindWordsBySyllables()
        {
            int i = 0;
            wordsDictionary = new Dictionary<int, int>();
            foreach (string word in wordList)
            {
                int syllables = textHandler.GetWordSyllables(word);
                wordsDictionary.Add(i++, syllables);
            }
            List<string> wl = new List<string>();
            using (StreamWriter sw = new StreamWriter("c:\\111111.txt"))
            {
                for (i = 1; i < 20; i++)
                {
                    int counter = 0;
                    foreach (KeyValuePair<int, int> kvp in wordsDictionary)
                    {
                        if (kvp.Value == i)
                        {
                            if (!wl.Contains(wordList[kvp.Key]))
                            {
                                wl.Add(wordList[kvp.Key]);
                                sw.WriteLine(wordList[kvp.Key]);
                                if (counter++ > 100)
                                    break;
                            }
```

```
                    }
                }
            }
        }
    }

    private void CreateWordImages()
    {
        int i = 1;
        foreach (string word in wordList)
        {
            Draw(word.Trim());
            Save(i++.ToString() + ".gif");
        }
    }
    private void CreateWordFormArgs()
    {
        WordFormArgs args = new WordFormArgs();
        args.Args = new List<WordFormArg>();
        int i = 1;
        foreach (string word in wordList)
        {
            string fileName = i++.ToString() + ".gif";
            Draw(word.Trim());
            Save(fileName);
            WordFormArg arg = new WordFormArg();
            arg.Word = word;
            arg.Syllales = textHandler.GetWordSyllables(word);
            arg.ImgPath = Path.Combine(dirNonFormatted, fileName);
            List<int> XArray, YArray;
            Shape(arg.ImgPath, out XArray, out YArray);
            arg.XPositions = XArray.ToArray();
            arg.YPositions = YArray.ToArray();
            args.Args.Add(arg);
        }
        WordFormArgs.Serialize("c:\\WordFormArgs.xml", args);
    }

    #endregion
  }
}
```

Додаток В
(Обов'язковий)

ЗАТВЕРДЖУЮ
Завідувач кафедри КСУ
д.т.н., проф. В.М. Дубовой
_____
« _____» _____2019 р.

ПЕРЕЛІК
ГРАФІЧНИХ МАТЕРІАЛІВ
для захисту магістерської кваліфікаційної роботи
на тему

РОЗРОБКА МЕТОДІВ ТА ДОСЛІДЖЕННЯ МОДЕЛЕЙ ОБРОБКИ ТЕКСТОВИХ ДОКУМЕНТІВ В ІНТЕЛЕКТУАЛЬНИХ СИСТЕМАХ АВТОМАТИЗАЦІЇ

1. Математична модель процесу розпізнавання тексту.
2. Структура програмного забезпечення.
3. UML-діаграма класів.
4. Схема алогоритму підготовки корпусу тексту до тестування.
5. Схема алгоритму пошуку морфем української мови.
6. Схема алгоритму визначення статистичних ймовірностей появи морфем.
7. Схема алгоритму визначення ймовірностей переходів між морфемами корпусу тексту
8. Схема алгоритму визначення залежності довжини слова від кількості складів.
9. Схема алгоритму визначення статистичних ймовірностей появи літер, що мають надрядкові та підрядкові ознаки.
10. Схема алгоритму визначення місцезнаходження літери в слові, яка має надрядові та підрядкові ознаки.
11. Результати експериментальних досліджень. Головне вікно програми.
12. Результати експериментальних досліджень. Графічний інтерфейс програми "Морфесор 1.0".
13. Результати експериментальних досліджень. База даних появи літер, що мають надрядкові та підрядкові ознаки.
14. Результати експериментальних досліджень. База даних знаходження літери на початку, середині чи кінця слова, яка має надрядкові чи підрядкові ознаки

Розробив: Калінчук Р.С.
_____ _____
(підпис) (дата)
Перевірив: Биков М. М.
_____ _____
(підпис) (дата)
Рецензент: Кривогубченко С.Г.
_____ _____
(підпис) (дата)

МАТЕМАТИЧНА МОДЕЛЬ ПРОЦЕСУ РОЗПІЗНАВАННЯ ТЕКСТУ

Узагальнений функціонально-статистичний критерій: $\Э = \left.\dfrac{\Э_{\text{Р}}}{\Э_{\text{П}}}\right|_{\text{E} = \text{E}_{\partial}}$

де $\Э_{\text{Р}}$ і $\Э_{\text{П}}$ – функціонально-статистичні критерії для реальної і потенціальної систем розпізнавання відповідно: $\Э_{\text{Р}} = \dfrac{\text{I}_{\text{Р}}}{C_{\text{Р}}}$,

E_{∂} – задана в технічному завданні точність розпізнавання;

$\text{I}_{\text{Р}}, \text{I}_{\text{П}}$ – кількість інформації, яку дістає реальна і потенціальна системи відповідно;

$C_{\text{Р}}, C_{\text{П}}$ – вартість реальної і потенціальної систем відповідно, $C_{\text{Р}} = C_{\text{Х}} + C_{\text{К}}$,

де $C_{\text{Х}}$ – складність обчислення ознакового опису образів;

$C_{\text{К}}$ - складність обчислень класифікації образів.

Формальна постановка задачі побудови оптимальної стратегії розпізнавання тексту: $\tilde{S}_{Gopt} = \arg u \max \Э(\tilde{S}_{Gi})\Big|_{\{\tilde{S}_{Gi} \in \tilde{S}_G, W_d, r_d, E_d\}}$,

де $\tilde{S}_{Gi}$ – одна із стратегій розпізнавання із замкнутої відносно доступної інформації множини стратегій розпізнавання $\tilde{S}_G$;

W_d, r_d, E_d – задані умовою задачі розпізнавання алфавіт образів, рівень завад і точність розпізнавання відповідно.

Декомпозицію змінних C_k і C_x в критерії $\Э_{\text{Р}}$ можна здійснити шляхом зображення стратегії розпізнавання в вигляді покрокової процедури класифікації на дереві рішень.

Оптимальна стратегія є композицією алгоритмів розпізнавання, що максимізують відношення $\dfrac{I_i}{C_i}$ на кожному з рівнів. Послідовність композиції алгоритмів в оптимальній стратегії повинна відповідати послідовності розміщення рівнів дерева класифікації, а ознаки на кожному рівні повинні вибиратися з умови їх мінімальної складності

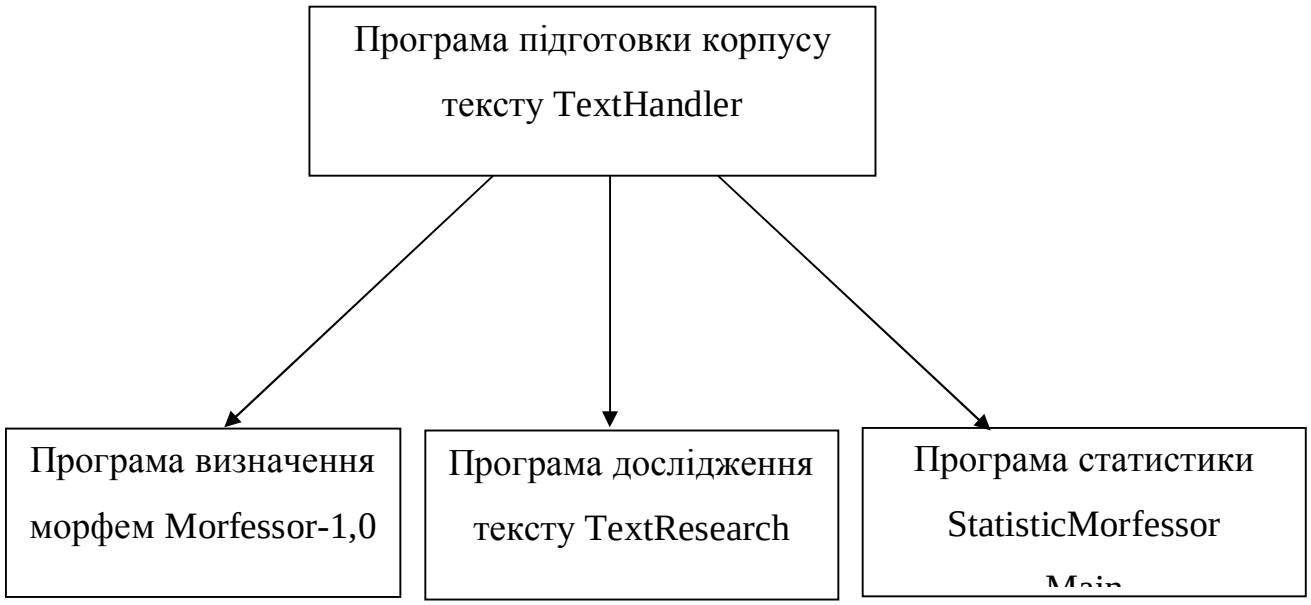
$$\tilde{S}_{Gopt} = A_1(\tilde{I}(S^1)) \otimes A_2(\tilde{I}(S^2)) \otimes \cdots \otimes A_W(\tilde{I}(W\ )),$$

$$\tilde{I}(S^i) = x^i_{opt}, x^i_{opt} = \arg u \min C_i(x^i_l)\ \ ,$$

де $\tilde{I}(W)$ – ознаковий опис еталонів образів; $C_i(x^i_l)$ – обчислювальна складність l-ї ознаки, що використовується для опису елементів i-го рівня; $A_h(\tilde{I}(S^h))$ – алгоритм попередньої класифікації образів тексту на групи, а $\tilde{I}(S^h))$- ознаковий опис образів S^h, що розпізнаються на даному рівні ієрархії.

Для сумісної оптимізації дерева рішень відносно помилок, швидкості і вартості необхідно на верхніх рівнях дерева використовувати більш інформативні ознаки.

СТРУКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



UML-ДІАГРАМА КЛАСІВ

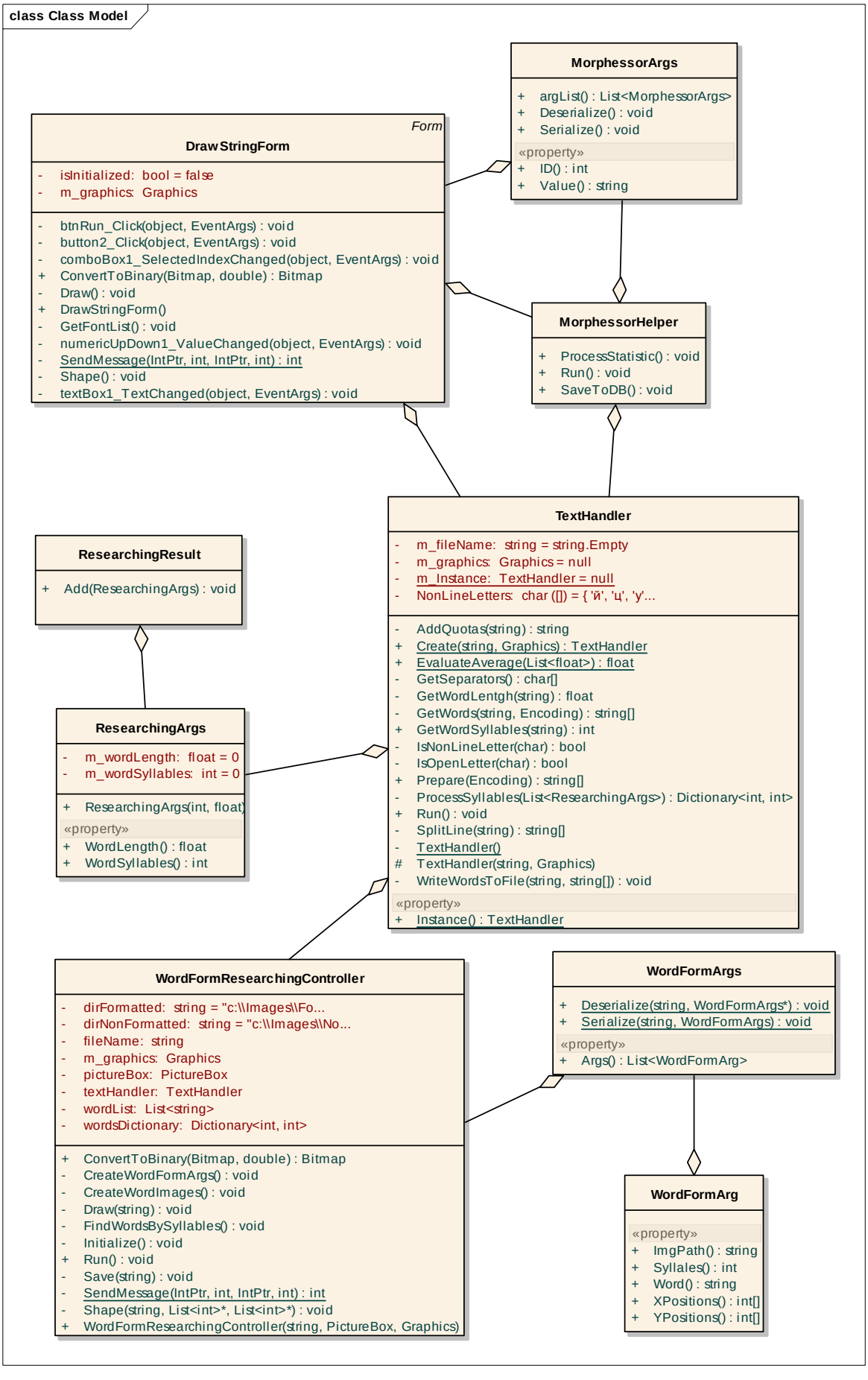


СХЕМА АЛГОРИТМУ ПІДГОТОВКИ КОРПУСУ ТЕКСТУ ДО ТЕСТУВАННЯ

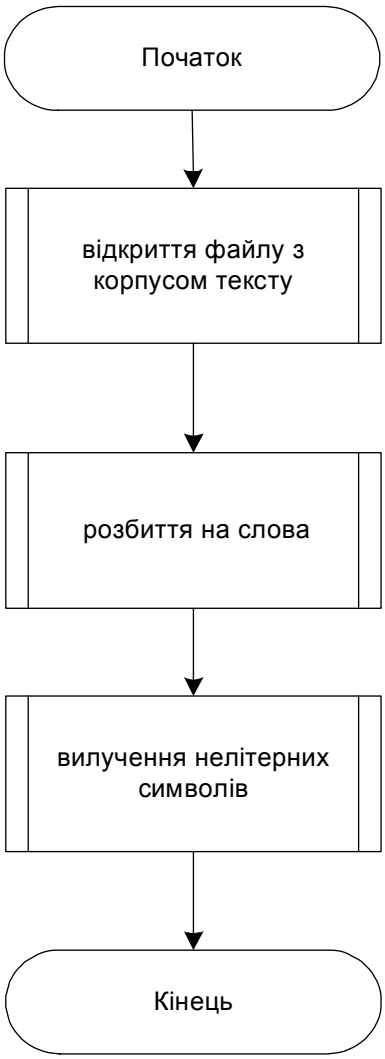


СХЕМА АЛГОРИТМУ ПОШУКУ МОРФЕМ УКРАЇНСЬКОЇ МОВИ

СХЕМА АЛГОРИТМУ ВИЗНАЧЕННЯ СТАТИСТИЧНИХ ЙМОВІРНОСТЕЙ ПОЯВИ МОРФЕМ

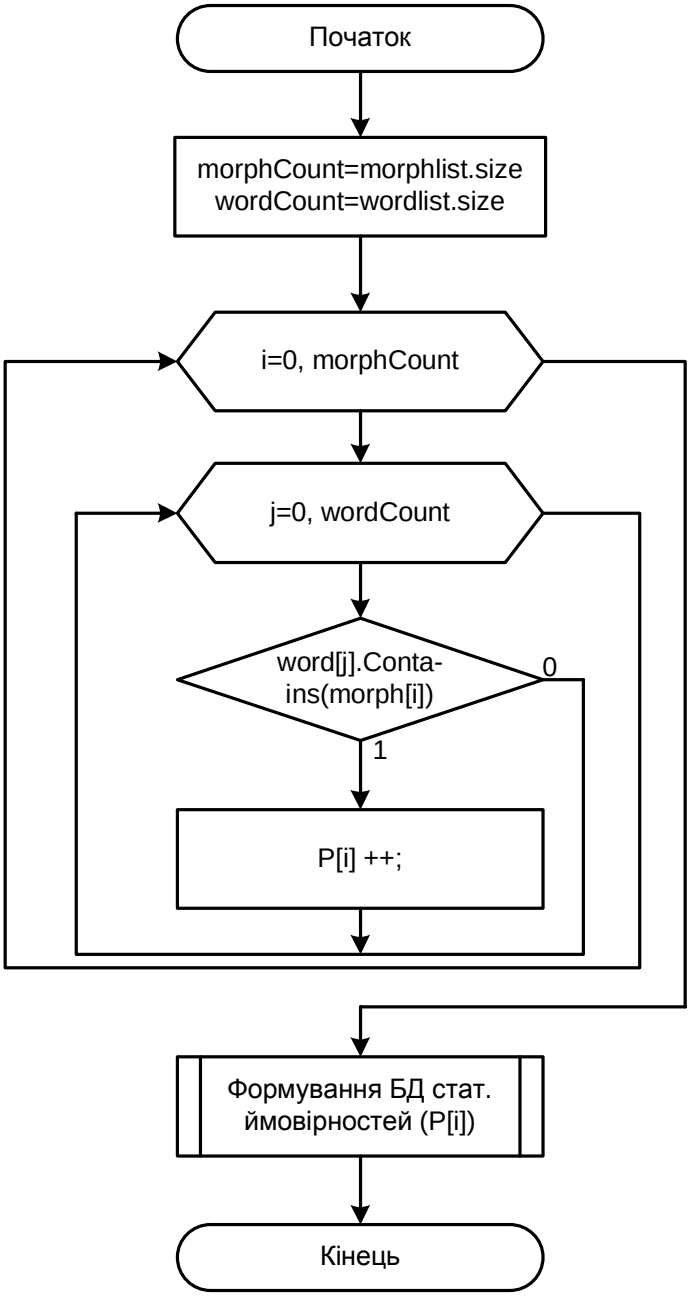


СХЕМА АЛГОРИТМУ ВИЗНАЧЕННЯ ЙМОВІРНОСТЕЙ ПЕРЕХОДІВ МІЖ МОРФЕМАМИ КОРПУСУ ТЕКСТУ

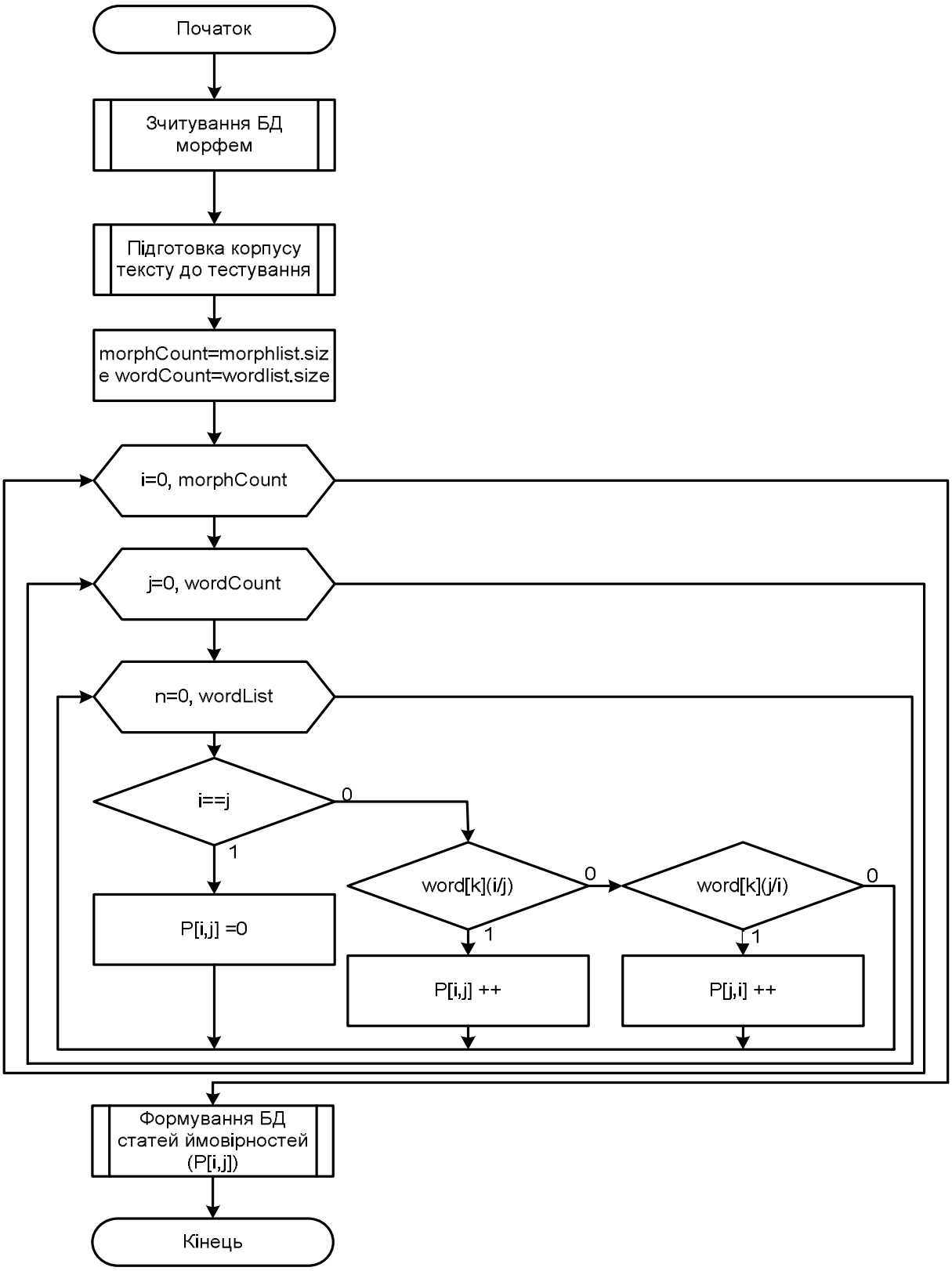


СХЕМА АЛГОРИТМУ ВИЗНАЧЕННЯ ЗАЛЕЖНОСТІ ДОВЖИНИ СЛОВА ВІД КІЛЬКОСТІ СКЛАДІВ

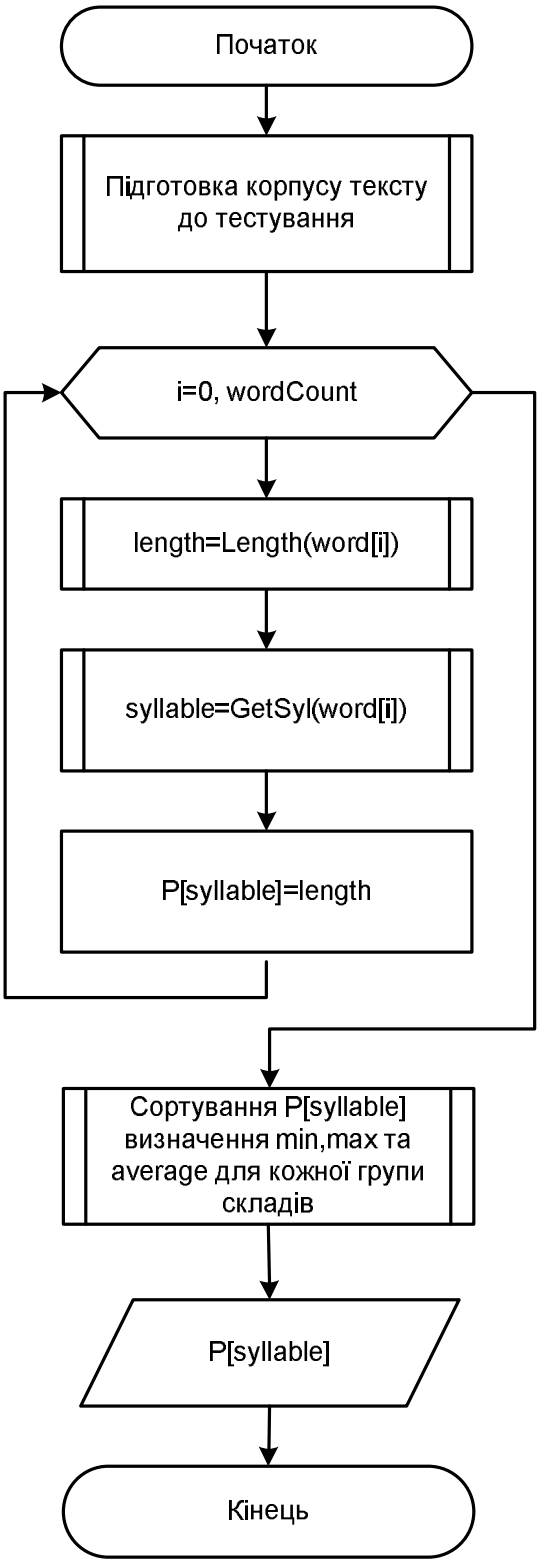


СХЕМА АЛГОРИТМУ ВИЗНАЧЕННЯ СТАТИСТИЧНИХ ЙМОВІРНОСТЕЙ ПОЯВИ ЛІТЕР, ЩО МАЮТЬ НАДРЯДКОВІ ТА ПІДРЯДКОВІ ОЗНАКИ

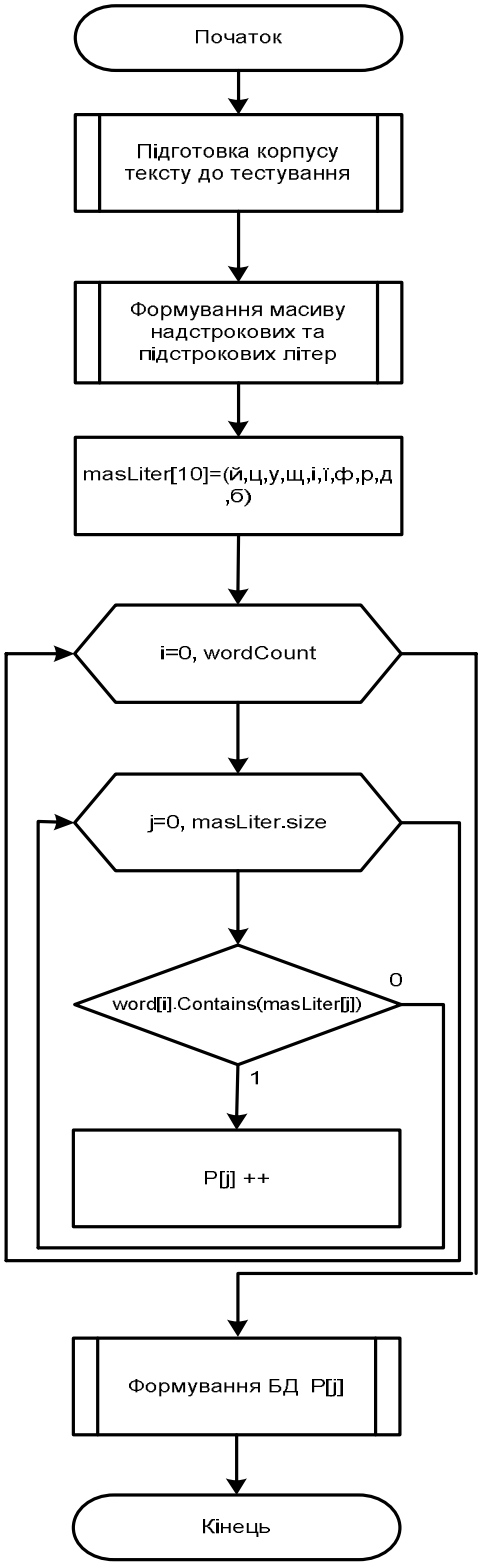
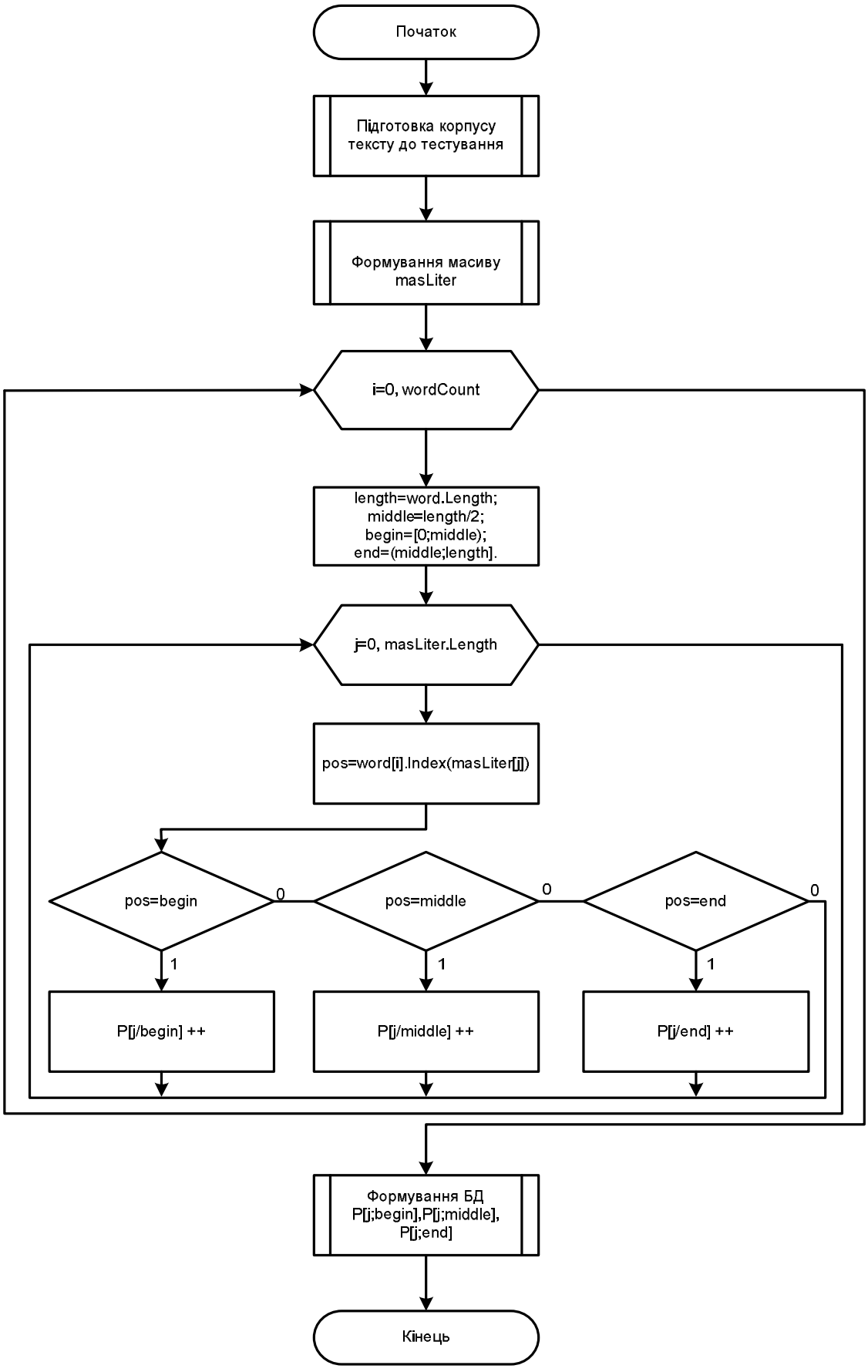
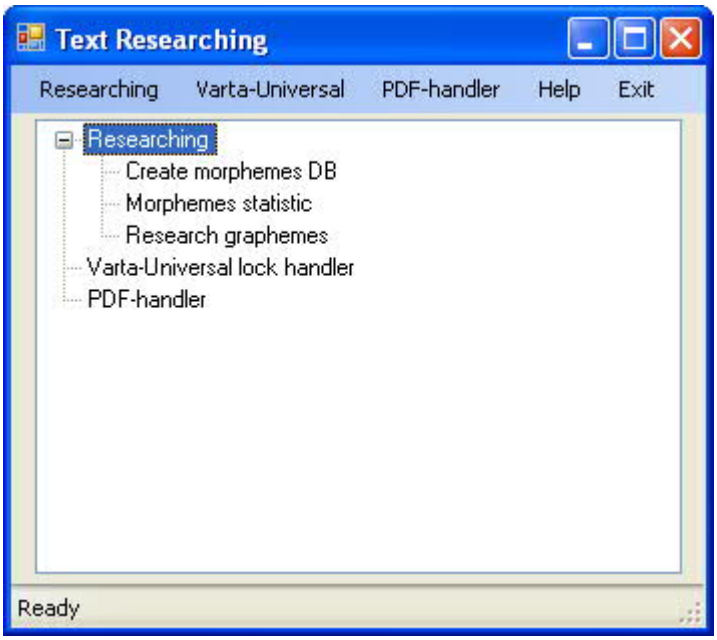


СХЕМА АЛГОРИТМУ ВИЗНАЧЕННЯ МІСЦЕЗНАХОДЖЕННЯ ЛІТЕРИ В СЛОВІ, ЯКА МАЄ НАДРЯДОВІ ТА ПІДРЯДКОВІ ОЗНАКИ



РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

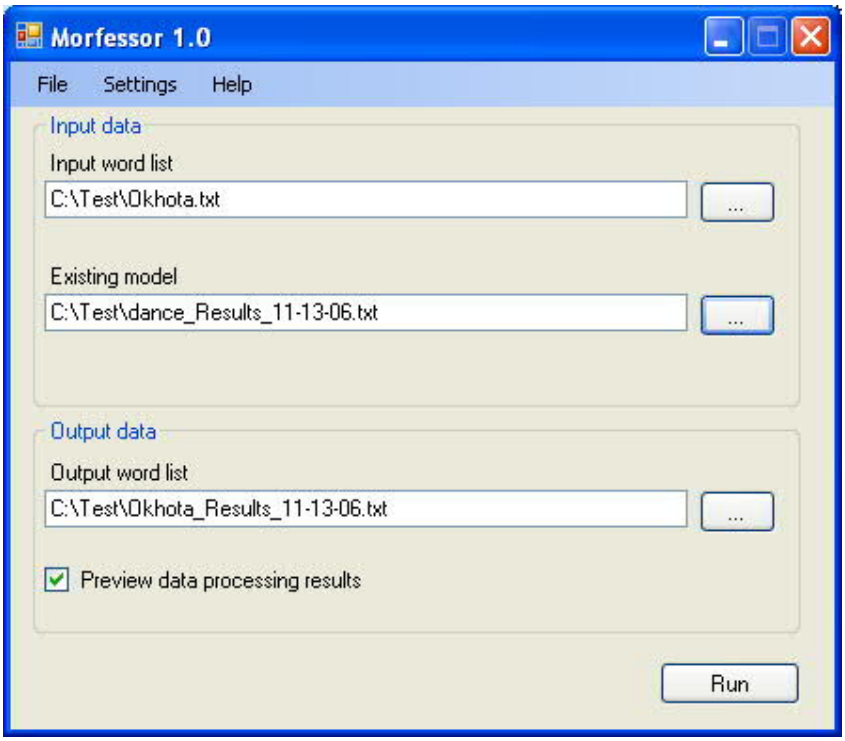
Головне вікно програми



Розроблене програмне забезпечення виконує такі основні функції:

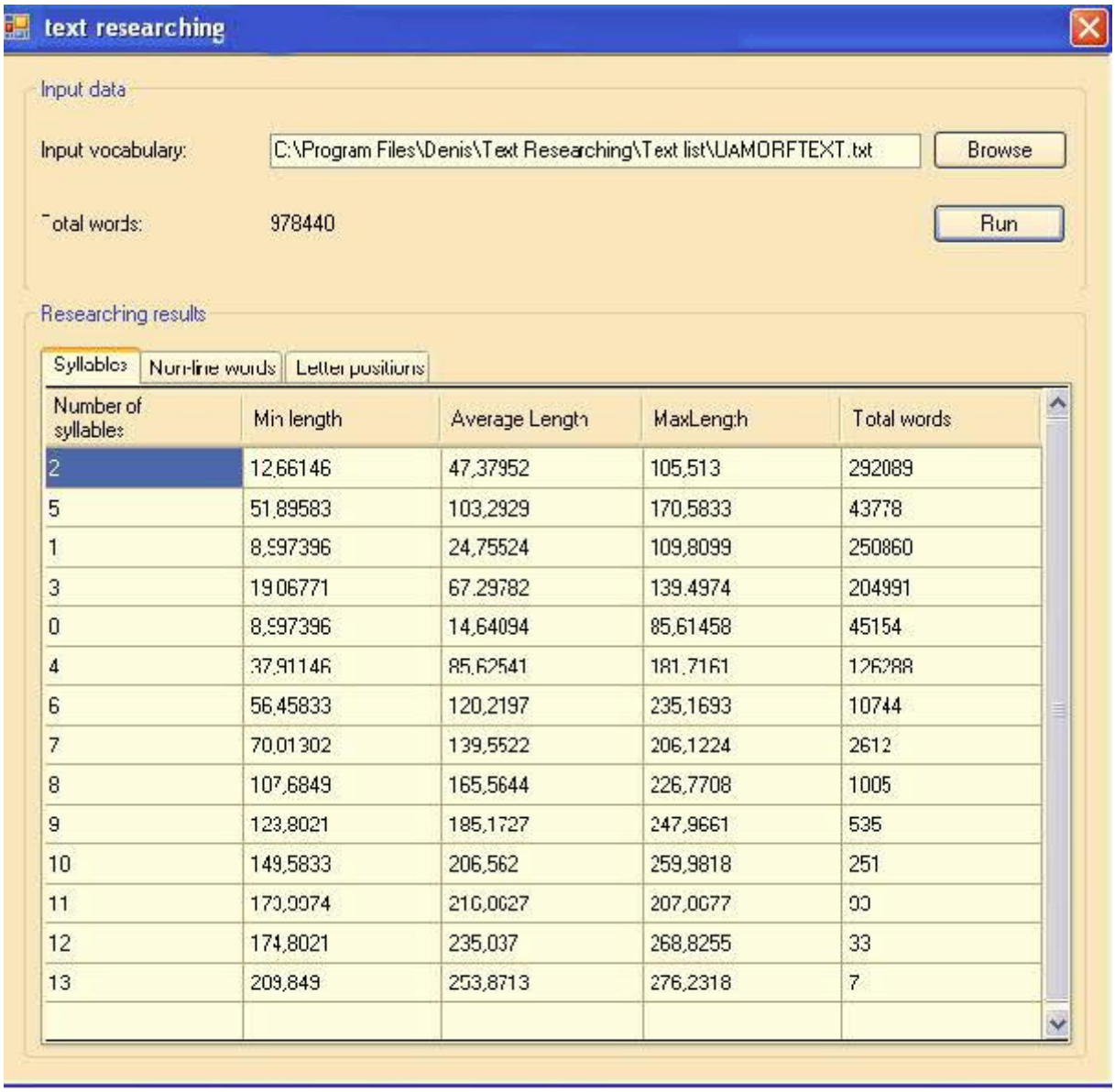
1) Дослідження текстових даних
    (a) Морфологічний аналіз морфем української мови, створення бази даних морфем української мови;
    (b) Створення бази даних статистичних та перехідних характеристик морфем української мови;
    (c) Оцінка інформативності графічник характеристик тексту
2) Опрацювання PDF-файлів.

Графічний інтерфейс програми “Морфесор 1.0”

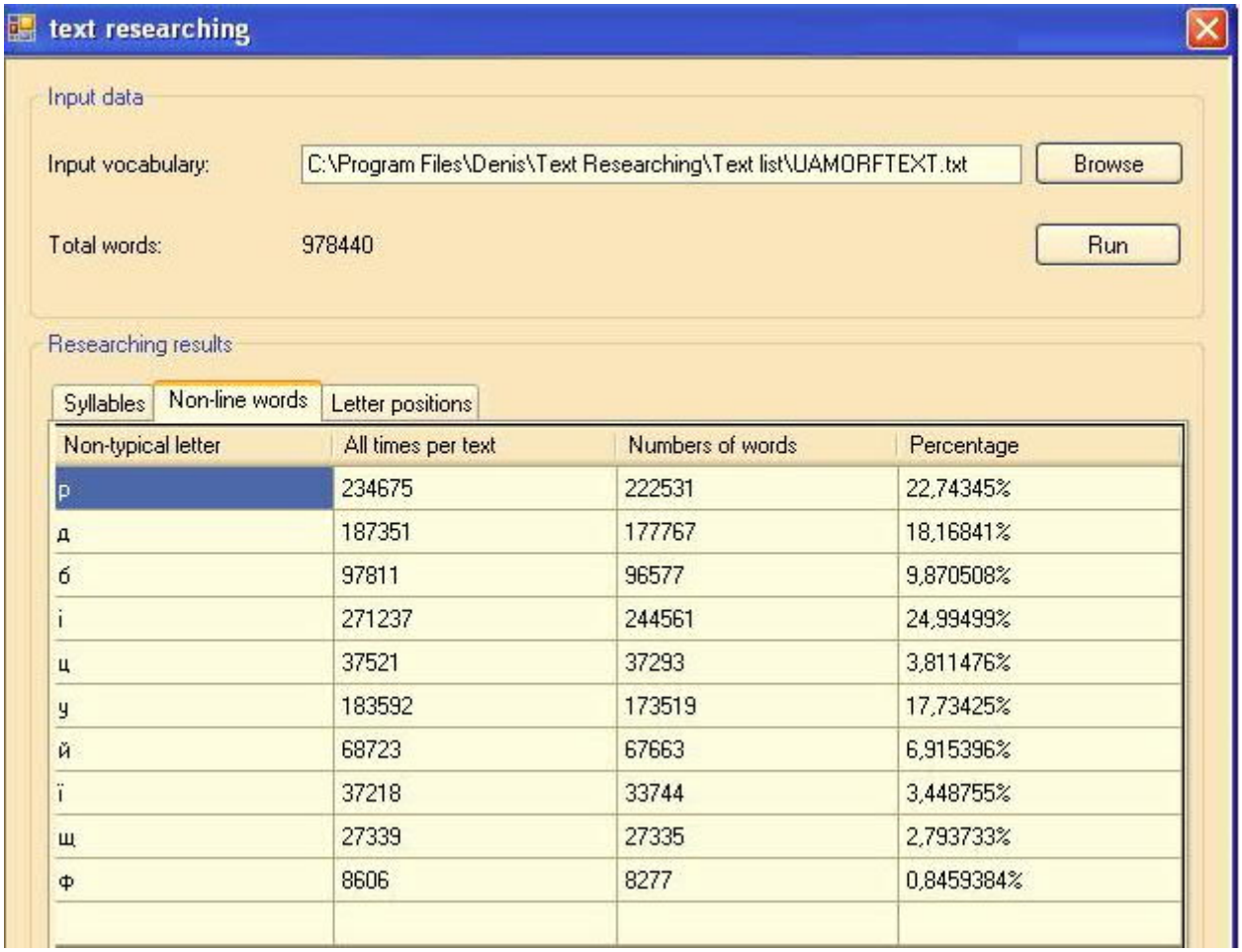


У головному вікні програми користувач має вказати ім’я текстового файлу, який необхідно сегментувати на морфеми і директорію для збереження та ім’я результуючого файлу. Програма автоматично призначає ту саму директорію для результуючого файлу, де знаходиться вхідний файл. Також програма генерує ім’я результуючого файлу, шляхом додавання до імені вхідного файлу службового слова «Results» та поточної дати. Як тільки модель морфемної сегментації навчена на деякому наборі даних, вона може використовуватися для сегментування нових словоформ. У цьому режимі сегментації програми “Морфесор” не відбувається навчання моделі. Кожне слово на вході сегментується на морфеми за допомогою алгоритму Вітербі, що знаходить самі схожі сегменти слова в послідовності морфем, що є в існуючій моделі. (для того, щоб забезпечити можливість існування як мінімум однієї сегментації, кожен окремий символ слова, що не існує, як морфема, може розглядатися як морфема з дуже малою імовірністю). Користувач може ввести ім’я файлу, що містить існуючу модель (друге вікно графічного інтерфейсу на рисунку).

База даних залежності кількості складів слова від його довжини



База даних появи літер, що мають надрядкові та підрядкові ознаки



Non-typical letter	All times per text	Numbers of words	Percentage
р	234675	222531	22,74345%
д	187351	177767	18,16841%
б	97811	96577	9,870508%
і	271237	244561	24,99499%
ц	37521	37293	3,811476%
у	183592	173519	17,73425%
й	68723	67663	6,915396%
ї	37218	33744	3,448755%
щ	27339	27335	2,793733%
ф	8606	8277	0,8459384%

База даних знаходження літери на початку, середині чи кінця слова, яка має надрядкові чи підрядкові ознаки