

Додаток А

Програмні коди

```

function [al, S] = hmm_alpha_prob(A_matr, p_vect, B_matr,
O_vect, Time)

    al = zeros(Time, length(A_matr(1, : )));

    al(1, : ) = p_vect' .* B_matr(:, O_vect(1))';

    S = zeros(Time, 1);
    S(1) = 1/sum(al(1, : ));
    al(1, : ) = al(1, : ) .* S(1);

    for t=1:Time-1
        al(t+1, : ) = (al(t, : ) * A_matr) .* B_matr(:,
O_vect(t+1))';

        S(t+1) = 1/sum(al(t+1, : ));
        al(t+1, : ) = al(t+1, : ) .* S(t+1);
    end
end
function [log_prob_t_j] = hmm_alpha_prob_log_approx(A_matr,
p_vect, B_matr, O_vect, Time)

    log_prob_t_j = zeros(Time, length(A_matr(1, : )));
    log_prob_t_j(1, : ) = log(p_vect') +
log(B_matr(:, O_vect(1))');

    for t=1:Time-1
        for j=1:length(A_matr(1, : ))
            log_prob_t_j(t+1, j ) = log(B_matr(j, O_vect(t+1))')
+ max(log(A_matr( : , j)') + log_prob_t_j(t, : ));
        end
    end

end
end function [ log_score ] = hmm_markov_assumption_scores(
A_matr, B_matr, O_vect, Time )

STATES = length(A_matr(1, : ));
SYMBOLS = length(B_matr(1, : ));

% Вероятность нахождения в j-м состоянии найдем из СЛАУ:
A = zeros(STATES + 1, STATES);
A(STATES + 1, : ) = ones(1, STATES);
A(1:STATES, : ) = A_matr( : , : )';
for i=1:STATES
    A(i, i) = A(i, i) - 1;
end
end

```

```

B = zeros(STATES + 1, 1);
B(STATES+1) = 1; P_assumption =
linsolve(A, B);

B_assumption = zeros(SYMBOLS, 1);
for k = 1:SYMBOLS
    B_assumption(k) = P_assumption' * B_matr( : , k, 1);
end

log_score = 0;
for t=1:Time
    log_score = log_score + log(B_assumption(O_vect(t)));
end

end
function [ A_spikes, B_spikes, p_spikes ] =
aom_create_spike_flows( A_matr, B_matr, p_vect,
spikes_flow_length )
%Генерация последовательностей спайков - носителей вероятностей A,
B, p
%-----
-----
    % ИНИЦИАЛИЗАЦИЯ
    % Генерация последовательностей спайков для моделирования
A_i_j, p_i и B_j_k:
    STATES = length(A_matr( : , 1));
    CODEBOOK_SIZE = length(B_matr(1, : ));
    A_spikes = false(STATES, STATES, spikes_flow_length)
    B_spikes = false(STATES, CODEBOOK_SIZE, spikes_flow_length);
    p_spikes = false(STATES, spikes_flow_length);

    for i=1:STATES
        if isnan(p_vect(i)) || isinf(p_vect(i))
            % maximal intensity for Inf:
            p_spikes(i, : ) = logical(binornd(1, 1, [1
spikes_flow_length]));
        else
            p_spikes(i, : ) = logical(binornd(1, p_vect(i), [1
spikes_flow_length]));
        end
        for j=1:STATES
            if isnan(A_matr(i, j)) || isinf(A_matr(i, j))

                A_spikes(i, j, : ) = logical(binornd(1, 1, [1
spikes_flow_length]));
            else
                A_spikes(i, j, : ) = logical(binornd(1,
A_matr(i, j), [1 spikes_flow_length]));
            end
        end
        for k=1:CODEBOOK_SIZE
            B_spikes(i, k, : ) = logical(binornd(1, B_matr(i,
k), [1 spikes_flow_length]));
        end
    end
end

```

```

        end

end function [ B_assumption_spikes, B_assumption_log_spikes ]
= aom_create_markov_assumption_vectors( A_spikes, B_spikes,
A_i_j, B_j_k, spikes_flow_length ) STATES = length(A_spikes(
: , 1, 1));
SYMBOLS = length(B_spikes(1, : , 1));

% Вероятность нахождения в j-м состоянии найдем из СЛАУ:
A = zeros(STATES + 1, STATES);

for i=1:STATES
    for j=1:STATES
        A(i, j ) = A_i_j( i , j);
    end
end

A(1:STATES, : ) = A(1:STATES, : )';
A(STATES + 1, : ) = ones(1, STATES);

for i=1:STATES
    A(i, i) = A(i, i) - 1;
end
B = zeros(STATES + 1, 1);
B(STATES+1) = 1;

P_assumption = linsolve(A, B);

B_assumption = zeros(SYMBOLS, 1);
for k = 1:SYMBOLS
    B_assumption(k) = P_assumption' * B_j_k( : , k);
end
B_assumption_spikes = false(1, SYMBOLS, spikes_flow_length);
B_assumption_log_spikes = zeros(1, SYMBOLS, spikes_flow_length);
correct = 10;
for k=1:SYMBOLS
    B_assumption_spikes(1, k, : ) = logical(binornd(1,
B_assumption(k), [1 spikes_flow_length]));
    B_assumption_log_spikes(1, k, : ) = logical(binornd(1,
correct*abs(log(B_assumption(k))/spikes_flow_length), [1
spikes_flow_length]));
end

end

function [ score_intensity ] = aom_markov_assumption_scores_log( B_assumption_log,
O_vect, Time, spikes_flow_length )

score_flow = false(1, 1, spikes_flow_length);
correct = 10;
for t=1:Time
    score_flow(1, 1, : ) = score_flow(1, 1, : ) |

```

```

(B_assumption_log(1, O_vect(t), : ));
end

score_intensity = (-
spikes_flow_length/correct)*aom_intensity(score_flow(1, 1, : ));

end
library IEEE;
use ieee.std_logic_1164.all;

entity aom_adder is
  generic (
    spikes_flow_length: natural:=256
  );
  port (
    CLK, CNT, RST, SRD: in std_logic;
    DI: in std_logic_vector(spikes_flow_length-1 downto 0);
    serial_out: out std_logic
  );
end aom_adder;

architecture behavior of aom_adder is
begin
  process(CLK)
    - кл. ансамбли "сумматор" ариф. блока:
    variable sum_buff: std_logic_vector(spikes_flow_length-1 downto 0);
  begin
    if rising_edge(CLK) then
      if (RST='1') then
        sum_buff := (others=>'0');
        serial_out <= '0';
      elsif (SRD='1') then
        - Чтение старшего разряда выходного регистра:
        serial_out<=sum_buff(spikes_flow_length-1);
        - Сдвиг вправо выходного регистра:
        for i in spikes_flow_length-1 downto 1 loop
          sum_buff(i):= sum_buff(i-1);
        end loop;
      elsif (CNT='1') then
        sum_buff := DI or sum_buff;
      end if;
    end if;
  end process;
end behavior;
library IEEE;
use ieee.std_logic_1164.all;
entity aom_differential is
  generic (charge_size: natural := 4);
  port (
    Q, S, CLK, EN, RST: in std_logic;
    out_q, out_s: out std_logic
  );
end aom_differential;

```

```

architecture behavior of aom_differential is
shared variable queue_q, queue_s: std_logic_vector(charge_size-1 downto 0);
begin

q_proc: process(CLK)
variable q_buf: std_logic;
begin

```

```

    if rising_edge(CLK) then
        if (EN='1') then
            if (RST='1') then
                queue_q:=(others=>'0');
                out_q <= queue_q(0);
            else
                queue_q(charge_size-1) or
q_buf := (Q and (not queue_s(charge_size-1))) or
            ((not Q) and (not queue_s(charge_size-1)) and
            (Q and queue_q(charge_size-1) and
            queue_s(charge_size-1));

            - сдвиг замкнутого осциллятора: for
            i in charge_size-1 downto 1 loop
                queue_q(i) := queue_q(i-1);
            end loop;
            queue_q(0):=q_buf; out_q
            <= queue_q(0);
        end if;
    end if;
end process q_proc;

s_proc: process(CLK)
variable s_buf: std_logic;
begin
    if rising_edge(CLK) then
        if (EN='1') then
            if (RST='1') then
                queue_s:=(others=>'0');
                out_s <= queue_s(0);
            else
                queue_s(charge_size-1) or

```

```

s_buf := (S and (not queue_q(charge_size-1))) or
         ((not S) and (not queue_q(charge_size-1)) and
          (S and queue_s(charge_size-1) and
           queue_q(charge_size-1)));

```

```

- сдвиг замкнутого осциллятора: for
  i in charge_size-1 downto 1 loop
    queue_s(i) := queue_s(i-1);
  end loop;
  queue_s(0) := s_buf;
  out_s <= queue_s(0);
end if;
end if;
end if;
end process s_proc;
end behavior;

```

Додаток Б
Слайди презентації

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
РАДІОЕЛЕКТРОНІКИ
Кафедра ПІ

Атестаційна робота
магістра

**Дослідження методів розпізнавання
мови в асоціативних середовищах**

Виконав
ст.гр. ПЗСзм-18-1
Боричев С.О.

Науковий керівник
проф. Шубін І.Ю.

1 1

Мета роботи

- Прогрес обчислювальної техніки в цілому, дозволив звернутися до рішення однієї із проблем штучного інтелекту – автоматичному розпізнаванню мови.
- **Мета** роботи полягає в розробці методів розпізнавання мови в асоціативних середовищах і побудові системи розпізнавання в цих середовищах
- **Об'єктом** дослідження є методи розпізнавання мови
- **Предметом** дослідження є методи й алгоритми розпізнавання мови й шляхи їх реалізації в асоціативних середовищах.

1 2

Постановка задачі

Для досягнення цієї мети вирішуються наступні завдання:

- вибір методу виділення й попередньої обробки мови, витягу ознак;
- алгоритмічна реалізація виділення мови і її попередньої обробки;
- вибір методу розпізнавання мови;
- вибір асоціативного середовища для реалізації в ній розпізнавання;
- створення програмної моделі розроблених методів розпізнавання мови в середовищі.

3

Класифікація векторів ознак

Наведена класифікація й опис різних підходів складання векторів ознак, що працюють як у частотній області (коефіцієнти лінійного передбачення, перцепційні коефіцієнти лінійного передбачення, так і в часовій (частота проходів через нуль, короткочасна енергія сигналу).

Серед проаналізованих методів знаходження векторів ознак для використання в роботі був обраний метод мел-кепстральних коефіцієнтів;

- наведена класифікація методів розпізнавання мови й опис найпоширеніших з них – динамічне вирівнювання часу, побудова штучних нейронних мереж і апарат прихованих марківських моделей.
- На основі аналізу цих методів були обрані приховані марківські моделі;
- Розглянуто різні типи асоціативних середовищ в історичному порядку їх появи. Аналіз досягнень у цій області дозволив вибрати асоціативне осциляторне середовище як найбільш підходящу для використання.

4

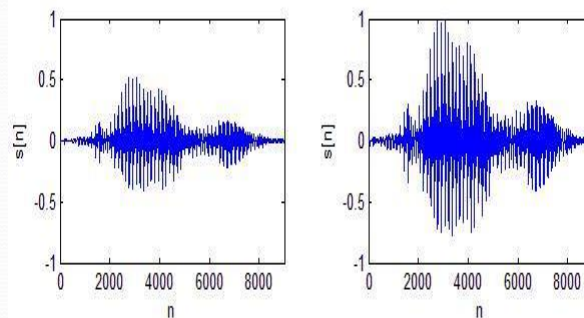
Клітинні ансамблі

- Клітинні ансамблі можна описувати подвійно: по-перше, з погляду залежності значення на виході в наступному такті від значень на вході в поточному такті; по-друге, через залежність інтенсивності потоку спайків на виході від інтенсивностей потоків спайків на входах.
- Найпростішим клітинним ансамблем є провідник, який ніяк не змінює вхідну інтенсивність потоку спайків.
- Для реалізації алгоритмів розпізнавання мови було обране асоціативне осциляторне середовище, тому що:
 - у ньому можлива організація поточкових і конвеєрних обчислень;
 - гнізда середовища можна гнучко з'єднувати один з одним, не обмежуючись матричною структурою;
 - за один такт кожне гніздо обробляє інформацію (вхідні спайки) відповідно закладеному в нього закону функціонування.
 - поставлене завдання розпізнавання мови й описана загальна структура системи автоматичного розпізнавання мови;

5

Нормалізація мовного сигналу

- Для усунення розкиду гучності застосовується процедура нормалізації мовного сигналу по амплітуді



Оцифрований мовний сигнал до (а) і після (б) нормалізації

6

Методи виділення мови із сигналу

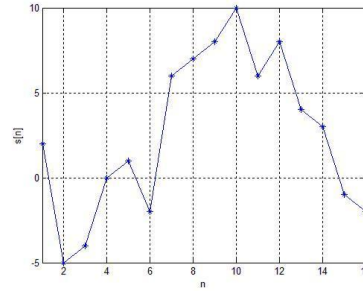
- Існуючі методи працюють у часовій області.
- До них відносяться знаходження короточасної енергії сигналу, короточасної потужності сигналу й частоти проходів через нуль

$$E_s(m) = \sum_{n=m-L+1}^m s^2(n)$$

$$P_s(m) = \frac{1}{L} \sum_{n=m-L+1}^m s^2(n)$$

$$Z_s(m) = \frac{1}{L} \sum_{n=m-L+1}^m \frac{|sgn(s(n)) - sgn(s(n-1))|}{2}$$

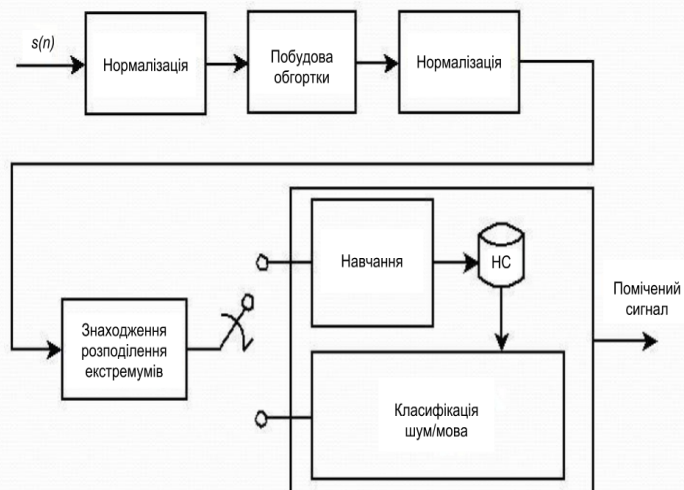
$$sgn(s(n)) = \begin{cases} +1, & s(n) \geq 0 \\ -1, & s(n) < 0 \end{cases}$$



Приклад ділянки дискретного мовного сигналу із шістьма екстремумами

7

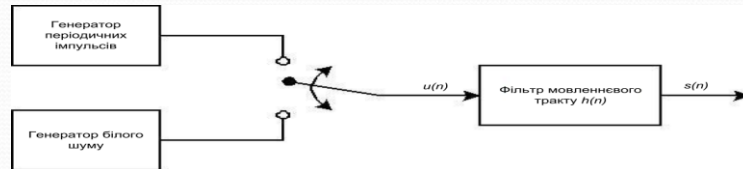
Підсистема виділення мовного сигналу



8

Алгоритми виділення ознак мовного сигналу

- Склад ознак визначається моделлю утвору мови, що включає в себе сигнал порушення і фільтр мовного тракту, імпульсна характеристика якого змінюється в часі .
- Вихідний сигнал $s(n)$, що представляє собою згортку $u(n)$ і $h(n)$, і є мова

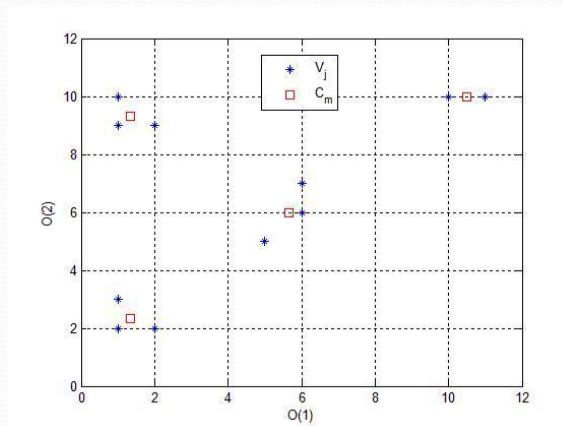


Модель утвору мови "порушений фільтр"

При проголошенні голосних звуків сигнал порушення приймає квазіперіодичну форму, а у випадку глухих звуків – форму білого шуму.
При цьому конкретну фонему визначає конфігурація мовного тракту, а не форма сигналу порушення.

9

Алгоритм векторного квантування



Результат кластеризації для дво-вимірних векторів

10

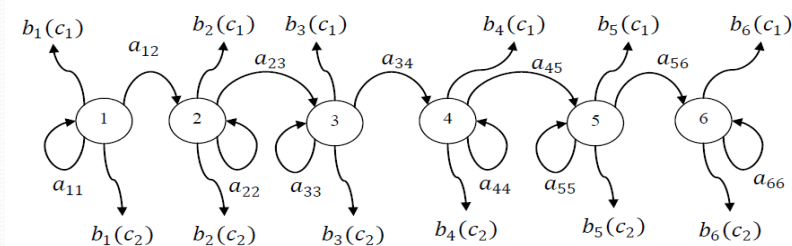
Процедура векторного квантування

- Для ефективного стиснення мовного сигналу, а також спрощення розпізнавача, запропоновано застосування процедури векторного квантування (Vector Quantization – VQ):
 - на етапі навчання (кластеризації) складається словник з $M=zb$ p -мірних векторів (кодових слів-еталонів),
 - на етапі класифікації розглянутий вектор заміняється індексом найбільш близького до нього кодового слова
- У такий спосіб досягаються наступні цілі:
 - стиснення мовного сигналу – замість послідовності p -мірних векторів одержують послідовність тієї ж довжини, але з b -бітних чисел;
 - перехід від безперервного вектора ознак до дискретного – кожний із коефіцієнтів є безперервною величиною, тоді як індекс еталона – ціле число в діапазоні $[1, zb]$, що дозволяє побудувати більш простий розпізнавач, наприклад, дискретні приховані марківські моделі.

11

Приклад ПММ

- Можна виділити наступні підходи до складання ПММ (вони можуть як бути взаємовиключними, так і взаємодоповнюючими):
- кількість станів ПММ відповідає числу фонем у слові, що моделюється або середньому числу спостережень у реалізації моделюемого слова;
- фонемі моделюються за допомогою трьох станів – початкового, середнього й кінцевого.

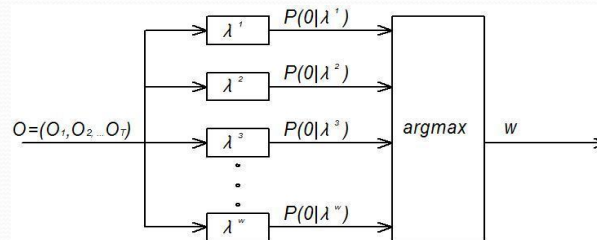


ПММ для слова «так», що містить фонемі /d/-/a/, кожна з яких включає три стани: початкове, середнє й кінцеве

12

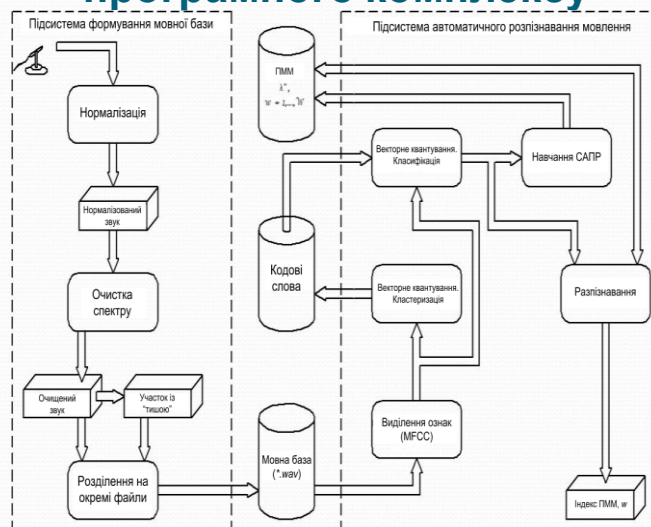
Загальна структура розпізнавача на ПММ

- На етапі навчання для кожного слова зі словника складається прихована Марківська модель λ^W , де W – кількість слів у словнику системи (класів розпізнавання).
- Блок розпізнавання побудований за наступним принципом: на етапі розпізнавання вибирається та модель, для якої ймовірність породити розглянуту послідовність ознак максимальна.
- Для знаходження ймовірності використовується алгоритм прямого ходу.



13

Структурна схема розробленого програмного комплексу



14

Алгоритм має наступні параметри

- мінімальна тривалість проголошення слова;
- мінімальна тривалість пауз між словами;
- поріг гучності ділянок з паузою між словами.

Для проведення досліджень у підсистемі розпізнавання були реалізовані:

- вихідний алгоритм прямого ходу - для вирішення проблеми експоненційного спадання використаний метод введення коефіцієнтів;
- модифікація алгоритму прямого ходу, при цьому розроблено два варіанти: **перший**, де усі обчислення виконуються без використання осциляторного середовища, і **другий**, що представляє собою програмну модель АОС

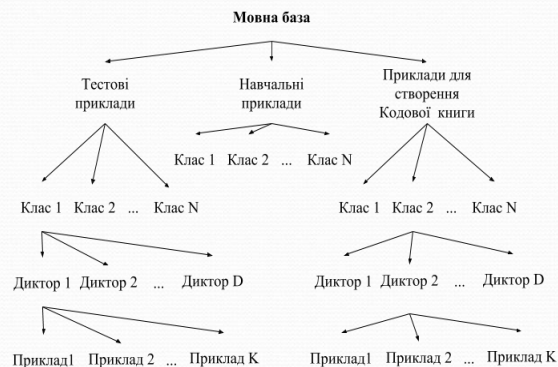
Запропонований алгоритм розпізнавання не враховує інформацію про порядок проходження звуків у слові.

Для нього також існують два варіанти: без використання осциляторного середовища і повністю на програмній моделі середовища - програмних моделях осциляторного середовища (потоки спайків представляються за допомогою булевих векторів)

15

Структура експериментальної мовної бази

- Експеримент мав наступні кроки:
- формування ПММ для кожного класу – етап навчання;
- розпізнавання на тестовій вибірці обраним методом розпізнавання.
- Використання апарата прихованих Марківських моделей припускає підбір великої кількості параметрів розпізнавання: кількість станів ПММ, початкові значення матриць і типу ПММ (ліво-права, повнозв'язна і т.д.) і багатьох інших.



16

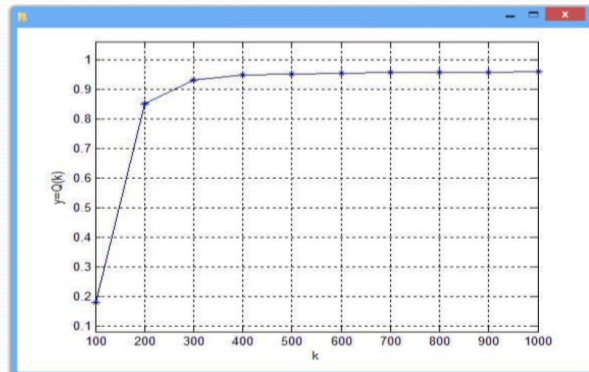


17



18

Залежність точності розпізнавання від довжини потоку спайків



19

Висновки

- В роботі досліджено нові методи розпізнавання мови в асоціативних середовищах.
- Проведений аналіз завдання зберігання та розпізнавання мови, виділені основні компоненти систем автоматичного розпізнавання мови.
- Розглянуто методи попередньої обробки та виділення ознак мовного сигналу, серед яких обраний підхід, заснований на знаходженні мел-кепстральних коефіцієнтів.
- Розроблений програмний комплекс та експериментальна перевірка запропонованих методів розпізнавання мови в асоціативному осциляторному середовищі :
 - програмний комплекс включає засоби для формування експериментальної мовної бази, а також навчання й тестування CAPM.
- Описані завдання, що виникають при складанні мовної бази, а також її основні характеристики.

20

Додаток В
Відгук і рецензії

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ
Науково-навчальний центр заочної форми навчання

ВІГУК
на атестаційну роботу магістра

Боричева Сергія Олександровича, гр. ПЗСзм 18-1

спеціальність 121 – *Інженерія програмного забезпечення*
освітньо-професійна програма - «*ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМ*»
Тема атестаційної роботи «Дослідження методів розпізнавання мови в асоціативних середовищах»

Елементами наукової новизни та здійснення інновацій в умовах невизначеності вимог і ступінь складності розробки відповідає усім нормативним вимогам до магістерських робіт освітньо-професійної програми «програмне забезпечення систем», а саме створення систем розпізнавання мови залишається надзвичайно складною проблемою. Це обумовлюється як її міждисциплінарним характером (необхідно мати знання у філології, лінгвістиці, цифровій обробці сигналів, акустиці, статистиці, розпізнаванні образів і т.п.), так і високою обчислювальною складністю розроблених алгоритмів.

В результаті роботи програма, якої відповідала більш висока надійність, мала більш просту структуру, і цей факт підтверджує гіпотезу, що встановилася, що ускладнення структури програм веде до зниження їх надійності.

Остаточний висновок щодо оригінальності роботи, враховуючи критерії оцінювання плагіату (п.4 Положення про протидію академічному плагіату в ХНУРЕ) – згідно звіту системи «Unicheck» ID перевірки:1000752356 – робота є повністю оригінальною.

Магістрант гр. ПЗСзм-18-1 Боричев Сергій Олександрович готовий до самостійної інженерної діяльності. Атестаційну роботу можна подати до захисту в ЕК за спеціальністю 121 – «Інженерія програмного забезпечення», освітньо-професійною програмою *Програмне забезпечення систем*.

Керівник атестаційної роботи магістра

І.Ю. Шубін, професор кафедри ПІ





Власник документу:
Нечволод Вадим Юрійович каф. ПІ

ID перевірки:
1000752356

Дата перевірки:
10.12.2019 11:31:55 GMT+0

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
10.12.2019 11:32:38 GMT+0

ID користувача:
94949

Назва документу: 2019_ПІ_ПЗСзм-18-1_Боричев С.О. _скорочений

ID файлу: 1000764354 Кількість сторінок: 33 Кількість слів: 6030 Кількість символів: 45979 Розмір файлу: 809.45 KB

3.32% Схожість

Найбільша схожість: 3.32% з джерело бібліотеки. ID файлу: 1000028244

0.13% Схожість з Інтернет джерелами 2 Page 35

3.32% Текстові збіги по Бібліотеці акаунту 1 Page 35

0% Цитат

Не знайдено жодних цитат

0% Вилучень

Вилучений текст відсутній

Підміна символів

Заміна символів 8

Рецензія

на атестаційну роботу магістра
магістранта групи ПЗСзм-18-1 *Боричева Сергія Олександровича*
спеціальність – 121- **Інженерія програмного забезпечення**
Освітньо-професійна програма **Програмне забезпечення систем**

Тема атестаційної роботи: «Дослідження методів розпізнавання мови в асоціативних середовищах»

Структура атестаційної роботи: пояснювальна записка _____ 87 стор.; графічна частина 17 аркушів; програмне застосування (прикладна програма) 12 файлів загальним обсягом 1200 Кбайт.

Широкий спектр застосування інтелектуальних методів обробки природної мови поспонує необхідність створення компактного, надійного, самостійного й максимально швидкодіючого обладнання. Над рішенням позначеного завдання працює безліч фахівців. Можна виділити наступні напрямки досліджень і розробок в області підвищення швидкодії й реалізації самостійних модулів розпізнавання мови. Для досягнення цієї мети вирішуються наступні завдання: вибір методу виділення й попередньої обробки мови, витягу ознак; програмна реалізація виділення мови і її попередньої обробки; вибір методу розпізнавання мови; вибір асоціативного середовища для реалізації в ній розпізнавання; розробка блоку розпізнавання на елементах асоціативного середовища; створення мовної бази для навчання й тестування системи; створення програмної моделі розроблених методів розпізнавання мови в середовищі.

Зміст роботи відповідає завданню та темі атестаційної роботи. Атестаційна робота виконана у відповідності до завдання та чинних вимог з дотриманням внутрішньої логіки, самостійно та в повному обсязі.

В атестаційній роботі магістрант Боричев С.О. детально проаналізував існуючі можливості сучасних алгоритмів та методів, ознайомився з відомими науковими роботами та привів аргументовані думки із застосуванням ілюстрацій прикладів та формул при побудові власної математичної моделі для покращення обчислень та оптимізації у розрахунках границь зі складною формою.

До недоліків можна віднести те, під час роботи програмного додатку є необхідність налаштування параметрів алгоритмів, зазначений недолік не впливає на загальну позитивну оцінку магістерської роботи.

Атестаційна робота магістра гр. ПЗСзм-18-1 Боричева Сергія Олександровича відповідає вимогам до атестаційних робіт магістрів і заслуговує оцінки «добре (85)».

Атестаційну роботу можна представити для захисту в ЕК за спеціальністю *121-Інженерія програмного забезпечення*, освітньо-професійною програмою *Програмне забезпечення систем*

Рецензент:

к.т.н., професор,
завідувач кафедри ПІ



З.В. Дулар

Рецензія

на атестаційну роботу магістра
магістранта групи ПЗСзм-18-1 *Боричева Сергія Олександровича*
спеціальність – 121- **Інженерія програмного забезпечення**
Освітньо-професійна програма **Програмне забезпечення систем**

Тема атестаційної роботи: «Дослідження методів розпізнавання мови в асоціативних середовищах»

Структура атестаційної роботи: пояснювальна записка 87 стор.;
графічна частина 17 аркушів; програмне застосування (прикладна програма) 12 файлів загальним обсягом 1200 Кбайт.

Пошук нових архітектурних рішень, що не базуються на архітектурі фон Неймана, є актуальною темою, особливо в її додатку до рішення завдань штучного інтелекту, до яких ставиться й розпізнавання мови. Описані алгоритми та методи розроблені на основі сучасних теорем та припущень, особливу увагу приділено оптимізації обробки та обчислень.

Зміст роботи відповідає завданню та темі атестаційної роботи. Атестаційна робота виконана у відповідності до завдання та чинних вимог з дотриманням внутрішньої логіки, самостійно та в повному обсязі.

В атестаційній роботі студент Боричев С. О. детально проаналізував існуючі можливості сучасних алгоритмів та методів, ознайомився з відомими науковими роботами та привів аргументовані думки із застосуванням ілюстрацій прикладів та формул при побудові власної математичної моделі для покращення обчислень та оптимізації у розрахунках границь зі складною формою.

Магістрант показав здатність формулювати власну думку, вміння чітко та зрозуміло проводити аналіз теоретичних матеріалів та на їх основі впроваджувати власні моделі та алгоритми. Описані розрахунки мають вагомий теоретичний і практичний результати, що свідчить про достатньо високий науково-технічний рівень. Як результат була розроблена система моделювання обробника природної мови.

До недоліків атестаційної роботи слід віднести те, що запропоновані алгоритми та інформаційна модель орієнтована на одну мову і не мають підсистеми локалізації.

Атестаційна робота магістра гр. ПЗСзм-18-1 Боричева Сергія Олександровича відповідає вимогам до атестаційних робіт магістрів і заслуговує оцінки «добре (85)».

Атестаційну роботу можна представити для захисту в ЕК за спеціальністю *121-Інженерія програмного забезпечення*, освітньо-професійною програмою *Програмне забезпечення систем*

Рецензент:

д.т.н., професор,
завідувач кафедри ШІ



В.О. Філатов