KLASYFIKACJA ZDAŃ W SYGNALE MOWY Z WYKORZYSTANIEM MODELU DTW

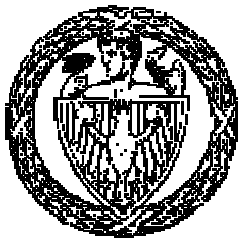
Włodzimierz Kasprzak

Raport IAiIS PW Nr 12-04

Warszawa, maj 2012 r.

POLITECHNIKA WARSZAWSKA
WYDZIAŁ ELEKTRONIKI I TECHNIK INFORMACYJNYCH
INSTYTUT AUTOMATYKI I INFORMATYKI STOSOWANEJ
ul. Nowowiejska 15 - 19, PL-00-665 WARSZAWA

Tel.: 0 22 234 7397, 0 22 825 52 80,
Fax: 0 22 825 37 19
E-mail: sekretariat@ia.pw.edu.pl



Streszczenie

Niniejszy raport jest pierwszym z dwóch raportów (IAiIS PW 12-04 i 12-05 [PRZ12]) dotyczącym badania przemysłowego n.t. klasyfikacji i rozpoznawania (izolowanych) zdań mówionych, wykonanego przez Autorów z Instytutu Automatyki i Informatyki Stosowanej Politechniki Warszawskiej. Celem niniejszego raportu jest przedstawienie algorytmu klasyfikacji mowy, realizującego 2 zadania: klasyfikację sekwencji słów i wyszukiwanie słów kluczowych w sygnale mowy, w oparciu o metodykę klasyfikcji sekwencji wzorców z „marszczeniem czasu" (ang. Dynamic Time Warping - DTW). Drugi raport [PRZ12] przedstawia proces rozpoznawanie słów kluczowych i izolowanych zdań w sygnale mowy w oparciu o metodykę Ukrytych Modeli Markowa (HMM).

Niniejszy raport został podzielony na trzy zasadnicze części, obejmujące kolejne fazy analizy sygnału mowy:

1. analizę akustyczną (od postaci sygnałowej do parametryzacji mowy – wyznaczeniu sekwencji cech numerycznych dla ramek sygnału);
2. klasyfikacja sekwencji cech metodą „marszczenia czasu" - DTW,
3. kwantyzacja wektorowa cech wspomagana wiedzą fonetyczną.

Przedmiot pracy jest już dobrze opracowany od strony teoretycznej. W literaturze przedmiotu znanych jest kilka typowych rozwiązań. Niniejszą pracę charakteryzuje autorskie ujęcie tematyki, wybór i powiązanie ze sobą w jednym algorytmie podstawowych funkcji analizy mowy, parametryzacja mieszana i klasyfikacja cech ramek wspomagane informacją fonetyczną o języku.

Opracowany algorytm kwantyzacji jest zasadniczo niezależny od języka. Wspomaganie wiedzą fonetyczną tego procesu oznacza, że wymaga się jedynie podania zestawu głosek (fonemów) języka i ich przynależności do jednej z 7 klas: 1) dyftong (dwu-samogłoska) i normalny monoftong (samogłoska), 2) skrócony monoftong, 3) półsamogłoska, 4) głoska zwarta, 5) głoska nosowa, 6) tnąca (szczelinowa, „frykatyw") i 7) afrykat. Próbki uczące powinny być opisane fonetycznie, tzn. z wykorzystaniem jedynie tych fonemów, które zostały dostarczone na wejście algorytmu klasyfikacji mowy.

Autor

Spis treści

1. Wprowadzenie

Na potrzeby niniejszej pracy zakłada się istnienie szerszego **systemu dialogowego mowy**, czyli systemu informatycznego do prowadzenia dialogu z komputerem w języku naturalnym. Punkty dialogowe tworzą graf rozmowy. Do każdego punktu dialogowego będzie dostępnych **N wzorców** o mieszanych charakterystykach płci i barwy głosu. Każdy wzorzec będzie opisany informacją o płci oraz będzie posiadać reprezentację tekstową, która może służyć jako pomoc przy segmentacji fonemów. Celem badań jest opracowanie **algorytmu klasyfikacji mowy**, realizującego 2 zadania:

1. klasyfikację sekwencji słów przy zadanym słowniku zawierającym do 20 sekwencji i
2. wyszukiwanie zadanych słów kluczowych lub kluczowych sekwencji słów.

DANE WEJŚCIOWE DLA ALGORYTMU:

1. plik audio z wypowiedzią użytkownika, przeznaczoną do klasyfikacji (format wav, 22050 kHz, 16bit lub 32 bit, mono),
2. lista sugerowanych tematów zawierająca następujące dane dla każdego elementu listy:

– identyfikator,
– łańcuch znaków (postać tekstowa elementu),
– plik audio (format wav, 22050 kHz, 16bit lub 32 bit, mono).

DANE WYJŚCIOWE Z SYSTEMU AUTOMATYCZNEGO KLASYFIKATORA MOWY:

Po klasyfikacji wzorca i wyborze punktu dialogowego, następuje zwrócenie listy kolejnych punktów dialogowych na podstawie grafu oraz stopnia dopasowania wzorca do bazy wzorców (procentowo). Lista sugerowanych tematów zawierająca następujące dane dla każdego elementu listy:

– identyfikator,
– wartość dopasowania danego elementu listy do wypowiedzi użytkownika, wyrażona w procentach.

INNE ZAŁOŻENIA:

1. liczba elementów na liście sugerowanych odpowiedzi **nie przekracza 20** elementów,
2. algorytm powinien działać **niezależnie od języka i płci** użytkownika,
3. algorytm powinien zakładać **minimalizacje elementów** w zbiorze uczącym,
4. algorytm powinien poprawnie klasyfikować zarówno **literalnie przeczytany przez użytkownika** jeden z sugerowanych tematów, jak i **przeczytane słowo lub słowa kluczowe** w niej występujące,
5. plik audio z wypowiedzią użytkownika **nie jest zaszumiony**,
6. użytkownik określa **manualnie** początek i koniec nagrania.

WYMAGANIA SZCZEGÓŁOWE

- Podczas projektowania algorytmu i prac badawczych należy szczególną uwagę zwrócić na struktury pośrednie, przyspieszające klasyfikację i rozdzielić wyraźnie fazę uczenia (dostarczania wzorców) od fazy rozpoznawania (klasyfikacji).
- Wypracowana metoda powinna maksymalnie korzystać z danych pół-przetworzonych, aby ograniczyć wymagania pamięciowe.
- Zakładamy, że faza uczenia i budowy danych dla klasyfikatorów może działać dużo wolniej (nawet offline) niż faza klasyfikacji.

2. Struktura klasyfikatora mowy

2.1 TRZY POZIOMY ANALIZY

Typowe podejście do komputerowej analizy zdań mówionych zakłada istnienie hierarchii trzech poziomów przetwarzania danych [RAB93, CSL00, WAL04, BEN08, KAS09]. W ich ramach wyróżniamy poniższe kroki przetwarzania (1)-(5).

A. analiza akustyczna

(1) analiza w dziedzinie czasu - akwizycja sygnału dźwiękowego i detekcja sygnału użytecznego mowy,

(2) analiza widmowa - przekształcenie ramek (okien) sygnału w dziedzinę częstotliwości,

(3) parametryzacja sygnału mowy - wyznaczenie wektorów (numerycznych) cech ramek (okien) sygnału;

B. analiza fonetyczna

(4) klasyfikator cech lub kwantyzacja wektorowa w terminach klas lub klastrów odpowiadających jednostkom fonetycznym mowy (tzw. trzy-fonom),

(4a) klasyfikator cech – uczenie klasyfikatora (np. geometrycznego, neuronowego, statystycznego) i wykorzystanie klasyfikatora do określania wiarygodności przynależności wektora cech do klasy fonetycznej,

(4b) kwantyzator wektorowy – klasteryzacja cech i wyznaczenie reprezentantów klastrów w trybie uczenia oraz kodowanie wektorów cech w trybie rozpoznawania;

C. analiza symboliczna

(5a) wyszukiwanie / rozpoznawanie słów kluczowych,

(5b) rozpoznawanie zdań.

Najbardziej rozpowszechnione podejścia stosowane w analizie symbolicznej mowy to: DTW („dynamic time warping" - dynamiczne marszczenie czasu) (omawiane w niniejszym raporcie) i modelowanie stochastyczne z użyciem HMM („Hidden Markow Model" – ukryte modele Markowa) (omawiane w kolejnym raporcie [PRZ12]).

2.2 STRUKTURA ALGORYTMÓW KLASYFIKACJI I ROZPOZNAWANIA ZDAŃ I SŁÓW MÓWIONYCH

Na rys. 2.1. przedstawiono podstawowe kroki przetwarzania i typy danych w zrealizowanych algorytmach klasyfikacji i rozpoznawania zdań. Zgodnie z ogólnie przyjętą metodyką wyróżnimy kolejne etapy analizy sygnału mowy:

1. **Analiza akustyczna**
   a. **Funkcje przetwarzania wstępnego** sygnału mowy (we/wy sygnału, ewentualna filtracja sygnału, detekcja sygnału mowy (VAD – „voice activity detector"), preemfaza,
   b. **Analiza widmowa** – funkcje wyznaczania ramek sygnału i okienkowej transformaty Fouriera (STFFT),
   c. **Parametryzacja** sygnału mowy – funkcje wyznaczania cech ramek sygnału (momenty widma, cechy mel-cepstralne, cechy dodatkowe takie, jak quasi periodyczność, dolnoprzepustowość, formanty).

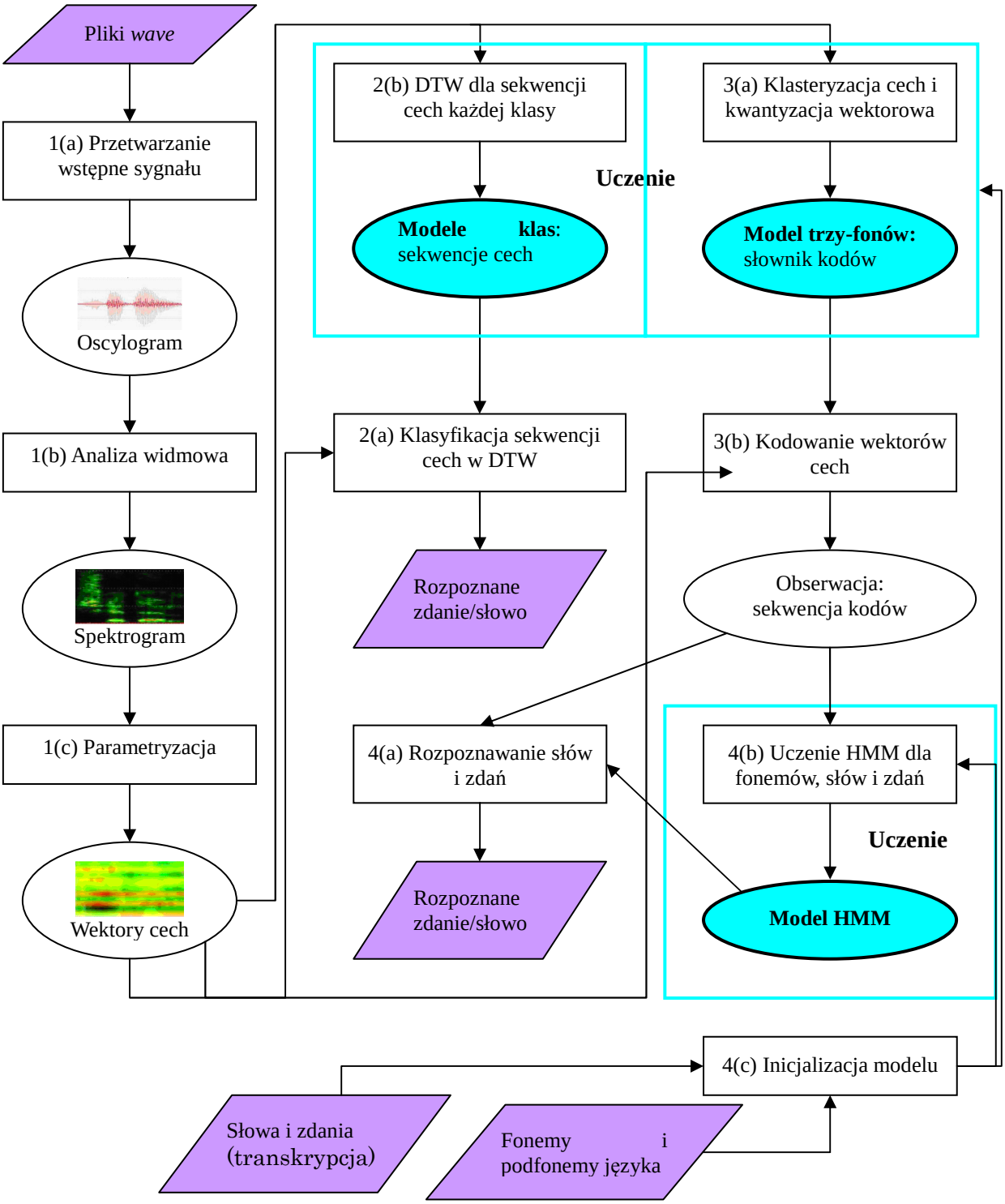
2. **Klasyfikacja sekwencji cech numerycznych**
    a. **Klasyfikator DTW** – dopasowanie ze sobą dwóch sekwencji wektorów cech według zasady „marszczenia czasu", uśrednianie sekwencji wektorów cech, dopasowywanie modelu zdania do sekwencji obserwowanych cech.
    b. **Uczenie klasyfikatora** – wyznaczanie wzorcowych sekwencji cech dla słów i zdań.

3. **Koder ramek**
    a. Funkcja **KoderCech**
        Kwantyzacja wektorowa cech - analiza skupień wektorów cech w zależności od rodzaju fonemu (spółgłoska/samogłoska, dźwięczna/bezdźwięczna, ustna/nosowa, F0 i inne cechy sygnału w ramce) – sterowanie (zwiększanie liczby klas od minimalnej do maksymalnej) procesem klatesteryzacji. W drugim etapie przewidziane jest różnicowanie reprezentantów klastra zależnie od płci mówcy i częstotliwości F0.
    b. Funkcja **KoderKomend**
        Kodowanie wektora cech poprzez wybór najbliższego reprezentanta klastra. Hierarchiczny klasyfikator, który uwzględnia różne części wektora cech zależnie od kategorii głoski.

4. **Rozpoznawanie słów i zdań na poziomie symbolicznym**
    a. Zastosowanie **stochastycznych modeli HMM** w procesie poszukiwania najlepszego dopasowania modeli słów i zdań do aktualnej sekwencji jednostek fonetycznych.
    b. Uczenie modeli HMM – tworzenie stochastycznych modeli słów i zdań.
    c. Inicjalizacja modelu – fonetyczny model języka i transkrypcje fonetyczne słów i zdań.

Etap 1 jest wspólny dla obu algorytmów – klasyfikacji lub rozpoznawania słów i zdań. Etap 3 ma charakter opcjonalny. Jego występowanie oznacza stosowanie w etapie 4 modeli HMM z dyskretnymi funkcjami wyjść. Przy braku etapu 3, funkcje wyjść w modelach HMM przyjmują postać mieszanin ciągłych rozkładów Gaussa.

W niniejszym raporcie przedstawione zostały etapy 1-3, w tym klasyfikator sekwencji cech stosujący zasadę „marszczenia czasu (DTW). Proces rozpoznawania w oparciu o symboliczny model HMM i jego proces uczenia przedstawione zostały w kolejnym raporcie [PRZ12].



Rys. 2.1 Struktura dwóch proponowanych algorytmów klasyfikacji i rozpoznawania słów i zdań mówionych.

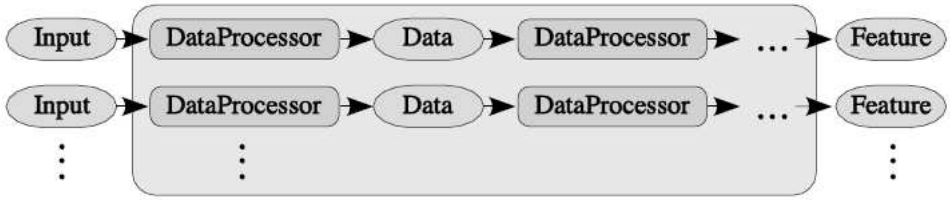
2.3 PRZYKŁADY INNYCH SYSTEMÓW ROZPOZNAWANIA ZDAŃ

Pomijając produkty komercyjne (np. Dragon systems, produkty IBM) można wymienić kilka rozwiązań publicznie dostępnych dotyczących rozpoznawania zdań mówionych.

W ramach projektu o charakterze „open source" w połowie lat 2000-ych powstał modułowy system **Sphinx-4** (Sun Microsystems, 2004) [WAL04]. Wyznacza on pewien standard implementacji podstawowych algorytmów do analizy mowy i jednocześnie zapewnia strukturę szkieletową („framework") dla tworzenia systemów analizy mowy.

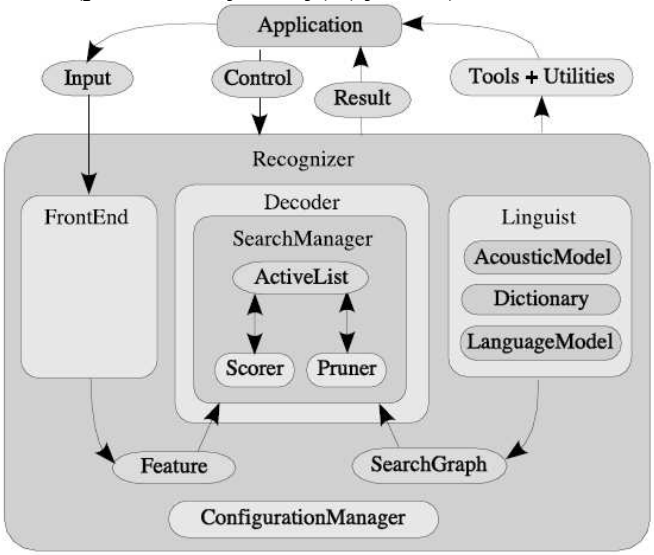
Niewątpliwie można też wymienić inne rozwiązania, wprawdzie nie o otwartym kodzie źródłowym, ale o dość dobrze opisanej metodyce – np. w przeszłości dostępne były dobrze rozwinięte narzędzia (np. CSLU [CSLU00]) lub też są dostępne narzędzia (np. HTK [HTK06]) do analizy sygnału mowy z wykorzystaniem modeli HMM. Dla stworzenia kompletnego systemu klasyfikacji wymagają one własnej konfiguracji lub dołączenia implementacji funkcji analizy akustycznej, a możliwości modyfikacji analizy symbolicznej (przy braku kodów źródłowych) są silnie ograniczone. Dlatego odniesiemy się w tej pracy jedynie do systemu Sphinx-4.

Wyróżnione przez nas dwa poziomy przetwarzania: analiza akustyczna i koder cech, tworzą w projekcie **Sphinx-4** poziom nazywany **FrontEnd** - ma on postać równoległych potoków analizy sygnałowej (rys. 2.2). Nasza metodyka jest podobna, gdyż także korzystamy z dobrze umotywowanych funkcji analizy sygnału a potok przetwarzania kończymy kodowaniem cech.



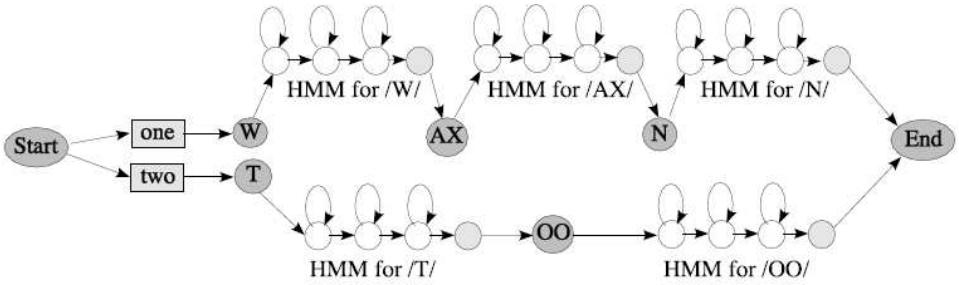
Rys. 2.2 Potokowa architektura podystemu do analiza akustycznej i kodowania cech w Sphinx-4 (źródło [WAL04])

W systemie szkieletowym **Spinx-4** nasz poziom symbolicznego modelowania zdań i klasyfikacji z użyciem takiego modelu odpowiada dwóm elementom: **Linguist** (nasz model języka i uczenie) i **Decoder** (proces klasyfikacji) (rys. 2.3).



Rys. 2.3 Ogólna struktura systemu Sphinx-4 [źródło WAL04]

Proces klasyfikacji postrzegany jest jako przeszukiwanie w przestrzeni stanów, np. z wykorzystaniem modelu HMM (rys. 2.4).



Rys. 2.4 Klasyfikacja słowa jako proces przeszukiwania 2-poziomowego modelu HMM dla słów z akceptowaniem aktualnej obserwacji jako sekwencji cech lub kodów (źródło [WAL04]).

3. Analiza akustyczna

Przedstawiono tu pierwszy etap klasyfikacji zdań i słów: (3.1) przetwarzanie wstępne sygnału i detekcja aktywnej mowy, (3.2) analiza widmowa i (3.3) parametryzacja - wyznaczanie cech.

3.1 STRUKTURA DANYCH KLASY CKLAMODOC

W niniejszym rozdziale kroki etapu analizy akustycznej zilustrowane zostaną kodami funkcji w przykładowej implementacji KlamoApp napisanej w języku C++ w środowisku MS Visual Studio 2005. Klasą głównego okna aplikacji okienkowej jest KLAMO. Szczegóły jej realizacji a także innych klas pomocniczych, związanych z wizualizacją, komunikacją z użytkownikiem i systemem plików pomijamy w niniejszym rozdziale.
Z punktu widzenia analizy mowy główną klasę stanowi klasa „dokumentu” dla okna: CKLAMODoc. Analiza mowy wymaga najpierw stworzenia i skonfigurowania obiektu tej klasy.

Fragmenty pliku CKLAMODoc.h bezpośrednio istotne dla funkcji analizy mowy

```
////////////////////////////////////////
// Program KlaMo
// Autor: Włodzimierz Kasprzak, IAiIS PW
// Data: 2.06.2005 - 30.09.2010
//////////
// Plik
// KLAMODoc.h : deklaracje klasy CKLAMODoc
////////////////////////////////////////

... // pominięte

// Odczyt i zapis plików WAV
#include "WAVCLASS\WaveOut.h"  //
#include "WAVCLASS\Wave.h"     //
#include "WAVCLASS\WaveDevice.h"        //
#include "WAVCLASS\WaveIn.h"   //

// STAŁE wyznaczające maksymalne obszary pamięci
#define MAKS_LA_MOWCOW 10
#define MAKS_LA_KLAS 52
#define MAKS_LA_PLIKOW 16
#define MAKS_LA_PFONEMOW 514
#define MAKS_LA_FONEMOW 244

// STAŁE dla przetwarzania sygnału
#define SKALA_ENERGII_OKNA 10000.0
#define WSPOLCZ_PREEMFAZY 0.9

// Domyślne wartości dla charakterystyk wektora cech
const double VAR_CECH[] = { 125.0, 22.0, 10.0, 10.0, 7.5, 6.0, 5.0, 4.5, 4.0, 4.0, 3.5, 3.5, 3.5,
                    3.0, 3.0, 3.0, 3.0, 2.5, 2.5,
                    45.0, 33.0, 17.0, 14.0, 11.0, 8.0, 7.0, 6.0, 6.0, 6.0, 5.0, 5.0, 4.5,
                    4.0, 4.0, 4.0, 4.0, 3.0, 3.0,
                    22.0, 50.0, 19.0, 0.33, 0.25, 70.0, 0.2, 1.0, 1.0, 1.0 };
const double MEAN_CECH[] = { 30.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                    0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
                    0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

```
                    25.0, 62.0, 6.0, 0.5, 0.6, 140.0, 0.25, 1.0, 1.0, 1.0 };

// Wagi cech dla dodatkowego skalowania elementów w odległości Mahalanobisa dwóch wektorów
const double WAGI_CECH[] = {2.0, 4.0, 4.0, 4.0, 2.0, 1.0, 1.0, 0.1, 0.1, 0.1,
                            0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
                           1.0, 2.0, 2.0, 2.0, 1.0, 0.5, 0.5, 0.1, 0.1, 0.1,
                            0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
                            1.0, 1.0, 0.2, 1.0, 1.0, 0.01, 0.01 };


// Definicja klasy
class CKLAMODoc : public COleDocument
{
        friend class ScrollOscyView;
        friend class ScrollSpektView;

protected: // create from serialization only
        CKLAMODoc();
        DECLARE_DYNCREATE(CKLAMODoc)

        // Własny konstruktor
        CKLAMODoc(int czProbek, int pCiszy, double wspOdstepuOkna, int lOkien, int lSlow,
                int lCechMFC,   char* nKatalogu, char* nSciezki);


... // pomijamy

// Implementation
public:
        virtual ~CKLAMODoc(); // Destruktor

#ifdef _DEBUG
        virtual void AssertValid() const;
        virtual void Dump(CDumpContext& dc) const;
#endif

// Pola chronione klasy
protected:
        int         LA_KLAS; // Liczba klas (komend)
        int         LA_PLIKOW; // Liczba próbek (plików) dla pojedynczej komendy
        long        LA_KOLUMN; // Liczba wektorów cech dla kwantyzacji
        int         LA_PFONEMOW; // Liczba klas podfonemowych
        int         LA_FONEMOW; // Liczba gruo podfonemów

        int         NORMALIZUJ_SPEKTROGRAM; // niepotrzebne w KlaMo

        ParametersDialog paramDialog; // Okno dialogowe do wprowadzania parametrów

        double * u[MAKS_LA_KLAS]; // Będą tu modele komend - sekwencje wektorów cech
        double * s[MAKS_LA_KLAS]; // Będą sekwencje indeksów podfonemow dla każdej komendy
        double * p[MAKS_LA_PFONEMOW]; // Będą tu modele podfonemów - sekwencje wektorów
                                      // cech - reprezentantow
        long *MAKS_DLUG_Klasy; // Spodziewane maksymalne długości wypowiedzi
                               // Jeśli ich nie ma program też będzie działać
        int *LA_KOLUMN_Klasy; // Użyteczna liczba kolumn reprezentanta klasy

        // Parametr przełączania trybów: klasyfikator DTW lub kwantyzer/koder
        bool bezSzumu;
```

```
        // Tablice nazw dla automatyzacji dostępu do plików z próbkami mowy
        CString NazwaSciezki, NazwaKatalogu;
        CString KatalogMowcy[MAKS_LA_MOWCOW];
        CString NazwaKlasy[MAKS_LA_MOWCOW][MAKS_LA_KLAS];
        CString NazwaKomendy[MAKS_LA_KLAS];
        CString NazwaPFonemu[MAKS_LA_PFONEMOW];

        // Tablice danych dla uczenia modeli komend
        double* licznikiSpekt[MAKS_LA_KLAS]; // Sumaryczne dane dla komend
        int mianownikiSpekt[MAKS_LA_KLAS]; // Sumaryczne dane dla komend
        double *cechySrednieModelu[MAKS_LA_KLAS]; // Srednie widma dla komend

        // Tablice danych dla kodowania podfonemów
        double* mozliweProbki; // Wszystkie wektory cech wszystkich próbek wszystkich komend
        double** probkiPFonemow;// Wskaźniki do kolumn - indywidualnych próbek
        int* klasaProbki; // Indeks klasy podfonemu dla aktualnego wektora cech podst.
        int* kategoriaProbki; // Indeks kategorii fonemowej dla aktualnego wektora cech dodat.

        // Tablice dla statystyk wypowiedzi
        // -- nadążne dla aktualnego mówcy
        double* meanCECH; // Wartości średnie elementów wektora cech
        double* stdDevCECH; // Wektor odchyleń standardowych dla elementów wektora cech
        double* minCECH; // Minimum wektora cech
        double* maxCECH; // Maksimum wektora cech
        double* centraMFCC; // Wartości średnie współczynników MFCC
        int numCECH; // liczba dotychczasowych wektorów cech

        // -- dla pojedynczej wypowiedzi
        double* meanCECH1; // Wartości średnie elementów wektora cech
        double* stdDevCECH1; // Wektor odchyleń standardowych dla elementów wektora cech
        double* minCECH1; // Minimum wektora cech
        double* maxCECH1; // Maksimum wektora cech
        double* centraMFCC1; // Wartości średnie współczynników MFCC

        // Tablice kodów cech
        double* reprezenCechPFonemow; // Reprezentaci wektorów cech dla klas podfonemow
        double** reprezenPFonemow; // Wskaźniki do reprezentantów cech dla klas podfonemow

        int* kategoriaPFonemow; // Indeks kategorii fonemowej dla reprezentanta klasy podfonemowej

        double* reprezenCechFonemow; // Reprezentanci wektory cech dla kategorii fonemow
        double** reprezenFonemow; // Wskaźniki do cech dla kategorii fonemow

        // Dla wyników odowania wektorów cech
        int* kategoriaKolumny; // Indeks kategorii fonemowej dla kolumny cech modeli komend
        int* klasaKolumny; // Indeksy klas podfonemowych dla kolumny cech modeli komend


    // Dla analizy aktualnego pliku wav
        // - VAD w dziedzinie czasu
        int LA_Slow; // Maksymalna liczba słów
        int liczbaSlow; // Aktualna liczba słów wykrytych w sygnale
        int* poczSlowa; // indeks początkowej próbki słowa
        int* koniecSlowa; // indeks końcowej próbki słowa

        // - analiza widmowa (spektralna)
        int liczbaOKIEN; // Ile okien użytecznego sygnału w aktualnej próbce
```

```
double *spektA; // Tu będą 2 ramki cech MFCC
double *spektAszer; // Tu będą 2 ramki szerokiego widma amplitudowego
double *windowFun; // Tu będzie funkcja okna Hamminga
double *filtryMEL; // Tu będą centra filtrów pasmowych według MEL skali

int *poczMEL, *koniecMEL;
double** wspolczMEL; // Współczynniki filtrów pasmowych w skali Mel
double *wagiMEL; // Normalizacja energii w pasmach Mel

double* wspolczMELWszystkie;

// Dla reprezentacji cech okien sygnału
int   liczbaCechMFC, lCechMFCC, lxDwaCechMFCC; // Liczba cech MFC i MFCC
int   liczbaCechDOD; // Liczba cech dodatkowych obok cech MFCC:
                                    // M_norm(1), M_norm,cen(2)
                                    // r_max, L_lp, L_res, F0
int liczbaCECH; // Długość wektora cech = lxDwaCechMFCC + liczbaCechDOD

int wierszeSpektrogramu; // Rozmiar y ramki cech MFCC
int kolumnySpektrogramu; // Rozmiar x ramki cech MFCC
int kolumnyUproszcz; // Rozmiar x widma amplitudowego
int okno;   // Liczba probek w 1-ym oknie = szerokosc transformaty FFT

double fSampling; // częstotliwość próbkowania
double czestotliwoscBazowa; // Czestotliwość zależna od fSampling i rozmiaru okna FFT

int odstepOkna; // Odstęp pomiedzy 2 kolejnymi oknami
int waska_ramka; // Położenie x znalezionej ramki w podwojnej ramce szerokiej
long poczatek_ramki; // początek użytecznego sygnalu (szerokiej ramki)
short typSpektrogramu; // Typ wyswietlanego spektrogramu:
                    0-brak, 1- cechyMFCC ,2- szerokie widmo
//////////////
// Metody klasy
////////////
// Zarządzanie modelem DTW
void ZerujModele();
void DodajCechyDoModelu(double* dodawany, int indeks);
void ZerujCharakterystyke(); // wartości średnie i odchylenie standardowe dla cech mówcy

// Metody dla przetwarzania wstępnego w dziedzinie czasu
int WczytajPlik(CString );
int InicjujAnalize(bool parVAD, int przerwy); // przetwarzanie wstępne w dziedzinie czasu:
                    // 1. dla DTW lub kodowania, 2. przerywaj wizualizacją lub nie
long ZnajdzRamke1(int); // wyznacza sekwencję okien sygnału użytecznego

// Metody dla analizy widmowej i detekcji cech
void ObliczCechyOkien(bool parCisza, bool parNorm, int przerwy, CString nazwa);
// Oblicza spektrogram, cechy MFCC i dodatkowe

// Metody dla kwantyzacji cech i kodowania komend
void UczenieSlownikaKodow(); // główna metoda dla wyznaczania słownika kodów
// druga jej część – kodowanie kolumn modelu DTW – w KlaMo nie jest wykonywana

// Wyznacza słownik kodowy (reprezentantów) dla próbek cech
int KwantyzatorCech(double warStopu, int liczbaCechMFCC, int liczbaCechDOD,
                    long lProbekCech, int lWierszy,
                    double** probkiCech,
                    int* klasaProbki, int* katProbki, int* katKlasy,
```

```
                        double** wynikReprezentKlasy, double** wynikReprezentKat);

        // Koduje jeden wektor cech według słownika kodowegp dla podfonemów
        void WykonajJednoKodowanie(CString nazwaPliku, int lCechPodst, int lCechDod);
                // Tworzy kody dla jednej wypowiedzi i zapisuje w pojedynczym pliku

        // Interfejs z analizą symboliczną – klasyfikatorem HMM
        void ZapiszJedneKody(CString nazwa, int* kody, double* jakosc, int lKolumn);

        // Pomocnicze we-wy sekwencji wektorów cech
        void WczytajCechy();
        void ZapiszCechy();

        // Metody dla klasyfikatora DTW
        // Automatyczne definiowanie klas i dołączanie indywidualnych klas
        void AutDefinicjaKlas();
        void AutDodanieKlasy();
        void ZapiszKlasy();
        void WczytajKlasy();
        void CzytajNazwyKlas(int, int, CString);
        void CzytajNazwyKomend(int, CString);
        void ZapiszSpektrogram(double *spekt, int lKolumn, int okno2, CString nazwaPliku);
        void ZapiszJedneCechy(CString nazwa, double* cechy, int lKolumn, int lWierszy);

        // Metody dla rozpoznawania komend – klasyfikacja DTW
        void TestujDTWlubKoduj(); // Wykonanie testowania modelu cech DTW lub kodowania
        void RozpoznajKomende(); // główna metoda rozpoznawania jednej komendy
        int WykonajKlasyfikacje(double*, double*); // dopasowanie jednej próbki z modelem
        void AutTestRozpoznawania(int, int, int, bool); // Automatyczne testowanie rozpoznawania


        ////////////
        // Dalsze pola
        int liczba_probekBezCiszy;
        bool otwarty;

        // Dla współpracy z plikami wav
        WAVEFORMATEX waveFormat;
        CWave tempWave;
        CWaveIn waveIn;
        CWaveDevice myDevice;
        CWaveOut waveOut;
        CString m_fileName; // Nazwa pliku wav

        // Dla pliku wav - dane wejsciowe do analizy
        int liczba_probek; // Dla aktualnej probki
        double *probki; // Tu beda probki sygnalu w czasie
        double *probkiBezCiszy; // Fragment pozostaly po usunieciu ciszy - poczatek wycinka sygnalu
        double poziomCiszy;          // prog dla amplitudy sygnału
        double maxWartosc;
        CWaveBuffer bufor;

        // Deklaracje metod obsługi zdarzeń
        .. //
        //////

private:
        // Konfiguruj obiekt - metoda jest wołana przez konstruktor
```

```
        void InicjujObiekt(int czProbek, int pCiszy, double wspOdstepuOkna,
                int lOkien, int lSlow, int lCechMFC,
                char* nKatalogu, char* nSciezki);

        void AlokujTabliceModelu(); // Alokacja tablic dla modelu cech
        void UsunTabliceModelu(); // Usunięcie tablic dla modelu cech
};
//////////////////////////////////////////////////////////////////////
```

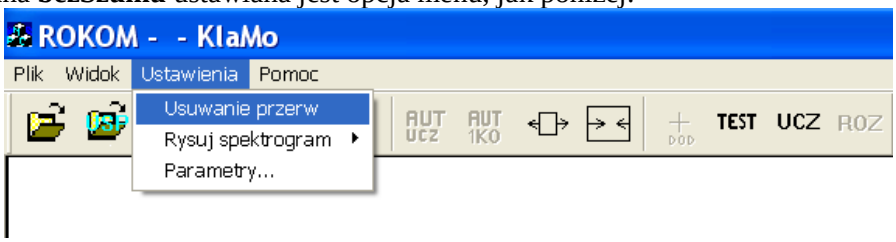
Główne funkcje użytkowe aplikacji **KlaMo**

Elementy poniższego paska narzędzi programu służą do wywołania w trybie obsługi zdarzeń podstawowych funkcji aplikacji. Nie wszystkie przyciski są obsługiwane w aktualnej implementacji KlaMo.

W odniesieniu do programu analizy mowy istotnymi przyciskami są:

Przycisk	Operacja	Uwagi
	Otwórz plik wav i wykonaj wstępną analizę w dziedzinie czasu i detekcję cech.	
	Zapisz model DTW do pliku względnie odczytaj model DTW z pliku.	
	Odtwórz aktualny sygnał mowy. Rozpocznij nagrywanie dźwięku (włącz mikrofon). Zakończ nagrywanie sygnału dźwięku.	
	Rozpocznij tworzenie modelu (lub tylko zestawu cech dla kwantyzatora) dla odpowiedniego zestawu komend i mówców. Należy podać plik konfiguracyjny w odpowiedzi na zapytanie.	O trybie racy decyduje zmienna **bezSzumu.**
	Automatyczne testowanie modelu DTW lub tworzenie kodów na podstawie słownika kodowego.	O trybie racy decyduje zmienna **bezSzumu.**
	Kwantyzacja – tworzenie słownika kodowego na podstawie zestawu wektorów cech.	
	Klasyfikacja DTW aktualnej jednej wypowiedzi przy założeniu istnienia modelu DTW.	

Uwaga: zmienna **bezSzumu** ustawiana jest opcja menu, jak poniżej:



3.2 ANALIZA W DZIEDZINIE CZASU

Funkcja konfigurowania analizy mowy – InicjujObiekt()

Proces analizy mowy (w naszej implementacji) rozpoczynamy od utworzenia i skonfigurowania obiektu klasy CKLAMODoc. Do tego celu służy odrębna metoda tej klasy **InicjujObiekt**().

```
 // Alokuj pamięć i dane pomocnicze dla głównego obiektu klasyfikatora mowy
void CKLAMODoc::InicjujObiekt(int czProbek, int pCiszy, double wspOdstepuOkna,
                              int lOkien, int lSlow, int lCechMFC,
                              char* nKatalogu, char* nSciezki)
{ …}
```

Wczytaj plik WAV

Do obsługi plików wav korzystamy z klasy CWave i współpracujących z nią klas biblioteki „open source”. Kody tych klas znajdują się w podkatalogu **WavClass**. Pewne własne modyfikacje tych funkcji były niezbędne z uwagi na rozszerzenie formatu Wave od 2001 r., kiedy biblioteka powstała.

Metoda **WczytajPlik**() odwołuje sie do powyższej biblioteki w celu wczytania pliku i konwersji próbek do bufora wejściowego według własności nagrania dźwięku.

Funkcja analizy wstępnej – **InicjujAnalize**()

Funkcja odrzuca początkowe i końcowe fragmenty nagrania, które mogą zawierać trzaski włączanego i wyłączanego mikrofonu. Dokonuje detekcji istnienia jakiejkolwiek informacji dźwiękowej w nagraniu. W przypadku „niezerowego” nagrania rozpoczyna się właściwa analiza sygnału pod kątem zawierania mowy. Wykonuje się wstępną normalizację amplitudy.

Następne koki przetwarzania wstępnego to: VAD (detekcja mowy), filtr preemfazy i (opcjonalnie) detekcja i usunięcie przerw między słowami.

VAD (Voice activity detector) - wykrycie użytecznej mowy

W tym kroku usuwane są fragmenty początkowy i końcowy zawierające jedynie szum. Sygnał mowy przeglądany jest najpierw od początku idąc do przodu w czasie a następnie od tyłu idąc wstecz w czasie. Algorytm sterowany jest dwoma progami:

1. jeden dotyczy minimalnej amplitudy, którą uznajemy za sygnał aktywny, a
2. drugi wyznacza maksymalny czas trwania impulsu szumowego.

Dopóki wartość bezwzględna amplitudy sygnału nie przekroczy progu ciszy, sygnał uznajemy za ciszę. Po wykryciu impulsu przekraczającego próg amplitudy, badamy średnią z wartości bezwzględnych próbek w kolejnym przedziale czasu nie większym niż drugi próg. Jeśli wartość średnia utrzymuje sie na poziomie mniejszym niż połowa pierwszego progu, to sygnał nadal uznajemy za ciszę. W przeciwnym razie wykrywamy początek względnie koniec aktywnej mowy.

Decyzja o tym, czy rozpoczęto aktywną część sygnału zawierającą mowę lub ją zakończono, podejmowana na podstawie samej amplitudy jest ryzykowna. Mowa rozpoczyna sie często tzw. zwarciem krtaniowym i często też kończy się głoskami o charakterze szumowym. W obydwu przypadkach amplituda sygnału jest niewielka i przypomina ciszę względnie szum.
Dlatego też, po określeniu hipotetycznych granic aktywnej mowy w sygnale, dołączane są z

przodu i z tyłu fragmenty o długości 4 okien sygnału.

Filtr preemfazy

Aby wzmocnić składowe o wyższych częstotliwościach w sygnale stosujemy w dziedzinie czasu przekształcenie zwanemu filtrem "preemfazy" o postaci:

$$x(t)' = x(t) - \varphi \cdot x(t-1), \quad \text{gdzie} \quad \varphi \in < 0.9, 099 > \qquad (3.1)$$

gdzie x(t) jest próbką sygnału w chwili *t*.

Wykrycie przerw między słowami (opcja)

W sytuacji, gdy analiza wstępna odbywa się na potrzeby klasyfikatora DTW (uczenie lub klasyfikacja) istotną potrzebą jest zredukowanie nadmiernych przerw między słowami. Odbywa się to w zbliżony sposób jak krok VAD dla początku i końca wypowiedzi. Ten krok także sterowany jest dwoma parametrami: minimalnej amplitudy aktywnego sygnału i minimalnego czasu pozostawionego dla zwarcia krtaniowego.

Implementacja powyższych trzech kroków – funkcja **InicjujAnalize**()

```
//
// Przetwarzanie wstępne aktualnego sygnału mowy
// Sygnał dźwiękowy jest już w wektorze probki[] obiektu CKLAMODoc
//
int CKLAMODoc::InicjujAnalize(bool parVAD, int przerwy)
{ .. }
```

ZnajdzRamke1() - segmentacja sygnału

Sygnał mowy jest dzielony na kolejne **ramki** o jednakowym czasie trwania. Przy częstotliwości próbkowania wynoszącej 22050 Hz przyjmujemy rozmiar okna za odpowiadający M = 512 próbkom, czyli czas trwania jednej ramki wynosi wtedy 23.2 *ms*. Odstęp pomiędzy kolejnymi ramkami ustalany jest obecnie na wartość od M/2 do M. W naszym przypadku oznacza to, że ramki rozpoczynają się względem poprzedniej ramki z opóźnieniem wynoszącym ($\Delta\tau$ = 11.6 – 23.2 *ms*).

3.3 ANALIZA WIDMOWA

Za analizę widmową sygnału i parametryzację cech odpowiada metoda **ObliczCechyOkien**().
W niniejszym punkcie omówimy standardowe kroki prowadzące do wyznaczenia cech MFCC. W następnym punkcie 3.3. przedstawimy pożyteczność cech dodatkowych dla normalizacji cech MFCC, tak, aby były zbliżone do siebie w wypowiedziach różnych mówców.

Funkcja ObliczCechyOkien()

Okienkowa transformata Fouriera

W celu przekształcenia każdej ramki sygnału w dziedzinę częstotliwości, czyli w celu dekompozycji sygnału na składowe częstotliwościowe o wielokrotnościach częstotliwości podstawowej, stosujemy okienkową Dyskretną Transformatę Fouriera (DFT) dla dyskretnego sygnału:

$$F(k,\tau) = \frac{1}{\sqrt{M}} \sum_{t=0}^{M-1} [x(\tau + t) e^{-i 2\pi k t / M} \cdot w_\tau(t)] \qquad (3.2)$$

gdzie M jest szerokością ramki rozpoczynanej w chwili $t = \tau$, x(t) jest próbką sygnału w chwili t, a $w_\tau(t)$ jest funkcją okna ustalonej postaci, przyjmującej wartości zerowe poza oknem o indeksie τ.

Funkcja okna ma za zadanie redukowanie zniekształceń (rozmycia) charakterystyki częstotliwościowej sygnału, powodowanej „wycięciem" okna i w praktyce „wyzerowaniem" reszty sygnału. Stosujemy okno **Hamminga -** ramka sygnału jest mnożona przez funkcję w postaci dzwonu :

$$w_\tau(t) = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi t}{M-1}), & dla \quad t = \{\tau+0, \tau+1, ..., \tau+M-1\} \\ 0, & w\ przeciwnym\ razie \end{cases} \tag{3.3}$$

gdzie M - *rozmiar okna* (liczba próbek sygnału w ramce).

W naszej implementacji okno Hamminga zostało stablicowane i przygotowane już w funkcji konfiguracji obiektu InicjujObiekt(). W implementacji DFT stosujemy szybką wersje transformaty (FFT). Kod implementacji okienkowej transformaty FFT znajduje się w pliku **Fft.cpp.**

Następnie tworzymy widmo amplitudowe (faza charakterystyki częstotliwościowej nie jest wykorzystywana w rozpoznawaniu mowy), tzn. obliczamy kwadrat amplitudy każdego współczynnika zespolonego **Fouriera** (dla k = 0 ,..., M-1):

$$FC(k,\tau) = \left| F(k,\tau) \right|^2 = \left| \frac{1}{M} \sum_{t=0}^{M-1} [x(\tau+t) e^{-i2\pi kt/M} \cdot w_\tau(t)] \right|^2; \tag{3.4}$$

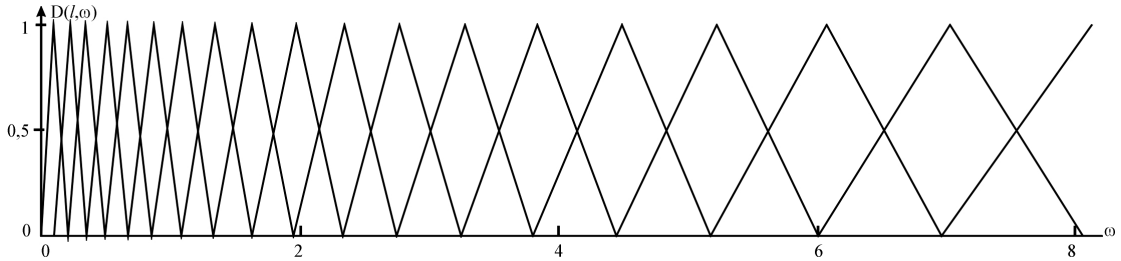
Skala MEL i filtry pasmowe

Ucho człowieka reaguje nieliniowo na częstotliwości sygnału dźwięku - różnice w zakresie niskich częstotliwości (< 1 kHz) są łatwiej wykrywane aniżeli podobne różnice w zakresie wysokich częstotliwości słyszalnego spektrum - czyli im wyższa częstotliwość tym gorsza dokładność - tym większe odstępy między kolejnymi pasmami są potrzebne dla zrekompensowania nieliniowości. Skala MEL skala została określona empirycznie i wynosi:

$$\omega_{mel} = 2595 \log\left(1 + \frac{\omega}{700[Hz]}\right), \tag{3.5}$$

gdzie ω jest częstotlwiością w liniowej skali.

Dla nieliniowego przekształcenia wektora współczynników tworzony jest zbiór filtrów dla kolejnych pasm częstotliwości, rozmieszczonych równomiernie w skali Mel, ale nierównomiernie w wejściowej skali liniowej (rys. 3.1).



Rys. 3.1 Trójkątne filtry pasmowe rozmieszczone wzdłuż skali częstotliwości, zgodnie z równomiernym rozmieszczeniem w Mel-skali.

Tablice danych dla wszystkich filtrów trójkątnych w Mel skali zostały przygotowane w funkcji **InicjujObiekt**().

Współczynniki "mel-spektralne"

Wykorzystujemy zbiór l trójkątnych filtrów $D(l, k)$ dla obliczenia (L = 42) tzw. współczynników *mel-spektralnych* $MFC(l, \tau)$ dla każdej ramki sygnału:

$$MFC(l,\tau) = \sum_{k=0}^{M-1} [D(l,k) \cdot FC(k,\tau)], \qquad l = 1,...,L \tag{3.6}$$

Wartość pojedynczego współczynnika *MFC* odpowiada ważonej sumie wartości *FC* należących do zakresu trójkątnego filtra pasmowego odpowiadającego danemu współczynnikowi *MFC*.

Współczynniki mel-spektralne obliczane są w dedykowanej funkcji **ObliczCechyMFC**().

Współczynniki mel-cepstralne

Po zastosowaniu funkcji logarytmu do współczynników MFC i odwrotnej transformacie Fouriera uzyskujemy współczynniki mel-cepstralne MFCC (*mel-frequency cepstral coefficients*). Całościowo przekształcenie ramki sygnału w wektor współczynników MFCC jest przykładem tzw. przekształcenia homomorficznego, które pozwala nam oddzielić sygnał użyteczny od charakterystyki impulsowej toru akwizycji, która nakładanie modeluje funkcja splotu.
Tym samym wyznaczamy K (=12 -18) współczynników mel-cepstralnych według wzoru:

$$MFCC(k,\tau) = \sum_{l=0}^{L-1} [\log MFC(l,\tau) \cdot \cos(\frac{k \cdot (2l+1)\pi}{2L})], \qquad k = 1,..., K\,. \tag{3.7}$$

Zamiast pełnej transformaty odwrotnej Fouriera wystarczy przekształcenie kosinusowe, gdyż wartości MFC są liczbami rzeczywistymi a nie zespolonymi.

Współczynniki MFCC i ich gradienty obliczane są bezpośrednio w funkcji **ObliczCechyOkien .**

3.4 PARAMETRYZACJA SYGNAŁU MOWY

Wektor cech

Zakładamy, że wektor cech ramki sygnału będzie niejednorodny (tzw. parametryzacja mieszana) i będzie zawierał przynajmniej 44 cechy pogrupowane w trzech grupach o różnym znaczeniu i wadze (Tab. 3.1):

Tab. 3.1 Przyjęty w pracy wektor cech sygnału mowy.

Ident.	Oznaczenie
c0	E_mel
c1	mfcc_1
c2	mfcc_2
...	...
c18	mfcc_18
c19	ΔE_mel
c20	Δmfcc_1
c21	Δmfcc_2
...	...
c37	Δmfcc_18
c38	E
c39	M1
c40	MC2
c41	r_max
c42	L_p
c43	F0

Wyjaśnienie grupo cech podanych w tabeli 3.1:

Grupa 1)

- c0 : amplituda całkowita sygnału sumowana po pasmach melowych,

Współczynniki MFCC

- c1-c18: 18 cech mel-cepstralnych (MFCC) tworzonych w oparciu o energie poszczególnych pasm częstotliwościowych, znormalizowane uprzednio przez energię łączną ramki;

Grupa 2)

Cechy różnicowe (dynamika współczynników MFCC w czasie):

- c19 : gradient cechy c0 (energii) w czasie (wyznaczony w oparciu o kolejnych 5 ramek),
- c20 – c37: gradienty cech MFCC w czasie (wyznaczone dla kolejnych 5 ramek),

Grupa 3)

Cechy dodatkowe dla odróżnienia głównych klas głosek od siebie: 1) silnie dźwięczne, 2) słabo dźwięczne, 3) bezdźwięczne o charakterze szumowym, 4) bezdźwięczne o charakterze plozyjnym (wybuchowym):

- c38: energia całkowita sygnału

Momenty widma, które są istotne dla głosek szumowych:

- c39: M1 – unormowany moment globalny pierwszego rzędu (środek ciężkości widma w ramce)
- c40: MC2 – unormowany moment centralny drugiego rzędu
- c41: r_max - quasi periodyczność (istotna dla rozróżnienia - bezdźwięczna / dźwięczna),
- c42: L_p – dolnopasmowość sygnału,
- c43: F0 - położenie częstotliwości podstawowej (w Hz, tylko dla głosek dźwięcznych)

Uwaga:
F0 to cecha wspomagająca jedynie normalizację widma dla różnych mówców.

Statystyka cech

Wstępna analiza rozkładu wartości cech (rys. 3.2-3.5) dla próbek uczących czterech zestawów zdań obejmuje współczynniki MFCC (cechy 1 – 19) i cechy dodatkowe (38-43, pomijamy cechę 44).
Widzimy wyraźne różnice zarówno co do przedziału wartości, jak i wartości średniej, co w praktyce bywa wynikiem różnicy cech osobniczych i zastosowanego mikrofonu (w teorii: odpowiedzi impulsowej toru modulacji i akwizycji sygnału).
W przypadku zestawów „Sylwia", „Michał" i „Olga" widoczne są nienaturalne zmiany dźwięczności i koncentracji energii w różnych częściach widma. Zestawy „Sylwia" i „Olga" posiadają bardzo równomierne rozkłady energii co wskazuje na osłabienie głosek dźwięcznych kosztem szumowych.
Dodatkowo zestaw „Olga" koncentruje swoją energię poza dolnym pasmem <0, 1 kHz>, co jest dość nienaturalne. Z kolei głos :Michał" jest wysoce dźwięczny i posiada także nadmiernie wysoki średni współczynnik „dolnopasmowości sygnału.

Cecha:	mean,	min,	max,	std.dev.
0:	27.621	1.000	148.261	26.333
1:	26.825	-42.615	64.341	22.881
2:	-1.174	-25.346	35.889	9.295
3:	9.697	-17.307	45.512	9.985
4:	-3.980	-34.114	16.409	7.362
5:	-1.083	-17.537	20.051	5.065
6:	2.943	-14.025	20.206	4.830
7:	-1.405	-19.296	14.441	5.521
8:	1.411	-14.497	14.469	4.097
9:	2.878	-9.870	14.842	3.194
10:	1.963	-9.679	18.793	3.437
11:	1.495	-11.499	11.385	2.839
12:	-0.136	-11.071	9.622	3.053
13:	1.767	-9.013	11.675	2.910
14:	-0.096	-8.533	9.496	2.330
15:	-0.124	-8.787	8.902	2.487
16:	0.714	-8.867	8.355	2.111
17:	1.428	-6.367	10.846	2.145
18:	-0.269	-8.348	10.794	2.143
19:	-0.268	-33.942	43.632	9.152
20:	0.048	-24.187	24.275	6.731
21:	0.093	-12.999	11.529	3.087
22:	0.034	-9.089	13.477	2.621
23:	0.056	-11.349	8.957	2.243
24:	-0.010	-6.631	6.498	1.586
25:	0.042	-5.857	5.377	1.449
26:	0.023	-6.035	6.454	1.689
27:	0.011	-5.868	5.550	1.373
28:	0.024	-6.489	4.278	1.048
29:	-0.014	-4.637	4.836	1.076
30:	-0.005	-4.453	5.111	0.931
31:	-0.007	-4.997	4.522	0.923
32:	-0.002	-4.703	3.402	0.908
33:	0.007	-3.454	3.842	0.765
34:	-0.004	-3.122	3.693	0.798
35:	0.003	-2.647	4.795	0.659
36:	0.002	-4.024	3.336	0.712
37:	-0.008	-2.573	4.501	0.696

Rys. 3.2 Charakterystyka cech podstawowych dla zestawu „001m (25x15)”

Cecha:	mean,	min,	max,	std.dev.
0:	28.932	1.005	180.452	26.001
1:	34.421	-46.847	81.226	21.724
2:	2.998	-32.877	41.358	11.138
3:	10.112	-31.529	49.240	9.665
4:	-2.228	-33.352	27.775	7.684
5:	-1.108	-25.197	24.822	6.102
6:	0.105	-17.239	20.188	5.034
7:	-1.301	-18.561	15.336	4.546
8:	0.226	-20.102	12.746	4.022
9:	0.095	-16.746	14.846	4.349
10:	2.045	-12.985	19.612	3.883
11:	-1.363	-19.838	9.969	3.800
12:	2.998	-10.080	15.681	3.463
13:	0.804	-12.035	11.756	2.983
14:	-1.532	-15.730	12.965	3.202
15:	0.907	-10.926	12.152	2.904
16:	-0.717	-14.108	10.633	3.055
17:	-0.394	-12.882	12.507	2.717
18:	0.017	-11.721	14.343	2.700
19:	-0.145	-58.614	42.196	10.044
20:	0.109	-27.730	23.992	7.345
21:	0.034	-13.201	15.152	3.863
22:	0.044	-10.505	12.637	3.005
23:	0.069	-10.424	10.080	2.600
24:	0.049	-9.389	8.426	1.833
25:	-0.011	-7.546	5.866	1.564
26:	0.011	-6.452	8.921	1.497
27:	0.029	-6.661	5.999	1.333
28:	0.022	-5.939	7.041	1.403
29:	-0.006	-5.228	5.834	1.266
30:	-0.002	-4.735	5.395	1.276
31:	0.007	-5.294	4.107	1.107
32:	0.010	-5.269	5.168	0.951
33:	-0.003	-4.803	5.197	1.030
34:	0.000	-4.004	4.283	0.919
35:	0.010	-4.749	4.838	0.989
36:	0.001	-4.831	4.383	0.897
37:	0.007	-5.638	5.061	0.887

Rys. 3.3 Charakterystyka cech podstawowych dla zestawu „Sylwia (30x10)”

Cecha:	mean,	min,	max,	std.dev.
0:	77.466	1.002	396.805	66.764
1:	68.171	-34.860	115.825	24.594
2:	10.963	-44.418	52.733	15.379
3:	12.722	-35.061	77.424	16.034
4:	-2.281	-52.173	35.218	10.826
5:	6.884	-22.088	40.602	8.142
6:	-6.912	-40.732	24.500	9.600
7:	4.646	-25.230	36.335	7.350
8:	3.014	-19.594	28.753	6.336
9:	-1.709	-24.536	19.352	5.638
10:	2.271	-18.201	22.157	5.107
11:	-0.243	-15.497	14.322	4.222
12:	0.734	-15.889	19.043	4.031
13:	2.448	-15.007	20.687	4.198
14:	2.031	-11.595	20.422	3.919
15:	1.820	-13.765	15.639	3.618
16:	1.599	-12.936	16.452	3.341
17:	1.517	-9.527	14.216	2.801
18:	1.467	-10.374	11.458	2.909
19:	-0.191	-84.361	104.439	20.323
20:	0.203	-26.166	34.202	8.326
21:	0.062	-18.504	17.732	4.347
22:	0.022	-15.687	16.962	4.488
23:	0.061	-11.994	10.987	3.205
24:	0.015	-10.167	9.047	2.498
25:	-0.033	-12.929	10.866	2.777
26:	0.011	-9.806	10.739	2.336
27:	0.043	-7.740	8.598	1.963
28:	0.042	-7.182	7.771	1.894
29:	0.018	-5.646	6.982	1.519
30:	-0.010	-5.255	5.219	1.315
31:	-0.002	-4.743	5.649	1.259
32:	0.008	-4.386	5.506	1.274
33:	0.009	-5.013	4.387	1.241
34:	-0.010	-5.391	4.001	1.163
35:	-0.006	-4.270	4.231	1.044
36:	0.012	-3.724	3.583	0.903
37:	0.001	-5.649	3.560	0.894

Rys. 3.4 Charakterystyka cech podstawowych dla zestawu „Michał (20x10)”

Cecha:	mean,	min,	max,	std.dev.
0:	8.667	1.001	219.326	8.207
1:	8.734	-64.986	72.754	24.503
2:	-1.033	-46.679	42.930	15.664
3:	15.895	-19.889	55.552	9.658
4:	-11.675	-52.448	16.341	9.278
5:	6.026	-19.265	37.274	6.108
6:	-5.523	-27.643	12.355	6.070
7:	2.427	-15.194	25.109	4.970
8:	-2.246	-23.833	15.821	4.868
9:	0.562	-12.945	19.029	4.232
10:	-0.551	-16.158	16.447	4.198
11:	-1.299	-14.928	14.589	4.252
12:	-0.442	-15.356	11.984	3.458
13:	-0.248	-12.670	12.450	3.393
14:	-2.610	-15.147	10.083	3.135
15:	-0.839	-14.050	14.791	3.288
16:	-1.499	-12.027	12.362	3.079
17:	-0.991	-11.718	14.310	3.572
18:	-0.566	-12.384	15.122	3.780
19:	-0.038	-87.284	45.469	3.312
20:	0.614	-27.888	39.442	7.790
21:	0.005	-14.549	16.482	4.759
22:	-0.004	-11.173	12.439	2.698
23:	0.021	-10.328	11.892	2.878
24:	-0.011	-7.974	7.307	1.857
25:	0.010	-7.929	9.768	1.746
26:	0.026	-5.992	8.855	1.590
27:	-0.012	-6.887	5.675	1.465
28:	-0.026	-4.779	4.874	1.273
29:	0.009	-5.957	5.534	1.255
30:	-0.045	-5.335	5.241	1.376
31:	-0.004	-4.781	3.853	1.063
32:	-0.010	-4.987	5.193	1.126
33:	-0.036	-3.689	5.834	1.028
34:	-0.013	-3.945	4.670	1.077
35:	-0.026	-4.585	5.015	0.990
36:	-0.029	-4.347	5.686	1.147
37:	-0.010	-4.495	4.919	1.157

Rys. 3.5 Charakterystyka cech podstawowych dla zestawu „Olga (20x10)”

Cechy dodatkowe:

38:	25.096	0.556	146.271	22.673
39:	63.355	3.831	200.339	42.319
40:	1.073	0.000	55.138	3.104
41:	0.360	0.000	0.986	0.287
42:	0.604	0.094	0.994	0.244
43:	107.163	60.411	355.645	47.960
44:	0.396	0.006	0.906	0.244

Rys. 3.6 Charakterystyka cech dodatkowych dla zestawu „001m (25x15)”.

Cechy dodatkowe:

38:	24.341	0.448	167.952	21.681
39:	58.642	1.505	238.541	50.206
40:	6.593	0.000	721.579	21.377
41:	0.498	0.000	1.000	0.335
42:	0.593	-0.000	0.992	0.251
43:	147.329	60.411	355.645	72.363
44:	0.407	0.008	1.000	0.251

Rys. 3.7 Charakterystyka cech dodatkowych dla zestawu „Sylwia (30x10)”

Cechy dodatkowe:

38:	37.088	0.381	200.664	30.661
39:	20.760	1.007	120.021	25.365
40:	0.131	0.000	90.831	1.717
41:	0.798	0.000	1.000	0.310
42:	0.793	0.218	1.000	0.181
43:	130.344	60.411	355.645	51.659
44:	0.207	0.000	0.782	0.181

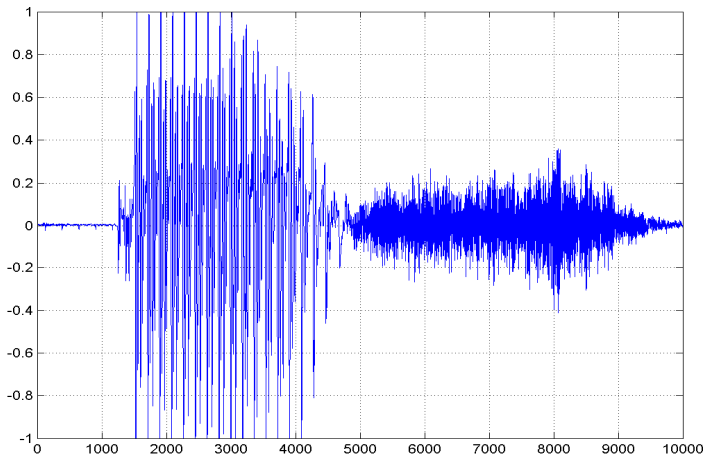
Rys. 3.8 Charakterystyka cech dodatkowych dla zestawu „Michał (20x10)”

Cechy dodatkowe:

38:	20.429	0.491	142.283	23.283
39:	81.401	1.893	191.972	30.973
40:	0.684	0.000	746.291	10.005
41:	0.146	0.000	1.000	0.220
42:	0.539	0.025	0.983	0.212
43:	154.753	60.411	355.645	85.540
44:	0.461	0.017	0.975	0.212

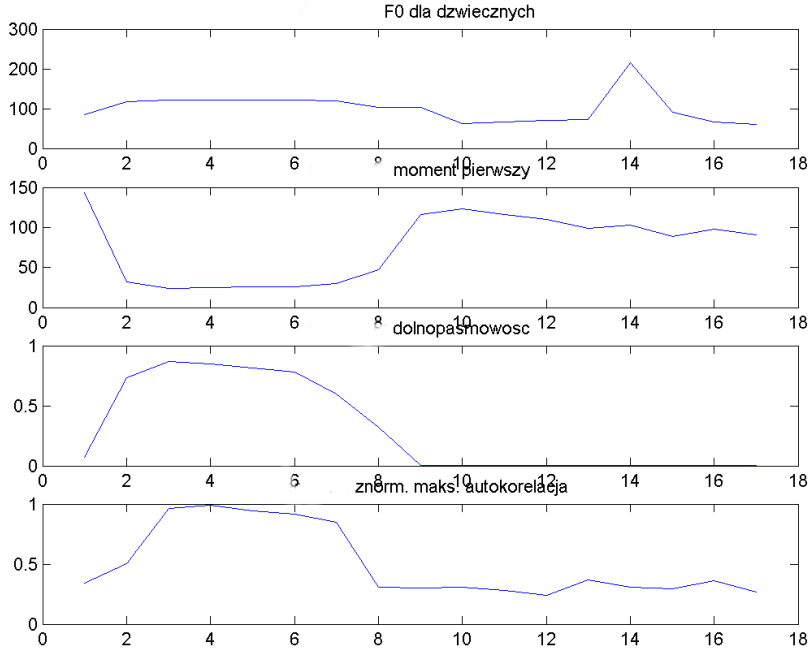
Rys. 3.9 Charakterystyka cech dodatkowych dla zestawu „Olga (20x10)”

Ilustracja przebiegu sygnału w czasie (oscylogram) dla dwóch wypowiedzi „oś” i „oś dwa dół start stop” (rys. 3.10 i 3.12) wraz z odpowiadającymi przebiegami wartości niektórych cech dodatkowych (rys. 3.11 i 3.13) pozwala wyjaśnić ideę wstępnego podziału głosek (fonemów) na kategorie. Podział ten będzie wykorzystany w kolejnych etapach algorytmu klasyfikacji mowy – podczas kwantyzacji wektorowej cech i podczas analizy symbolicznej w modelu HMM.

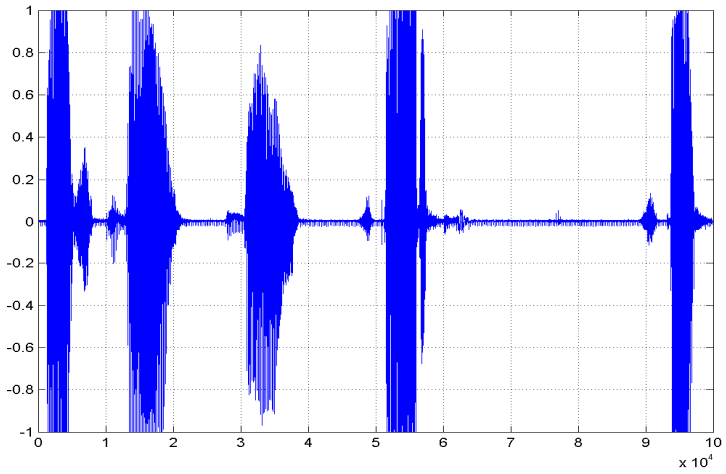


Rys. 3.10 Oscylogram wypowiedzi słowa „oś”

W słowie „oś” występuje silna samogłoska „o” i słaba głoska szczelinowa „ś” (rys. 3.10). Dla dźwięcznej samogłoski możliwy jest pomiar chwilowej częstotliwości podstawowej mówcy. Jest on bardzo stabilny i wartość ta wynosi ok. 120 Hz. Pomiar dla głoski szczelinowej nie jest miarodajny i pomija się go. Zamiast tego stosuje się interpolacje pomiędzy kolejnymi pomiarami dla głosek dźwięcznych (rys. 3.11).
Kolejne cechy dodatkowe - „dolnopasmowość” i „znormalizowana autokorelacja” pozwalają wyznaczyć okna z mową dźwięczną. Dla takich okien sygnału obie cechy przyjmują wartości w przedziale <0.8 – 1.0>. Koresponduje z tym niska wartość momentu pierwszego, który podaje indeks częstotliwości, wokół której koncentruje się energia w oknie sygnału.



Rys. 3.11 Przebieg wartości wybranych cech dodatkowych w wypowiedzi słowa „oś”.

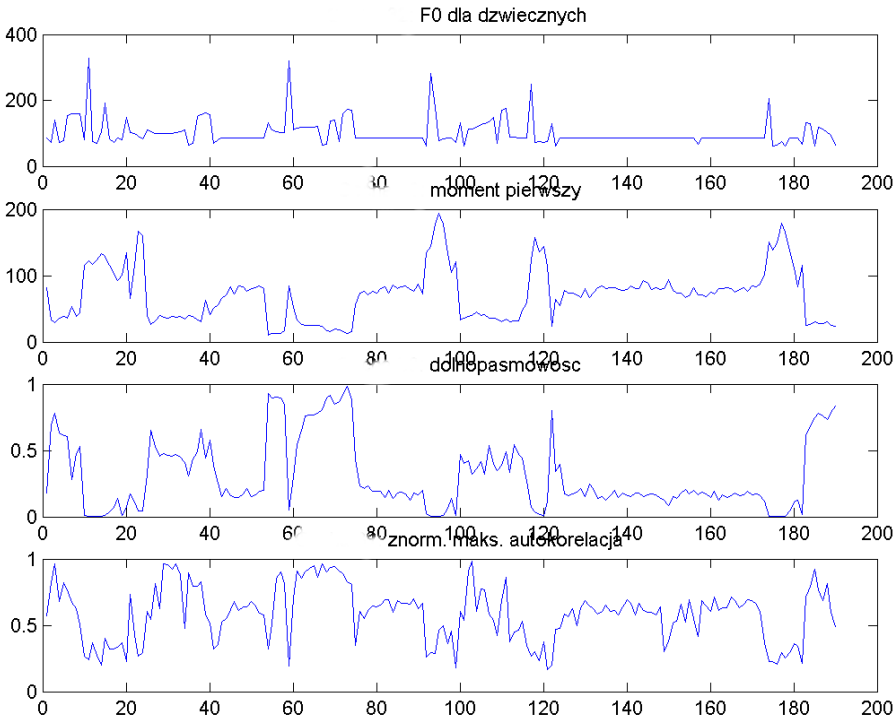


Rys. 3.12 Oscylogram zdania „Oś dwa dół start stop”

W wypowiedzi zilustrowanej na rys. 3.12 stabilny pomiar F0 na poziomie 60 (pokazany na rys. 3.13) jest w istocie błędem i dotyczy okresów ciszy. Także wysokie wartości F0, o charakterze nagłych skoków, są błędne. Podsumowując, FO mierzone jest w miarodajny sposób jedynie dla okien zawierających sygnał głosek dźwięcznych.

Normalizacja cech dla różnych mówców - NormalizujSpektogram()

Normalizacja cech występuje w dwóch miejscach:
1. normalizacja spektrogramu (zobacz funkcję **NormalizujSpektrogram()**)
2. nadążne obliczanie wartości średniej współczynników MFCC dla danego mówcy i ich odejmowanie od współczynników MFCC dla aktualnego mówcy (realizowane bezpośrednio w funkcji **ObliczCechyOkien**()).



Rys. 3.13 Pomiar F0 i wybranych cech dodatkowych dla wypowiedzi „Oś dwa dół start stop”.

4. Model sekwencji cech

Przedstawiono alternatywny do modelu HMM klasyfikator oparty o zasadę „marszczenia czasu” (DTW), zwykle wystarczająco skuteczny dla zestawów o niedużej liczbie zdań.

Model komendy w klasyfikatorze DTW składa się z sekwencji cech uśrednionych po wszystkich próbkach uczących. Zarówno podczas uczenia modelu jak i późniejszej klasyfikacji DTW stosowany jest zbliżony algorytm poszukiwania najlepszego dopasowania dwóch sekwencji cech.
Implementacje funkcji uczenia modelu, dopasowania dwóch sekwencji cech i klasyfikatora DTW podane są w pliku **KlasyfikatorDTW.cpp**.

4.1 Zasada „marszczenia czasu”

Dynamiczne „marszczenie czasu” (ang. "*dynamic time warping*", DTW), to początkowa metoda rozpoznawania mowy, stosowana głównie przed 1990 r. W tym rozwiązaniu w bazie modeli występują prototypy wypowiedzi w postaci sekwencji wektorów cech: $\{Y_1, ... ; Y_M\}$.
Niech X będzie aktualną sekwencją wektorów cech. Klasyfikacja sekwencji X polega na wyznaczeniu jego odległości od wszystkich reprezentantów klas $D(X; Y_i)$ nawet wtedy, gdy wzorce posiadają **różne czasy** trwania. W procesie klasyfikacji stosujemy regułę decyzyjną minimalizacji odległości:

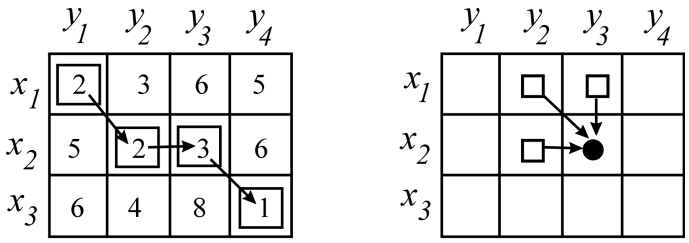
$$l = \arg\min_{l} D(X, Y_l) \tag{4.1}$$

O procesie klasyfikacji zakładamy, że możliwa jest **dekompozycja miary** odległości, tzn. odległość dwóch sekwencji cech składa się z **sumy lokalnych** odległości, $d_{ij} = d(x_i, y_j)$, liczonych dla par pojedynczych cech, wzdłuż ścieżki w przestrzeni dopasowania stanowiącej rozwiązanie:

$$D_{ij} = \begin{cases} 0 & ; \quad i = j = 0 \\ \min\{D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1}\} + d_{ij} & ; \quad i > 0, j > 0 \\ \infty & ; \quad pozostale \end{cases} \tag{4.2}$$

Poszukiwana jest **ścieżka o specyficznej** postaci:
a) ścieżka nie może "***cofać się***" ani po wcześniej dopasowane segmenty. ani też po wcześniejsze elementy modelu;
b) możliwe jest powtórzenie na ścieżce rozwiązania (raz po razie) elementu reprezentanta w modelu lub elementu aktualnej sekwencji.

Przykład.
Dana jest sekwencja trzech segmentów (x_1 x_2 x_3). Podczas sekwencji klasyfikacji tej sekwencji z sekwencją elementów modelu (słowa) (y_1 y_2 y_3 y_4) wymagane jest dopasowanie jednego segmentu z dwoma elementami modelu, gdyż obie sekwencje (segmenty, model) różnią się ilością elementów. Pojedynczy krok posiada jedynie elementarną długość - ale poprzednikiem może być zarówno stan poprzedni "w poziomie” (jak w programowaniu dynamicznym) jak i "w pionie" (rys. 4.1).



Rys. 4.1 Ilustracja kosztów dopasowania i postaci ścieżki w metodzie DTW

W naszej implementacji procesu poszukiwania najlepszej ścieżki dopasowania dwóch sekwencji wprowadziliśmy pewne ograniczenie na kroki „marszczenia" czasu. Za takie kroki uznajemy powtórzenie się dla następnego elementu sekwencji referencyjnej dopasowania z tym samym elementem co poprzednio (krok poziomy na rys. 4.1b) lub „przeskoczenie" nad następnym elementem (krok pionowy plus krok po przekątnej, na rys. 4.1b). Krokiem nie marszczącym jest krok po przekątnej. Otóż ograniczamy możliwość powtarzania się tego samego rodzaju marszczenia czasu dwa razy. W następnym kroku dopasowania takie marszczenie jest zabronione. Kierowaliśmy się zamiarem zwiększenia efektywności funkcji przeszukania. Warto uzupełnić, że ścieżka pasowania dwóch sekwencji może rozpocząć się na 20 różnych sposobów (liczbę 20 przyjęliśmy empirycznie i może być ona zmieniona). Odnosząc to do sekwencji referencyjnej – początek aktualnej sekwencji umieszczany jest w zakresie od -10 do 10 elementów (kolumn cech) względem jej początku. Taka zmienność rekompensuje ograniczanie kroków marszczenia czasu.

Tworzenie modelu – **DodajCechyDoModelu**()
Każda nowa wypowiedź ze zbioru uczącego dopasowywana jest do już istniejącego dotychczasowego modelu cech danej komendy a następnie tak znaleziona najlepsza sekwencja dodawana jest do sekwencji referencyjnej (modelu) i sumaryczny wynik jest uśredniany.
Implementacją tej operacji jest metoda **DodajCechyDoModelu**().

4.2 KLASYFIKACJA DTW

Podstawową funkcją dla procesu klasyfikacji DTW jest **WykonajKlasyfikacje**(). Jej zadaniem jest znalezienie najlepszego dopasowania sekwencji kolumn (wektorów cech) modelu z aktualnie obserwowaną sekwencją. Są tu podobne ograniczenia dla marszczenia czasu i możliwości rozpoczynania dopasowania w zakresie od -10 do 10 elementów (kolumn cech) względem początku sekwencji referencyjnej jak w poprzedniej funkcji **DodajCechyDoModelu**(). Jednak dodatkowym elementem oceny jakości znalezionego dopasowania jest stosowanie kary wtedy, gdy część kolumn aktualnej obserwacji nie została dopasowana z żadnym elementem sekwencji modelowej. Badaliśmy tu dwa rozwiązania: 1) określanie relacji energii sygnału dla części pozostawionej bez dopasowania do całej energii sygnału obserwowanego, 2) określanie relacji liczby ramek części pozostawionej bez dopasowania do liczby ramek obserwowanej sekwencji. Te rozwiązania nie muszą być alternatywne, mogą być łączone ze sobą. Rozwiązanie 2 jest prostsze i w praktyce zostało zastosowane, ale kody programu zawierają implementacje obu rozwiązań.

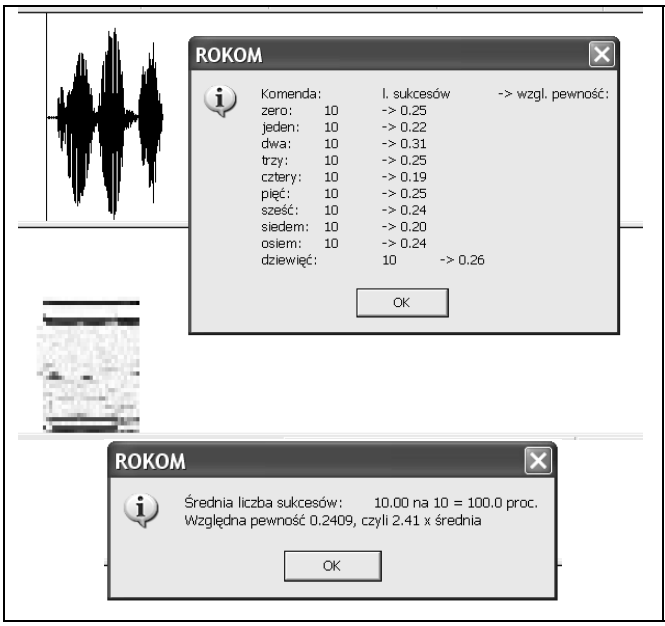
Dopasowanie 2 sekwencji - funkcja WykonajKlasyfikacje()

```
//////////////////////////////////////////////////////
// Klasyfikacja DTW
//
```

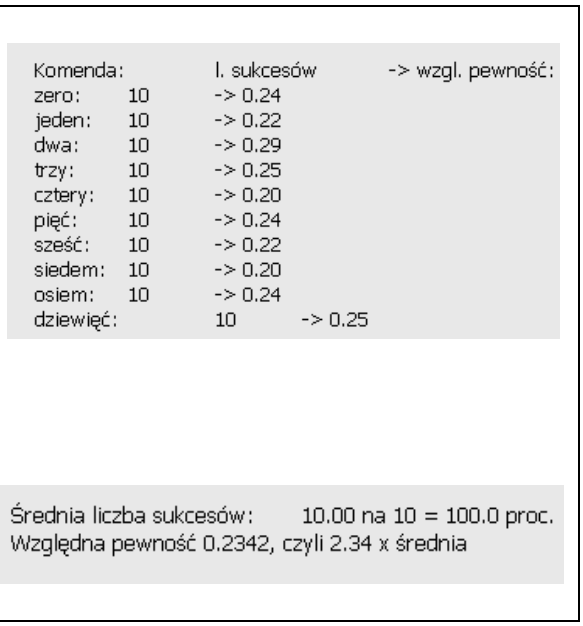
```
int CKLAMODoc::WykonajKlasyfikacje(double* ymin, double* ySrednia)
// Zwraca indeks zwycieskiej komendy dla aktualnej wypowiedzi (sekwencji)
// tablicę jakosci dopasowań obserwacji do wszystkich komend w modelu (rozpoznanie zdania),
// i tablicę jakości komendy w obserwacji (rozpoznanie słowa)
{ }
```

Liczba współczynników MFCC (12 czy 18?)

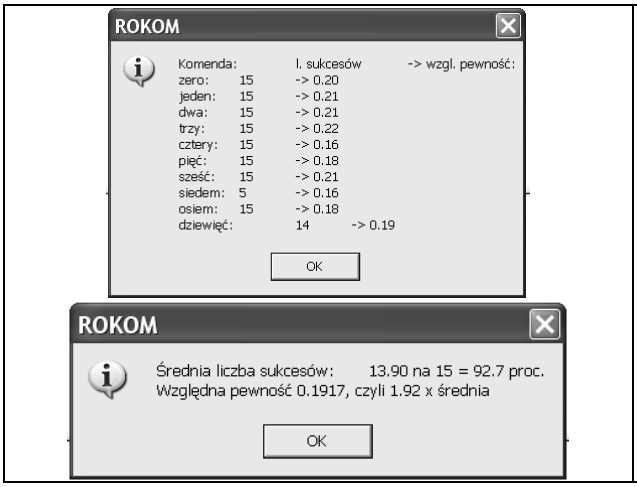
Rys. 4.2-4.5 pozwalają porównać skuteczność rozpoznawania dla zestawu 001m 10x25 (cyfry mówione) w sytuacji, gdy zastosowano 12 współczynników MFCC i 18 wsp. MFCC. W literaturze przedmiotu zazwyczaj mówi się o stosowaniu 12-18 współczynników podstawowych MFCC. Przy wyborze tej liczby należy kierować się jej relacją do liczby pasm melowych. Rysunki wskazują, że 12 współczynników nie zapewnia pełnej skuteczności rozpoznawania mowy nawet dla tego prostego przypadku. Wprawdzie na etapie tworzenia cech średnich po zwiększeniu liczby współczynników trudniej jest zapewnić odpowiedni odstęp reprezentantom klas między sobą (rys. 4.2-4.3), ale wyniki klasyfikacji nowych wypowiedzi poprawiają się (rys. 4.4-4.5).



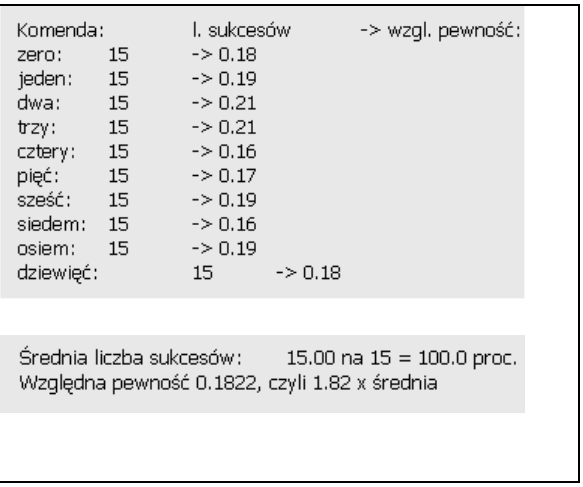
Rys. 4.2 Wyniki zbiorcze klasyfikacji 10 próbek uczących przy 12 cechach MFCC



Rys. 4.3 Wyniki zbiorcze klasyfikacji 10 przy 18 cechach MFCC



Rys. 4.4 Wyniki zbiorcze klasyfikacji 15 próbek dodanych (nieznanych w fazie uczenia) przy 12 wsp. MFCC



Rys. 4.5 Wyniki zbiorcze klasyfikacji 15 próbek dodanych (nieznanych w fazie uczenia) przy 18 wsp. MFCC

Klasyfikacja zdań zestawu „001m (20x25)”

Modele klas (w postaci sekwencji cech) uczone były na podstawie części dostępnych próbek, Następnie te same próbki i nowe próbki (druga część zestawu próbek) były klasyfikowane z metodą dopasowania DTW. Wynikiem dopasowania pojedynczego zdania z modelem klas był wektor „jakości dopasowania”. Następnie ustalana była „względna pewność” klasyfikacji w wyniku podzielenia pojedynczej wartości dopasowania do danej klasy przez sumę wszystkich dopasowań danego zdania do wszystkich klas. Średnia względna pewność w przypadku 20 klas wynosi „1/20 = 0.05”. Za sukces klasyfikacji uznawano fakt spełnienia jednocześnie dwóch kryteriów:

1. dopasowanie zdania do właściwej klasy uzyskało najwyższą względną pewność i
2. najwyższa pewność > 125% średniej względnej pewności.

Tab. 4.1 Wyniki zbiorcze klasyfikacji 10 próbek uczących

Komenda:	l. sukcesów	-> wzgl. pewność:
1: zero: 10	-> 0.14	
2: jeden:	10	-> 0.14
3: dwa: 10	-> 0.14	
4: trzy: 10	-> 0.14	
5: cztery:	10	-> 0.12
6: pięć: 10	-> 0.15	
7: sześć:	10	-> 0.14
8: siedem:	10	-> 0.13
9: osiem:	10	-> 0.14
10: dziewięć:	10	-> 0.17
11: start:	10	-> 0.12
12: stop:	10	-> 0.16
13: lewo:	10	-> 0.15
14: prawo:	10	-> 0.16
15: góra:	10	-> 0.18
16: dół: 10	-> 0.19	
17: puść:	10	-> 0.15
18: złap:	10	-> 0.16
19: oś: 10	-> 0.14	
20: chwytak:	10	-> 0.14

Średnia liczba sukcesów: 10.00 na 10 = 100.0 proc.
Względna pewność 0.1478, czyli 2.96 x średnia

Tab. 4.2 Wyniki zbiorcze klasyfikacji 15 próbek dodanych (nieznanych podczas uczenia)

Komenda:	l. sukcesów	-> wzgl. pewność:
1: zero: 14	-> 0.10	
2: jeden:	15	-> 0.12
3: dwa: 15	-> 0.09	
4: trzy: 15	-> 0.11	
5: cztery:	15	-> 0.09
6: pięć: 15	-> 0.09	
7: sześć:	15	-> 0.11
8: siedem:	15	-> 0.10
9: osiem:	15	-> 0.11
10: dziewięć:	14	-> 0.11
11: start:	13	-> 0.09
12: stop:	15	-> 0.11
13: lewo:	15	-> 0.10
14: prawo:	15	-> 0.10
15: góra:	15	-> 0.10
16: dół: 15	-> 0.11	
17: puść:	15	-> 0.11
18: złap:	13	-> 0.10
19: oś: 15	-> 0.11	
20: chwytak:	15	-> 0.10

Średnia liczba sukcesów: 14.70 na 15 = 98.0 proc.
Względna pewność 0.1029, czyli 2.06 x średnia

Klasyfikacja zdań zestawu „Sylwia” (30x10)

Tab. 4.3 Wyniki zbiorcze klasyfikacji 5 próbek uczących

Komenda:	l. sukcesów	-> wzgl. pewność:
zd 1 "Bilety i rezerwacje":	5	-> 0.08
zd 2 "Programy lojalnościowe":	5	-> 0.08
zd 3 "Przewóz bagażu":	5	-> 0.09
zd 4 "Podróż samolotem":	5	-> 0.09
zd 5 "Odprawa i dokumenty":	5	-> 0.08
zd 6 "Informacje dla pasażerów udających się do Stanów Zjednoczonych o przekazywaniu ich danych osobowych":	5	-> 0.16
zd 7 "Jak kupić":	5	-> 0.13
zd 8 "Jak rezerwować":	5	-> 0.09
zd 9 "Jak płacić i odebrać":	5	-> 0.08
zd 10 "Ceny biletów":	5	-> 0.11
zd 11 "Nie interesują mnie te informacje":	5	-> 0.09
zd 12 "Przez internet":	5	-> 0.12
zd 13 "Telefonicznie przez kol senter":	5	-> 0.08
zd 14 "W biurze lotu lub u agenta sprzedaży":	5	-> 0.09
zd 15 "Chcę kupić bilet ale nie dla siebie":	5	-> 0.08
zd 16 "Nie chcę kupić biletu, chcę wiedzieć jaka jest cena":	5	-> 0.09
zd 17 "Powrót":	5	-> 0.17
zd 18 "Jakie są to ograniczenia":	5	-> 0.08
zd 19 "Chcę kupić bilet":	5	-> 0.10
zd 20 "Czy zakup biletu przez internet jest bezpieczny?":	5	-> 0.10
zd 21 "Czy mogę w Internecie kupić bilet tylko dla dziecka?":	5	-> 0.08
zd 22 "Czy mogę zmienić nazwisko na bilecie?":	5	-> 0.09
zd 23 "Mam problem z zakupem biletu przez Internet":	5	-> 0.09
zd 24 "Co to jest bilet elektroniczny?":	5	-> 0.08
zd 25 "Czy zawsze trzeba się logowac aby kupic bilet?":	5	-> 0.09
Komenda:	l. sukcesów	-> wzgl. pewność:
zd 26 "Powiedz mi więcej o SSL":	5	-> 0.08
zd 27 "Nie mogę się zalogować":	5	-> 0.08
zd 28 "Kiedy próbuję kupić bilet na następny dzień system wyświetla błąd":	5	-> 0.11
zd 29 "Jakie korzyści daje rejestracja?":	5	-> 0.09
zd 30 "Dlaczego nie mogę się zalogować?":	5	-> 0.08

Średnia liczba sukcesów: 5.00 na 5 = 100.0 proc.
Względna pewność 0.0955, czyli 2.86 x średnia

Tab. 4.4 Wyniki zbiorcze klasyfikacji 5 próbek dodatkowych

Komenda:	l. sukcesów	-> wzgl. pewność:
zd 1 "Bilety i rezerwacje":	5	-> 0.07
zd 2 "Programy lojalnościowe":	4	-> 0.06
zd 3 "Przewóz bagażu":	4	-> 0.07
zd 4 "Podróż samolotem":	5	-> 0.08
zd 5 "Odprawa i dokumenty":	4	-> 0.07
zd 6 "Informacje dla pasażerów udających się do Stanów Zjednoczonych o przekazywaniu ich danych osobowych":	5	-> 0.12
zd 7 "Jak kupić":	5	-> 0.11
zd 8 "Jak rezerwować":	5	-> 0.07
zd 9 "Jak płacić i odebrać":	5	-> 0.07
zd 10 "Ceny biletów":	5	-> 0.10
zd 11 "Nie interesują mnie te informacje":	5	-> 0.08
zd 12 "Przez internet":	5	-> 0.11
zd 13 "Telefonicznie przez kol senter":	5	-> 0.07
zd 14 "W biurze lotu lub u agenta sprzedaży":	5	-> 0.07
zd 15 "Chcę kupić bilet ale nie dla siebie":	5	-> 0.07
zd 16 "Nie chcę kupić biletu, chcę wiedzieć jaka jest cena":	5	-> 0.07
zd 17 "Powrót":	5	-> 0.14
zd 18 "Jakie są to ograniczenia":	5	-> 0.08
zd 19 "Chcę kupić bilet":	5	-> 0.08
zd 20 "Czy zakup biletu przez internet jest bezpieczny?":	5	-> 0.08
zd 21 "Czy mogę w Internecie kupić bilet tylko dla dziecka?":	5	-> 0.07
zd 22 "Czy mogę zmienić nazwisko na bilecie?":	5	-> 0.07
zd 23 "Mam problem z zakupem biletu przez Internet":	5	-> 0.07
zd 24 "Co to jest bilet elektroniczny?":	5	-> 0.06
zd 25 "Czy zawsze trzeba się logowac aby kupic bilet?":	5	-> 0.07
Komenda:	l. sukcesów	-> wzgl. pewność:
zd 26 "Powiedz mi więcej o SSL":	5	-> 0.07
zd 27 "Nie mogę się zalogować":	5	-> 0.07
zd 28 "Kiedy próbuję kupić bilet na następny dzień system wyświetla błąd":	5	-> 0.09
zd 29 "Jakie korzyści daje rejestracja?":	5	-> 0.07
zd 30 "Dlaczego nie mogę się zalogować?":	5	-> 0.07

Średnia liczba sukcesów: 4.90 na 5 = 98.0 proc.
Względna pewność 0.0783, czyli 2.35 x średnia

Klasyfikacja zdań zestawu „Michał” (20x10)

Tab. 4.5 Wyniki zbiorcze klasyfikacji 5 próbek uczących

```
Komenda:            l. sukcesów         -> wzgl. pewność:
zd 31 "Informacje o cenach":            5           -> 0.12
zd 32 "Informacje o bagaZu":            5           -> 0.12
zd 33 "Moja przgloNdarka nie akceptuje kukis":   5          -> 0.12
zd 34 "Co to soN kukis?":       5           -> 0.16
zd 35 "Jak inaczej kupić bilet?":       5           -> 0.12
zd 36 "Jak kupić bilet?":       5           -> 0.13
zd 37 "Co vchodzi vskwad ceny biletu?":         5           -> 0.11
zd 38 "Dlaczego varunki taryfy na stronie internetowej soN w jeNzyku angielskim?":      5
-> 0.17
zd 39 "Dlaczego cena z kalendarza cenowego jest inna na kolejnym ekranie?":             5
-> 0.15
zd 40 "Jakie zniZki soN oferowane? ":   5           -> 0.12
zd 41 "O ofercie dla grup":     5           -> 0.14
zd 42 "Gdzie znajdeN informacjeN na temat taryfy?":         5           -> 0.10
zd 43 "Zmiana rezervacji lub jej anulowanie":       5           -> 0.12
zd 44 "Potvierdzenie rejsu":    5           -> 0.14
zd 45 "PodgloNd ve vwasnoN rezervacjeN":        5           -> 0.11
zd 46 "Rezervacja miejsc w samolocie":          5           -> 0.12
zd 47 "Jak anulować lub zmienić rezervacjeN":   5           -> 0.13
zd 48 "Zvrot pienieNdzy na nievykorzystany bilet":          5           -> 0.12
zd 49 "LeceN do uesa kanady":           5           -> 0.12
zd 50 "Nie leceN do uesa kanady":       5           -> 0.11
```

```
Średnia liczba sukcesów:      5.00 na 5 = 100.0 proc.
Względna pewność 0.1262, czyli 2.52 x średnia
```

Tab. 4.6 Wyniki zbiorcze klasyfikacji 5 próbek dodatkowych

```
Komenda:            l. sukcesów         -> wzgl. pewność:
zd 31 "Informacje o cenach":            5           -> 0.11
zd 32 "Informacje o bagaZu":            5           -> 0.11
zd 33 "Moja przgloNdarka nie akceptuje kukis":   5          -> 0.09
zd 34 "Co to soN kukis?":       5           -> 0.11
zd 35 "Jak inaczej kupić bilet?":       5           -> 0.10
zd 36 "Jak kupić bilet?":       5           -> 0.12
zd 37 "Co vchodzi vskwad ceny biletu?":         5           -> 0.08
zd 38 "Dlaczego varunki taryfy na stronie internetowej soN w jeNzyku angielskim?":      5
-> 0.13
zd 39 "Dlaczego cena z kalendarza cenowego jest inna na kolejnym ekranie?":             5
-> 0.11
zd 40 "Jakie zniZki soN oferowane? ":   5           -> 0.10
zd 41 "O ofercie dla grup":     5           -> 0.11
zd 42 "Gdzie znajdeN informacjeN na temat taryfy?":         5           -> 0.09
zd 43 "Zmiana rezervacji lub jej anulowanie":       5           -> 0.10
zd 44 "Potvierdzenie rejsu":    5           -> 0.11
zd 45 "PodgloNd ve vwasnoN rezervacjeN":        5           -> 0.09
zd 46 "Rezervacja miejsc w samolocie":          5           -> 0.09
zd 47 "Jak anulować lub zmienić rezervacjeN":   5           -> 0.10
zd 48 "Zvrot pienieNdzy na nievykorzystany bilet":          5           -> 0.10
zd 49 "LeceN do uesa kanady":           5           -> 0.10
zd 50 "Nie leceN do uesa kanady":       5           -> 0.09
```

```
Średnia liczba sukcesów:      5.00 na 5 = 100.0 proc.
Względna pewność 0.1010, czyli 2.02 x średnia
```

Klasyfikacja zdań zestawu „Olga” (20x10)

Tab. 4.7 Wyniki zbiorcze klasyfikacji 5 próbek uczących

Komenda:	l. sukcesów	-> wzgl. pewność
zd 51 : 5	-> 0.11	
zd 52: 5	-> 0.14	
zd 53: 5	-> 0.11	
zd 54: 5	-> 0.13	
zd 55 : 5	-> 0.12	
zd 56: 5	-> 0.11	
zd 57: 5	-> 0.13	
zd 58: 5	-> 0.14	
zd 59 : 5	-> 0.12	
zd 60: 5	-> 0.11	
zd 61: 5	-> 0.13	
zd 62: 5	-> 0.12	
zd 63 : 5	-> 0.23	
zd 64: 5	-> 0.29	
zd 65: 5	-> 0.11	
zd 66: 5	-> 0.11	
zd 67: 5	-> 0.12	
zd 68: 5	-> 0.10	
zd 69: 5	-> 0.11	
zd 70: 5	-> 0.12	

Średnia liczba sukcesów: 5.00 na 5 = 100.0 proc.
Względna pewność 0.1334, czyli 2.67 x średnia

Komenda:	l. sukcesów	-> wzgl. pewność:
zd 51 : 5	-> 0.08	
zd 52: 5	-> 0.10	
zd 53: 3	-> 0.07	
zd 54: 5	-> 0.09	
zd 55 : 5	-> 0.10	
zd 56: 5	-> 0.08	
zd 57: 5	-> 0.12	
zd 58: 5	-> 0.10	
zd 59 : 5	-> 0.10	
zd 60: 5	-> 0.10	
zd 61: 5	-> 0.11	
zd 62: 5	-> 0.09	
zd 63 : 5	-> 0.18	
zd 64: 4	-> 0.21	
zd 65: 4	-> 0.08	
zd 66: 5	-> 0.09	
zd 67: 5	-> 0.10	
zd 68: 1	-> 0.07	
zd 69: 5	-> 0.10	
zd 70: 5	-> 0.11	

Średnia liczba sukcesów: 4.60 na 5 = 92.0 proc.
Względna pewność 0.1035, czyli 2.07 x średnia

4.3 WYNIKI KLASYFIKACJI DTW DLA WIELU MÓWCÓW

Przebadano dwa zestawy głosów:
1) 10 głosów (5 męskich i 5 żeńskich) o dość znacznym poziomie szumu w nagraniach (tzw. „zestaw WAT" [ADA00]),
2) trzy głosy (dwa żeńskie i jeden męski) o niskim poziomie szumu ale o przetworzonych charakterystykach – jeden głoś żeński (Olga) miał wytłumiony dolny zakres, a głos męski (Michał) - górny zakres (tzw. „zestaw LOT" [INT10]).

Zestaw WAT – 10 głosów

Dysponujemy próbkami słów (mówione cyfry i liczby) dla 10 mówców – 5 mężczyzn i 5 kobiet. Głosy męskie oznaczone są jako: 001m, 003m, 004m, 005m, 006m. Oznaczenie głosów kobiecych to: 002k, 018k, 020k, 025k, 028k.

Tab. 4.8 Dwadzieścia pięć klas (słów wzgl. par słów) wybranych z zestawu WAT:

Lp.	Kod klasy	Wypowiedź
1	10000	zero
2	10001	jeden
3	10002	dwa
4	10003	trzy
5	10004	cztery
6	10005	pięć
7	10006	sześć
8	10007	siedem
9	10008	osiem
10	10009	dziewięć
11	10010	start
12	10011	stop
13	10012	lewo
14	10013	prawo
15	10014	góra
16	10015	dół
17	10016	puść
18	10017	złap
19	10018	oś
20	10019	chwytak
21	30000	zero cztery
22	30001	siedemnaście
23	30002	czterdzieści dziewięć
24	30003	siedemdziesiąt dwa
25	30004	dziewięćdziesiąt trzy

Tab. 4.9 Wyniki klasyfikacji 10 klas (słowa 1-10), przy uczeniu 5 próbek/zdanie jednego mówcy. Podano procent rozpoznanych zdań i średni odstęp dopasowania dla prawidłowego reprezentanta (jako wielokrotność średniego odstępu):

klasyfikacja / uczenie	001m 1-5	001m 6-10	002k 1-5	003m 1-5	Podsumowanie
001m 1-5	98% x2.34	84% x1.89	46% x1.23	74% x1.40	w grupie mężczyzn ≥ 74%
002k 1-5	78% x1.36	82% x1.38	100% x2.24	68% x1.36	w grupie mężczyzn ≥ 68%
003m 1-5	76% x1.41	90% x1.54	58% x1.26	98% x2.24	w grupie mężczyzn ≥ 83%

Tab. 4.10 Wyniki klasyfikacji 20 klas (słowa 1-20), przy uczeniu 5 próbek/zdanie jednego mówcy. Podano procent rozpoznanych zdań i średni odstęp dopasowania dla prawidłowego reprezentanta (jako wielokrotność średniego odstępu):

klasyfikacja / uczenie	001m 1-5	001m 6-10	002k 1-5	003m 1-5	Podsumowanie
001m 1-5	99% x3.10	92% x2.47	45% x1.35	63% x1.49	w grupie mężczyzn ≥ 63%
002k 1-5	67% x1.43	66% x1.42	97% x2.63	62% x1.47	w grupie mężczyzn 62 – 66,5%
003m 1-5	56% x1.50	66% x1.54	37% x1.37	97% x2.57	w grupie mężczyzn ≥ 61%

Tab. 4.11 Wyniki klasyfikacji 25 klas (słowa 1-25), przy uczeniu dla jednego mówcy (próbki 1-5) - dla 5 głosów żeńskich i 5 męskich.

A) Głosy męskie (uczenie i klasyfikacja)

klasyfikacja / uczenie	001m 1-5	001m 6-10	003m 1-5	004m 1-5	005m 1-5	006m 1-5	Podsumowanie w grupie mężczyzn
001m 1-5	99.2% x3.51	93,6% x2.73	69,6% x1.71	73,6% x1.84	78,4% x1.88	82,4% x1.87	69% - 93%
003m 1-5	64% x1,74		97,6% x3,15	84% x1,93	84,8% x2,07	85,6% x2,05	64% – 85%

B) Głosy żeńskie (uczenie i klasyfikacja)

klasyfikacja / uczenie	002k 1-5	018k 1-5	020k 1-5	025k	028k	Podsumowanie w grupie kobiet
002k 1-5	97,6% x3.09	80% x1.80	80% x1.80	61,6% x1.51	52% x1.48	52% - 80%
018k 1-5	51,2% x1.56	100% x3.26	90,4% x2.00	67,2% x1.80	67,2% x1.76	51% - 90%

C) Uczenie głos męski – klasyfikacja głos żeński

klasyfikacja / uczenie	002k 1-5	018k 1-5	020k 1-5	025k	028k	Podsumowanie w grupie kobiet
001m 1-5	48% x1.45	69,6% x1.61	76% x1.74	48,8% x1.42	49,6% x1.46	48% - 76%
003m 1-5	36% x1,48	80% x1,71	81,6% x1,75	53,6% x1,50	64% x1,56	36% - 81%

Tab. 4.12 Wyniki klasyfikacji 25 klas (słowa 1-25), przy uczeniu 2 próbkami/zdanie jednocześnie wszystkimi 5 głosami męskimi lub wszystkimi 5 głosami żeńskimi lub wszystkimi 10 głosami.

A) Uczenie jednocześnie 5 głosami męskimi (próbki 1-2) – klasyfikacja wszystkich głosów

klasyfikacja / uczenie	001m 1-2 3-5	003m 1-2 3-5	004m 1-2 3-5	005m 1-2 3-5	006m 1-2 3-5	Podsumowanie w grupie mężczyzn
męskie 1-2	94% 90,7% ×2.31 x2.18	98% 86,7% ×2,27 x2.02	100% 86,7% ×2,40 x2,10	100% 94,7% ×2,32 x2,27	96% 98,7% ×2.33 x2,20	96-100% -próbki uczące 86-98% - nowe próbki

klasyfikacja / uczenie	002k 1-5	018k 1-5	020k 1-5	025k 1-5	028k 1-5	Podsumowanie w grupie kobiet
męskie 1-2	57,6% x1,53	80% x1,73	88% x1,86	66,4% x1,53	62,4 % x1,52	57 - 88% x1,52 – x1,86

B) Uczenie jednocześnie 5 głosami żeńskimi (próbki 1-2) – klasyfikacja wszystkich głosów

klasyfikacja / uczenie	001m 1-5	003m 1-5	004m 1-5	005m 1-5	006m 1-5	Podsumowanie w grupie mężczyzn
żeńskie 1-2	75,2% x1,65	82,4% x1,74	84.8% x1,76	86,4% x1,74	89,6 % x1,87	75 – 89%

klasyfikacja / uczenie	002k 1-2 3-5	018k 1-2 3-5	020k 1-2 3-5	025k 1-2 3-5	028k 1-2 3-5	Podsumowanie w grupie kobiet
żeńskie 1-2	96% 78,7% ×2,08 x1,83	98% 85,3% ×2,47 x2,15	94% 92% ×2,29 x2,10	90% 78,7% ×1,96 x1,93	88% 80% ×1,95 x1,82	88-98% -próbki uczące 78-92% - nowe próbki

C) Uczenie 10 głosami (męskie + żeńskie) (próbki 1-2) – klasyfikacja wszystkich głosów:

klasyfikacja / uczenie	001m 1-2 3-5	003m 1-2 3-5	004m 1-2 3-5	005m 1-2 3-5	006m 1-2 3-5	Podsumowanie w grupie mężczyzn
10 głosów (1-2)	84% 92% ×2,03 x1,96	98% 85,3% ×2.05 x1,89	100% 89.3% ×2,12 x1,95	100% 97,3% ×2,06 x2,05	98% 98,7% ×2.16 x2,09	84-100% -próbki uczące 85-98% - nowe próbki

klasyfikacja / uczenie	002k 1-2 3-5	018k 1-2 3-5	020k 1-2 3-5	025k 1-2 3-5	028k 1-2 3-5	Podsumowanie w grupie kobiet
10 głosów (1-2)	72% 62,7% ×1,74 x1,66	96% 84,0% ×2,10 x1,95	96% 90,7% ×2,15 x2,04	84% 82,7% ×1,77 x1,75	80% 81,3% ×1,74 x1,70	72-96% -próbki uczące 62-90% - nowe próbki

Zestaw LOT – zdania 1-50 – 3 głosy: sylwia, michał, olga

Tab. 4.13 Pięćdziesiąt klas (zdań) wybranych z zestawu LOT:

1 "Bilety i rezerwacje"
2 "Programy lojalnościowe"
3 "Przewóz bagażu"
4 "Podróż samolotem"
5 "Odprawa i dokumenty"
6 "Informacje dla pasażerów udających się do Stanów Zjednoczonych o przekazywaniu ich danych osobowych"
7 "Jak kupić"
8 "Jak rezerwować"
9 "Jak płacić i odebrać"
10 "Ceny biletów"
11 "Nie interesują mnie te informacje"
12 "Przez internet"
13 "Telefonicznie przez kol senter"
14 "W biurze lotu lub u agenta sprzedaży"
15 "Chcę kupić bilet ale nie dla siebie"
16 "Nie chcę kupić biletu, chcę wiedzieć jaka jest cena"
17 "Powrót"
18 "Jakie są to ograniczenia"
19 "Chcę kupić bilet"
20 "Czy zakup biletu przez internet jest bezpieczny?"
21 "Czy mogę w Internecie kupić bilet tylko dla dziecka?"
22 "Czy mogę zmienić nazwisko na bilecie?"
23 "Mam problem z zakupem biletu przez Internet"
24 "Co to jest bilet elektroniczny?"
25 "Czy zawsze trzeba się logować aby kupić bilet?"
26 "Powiedz mi więcej o eSeSeL"
27 "Nie mogę się zalogować"
28 "Kiedy próbuję kupić bilet na następny dzień system wyświetla błąd"
29 "Jakie korzyści daje rejestracja?"
30 "Dlaczego nie mogę się zalogować?"
31 "Informacja o cenach"
32 "Informacja o bagażu"
33 "Moja przeglądarka nie akceptuje kukis"
34 "Co to są kukis?"
35 "Jak inaczej kupić bile?"
36 "Jak kupić bilet?"
37 "Co wchodzi w skład ceny biletu?"
38 "Dlaczego warunki taryfy na stronie internetowej są w języku angielskim?"
39 "Dlaczego cena z kalendarza cenowego jest inna na kolejnym ekranie?"
40 "Jakie zniżki są oferowane?"
41 "O ofercie dla grup"
42 "Gdzie znajdę informacje na temat taryfy?"
43 "Zmiana rezerwacji lub jej anulowanie"
44 "Potwierdzenie rejsu"
45 "Wgląd we własną rezerwację"
46 "Rezerwacja miejsc w samolocie"
47 "Jak anulować lub zmienić rezerwację?"
48 "Zwrot pieniędzy za niewykorzystany bilet"

49 "Lecę do Stanów Zjednoczonych / Kanady"
50 "Nie lecę do Stanów Zjednoczonych/ Kanady"

Tab. 4.14 Wyniki klasyfikacji 10 zdań (zdania 1-10), przy uczeniu 5 próbek/zdanie jednego mówcy. Podano procent rozpoznanych zdań i średnią jakość dopasowania dla prawidłowego reprezentanta (jako wielokrotność średniej jakości):

klasyfikacja / uczenie	Sylwia 1-5	Sylwia 6-10	Michał 1-5	Michał 6-10	Olga 1-5	Olga 6-10	Podsumowanie
Sylwia 1-5	100% ×2.58	98% x2.21	53% x1.51		91% x1.65		w grupie kobiet ≥ 91%
Michał 1-5	72% x1.58		100% ×2.57	100% x2.23	64% x1.48		w grupie kobiet: 64 – 72%
Olga 1-5	95% x1.76		30% x1.44		100% ×2.76	96% x2.08	w grupie kobiet ≥ 95%

Tab. 4.15 Wyniki klasyfikacji 20 zdań (zdania 1-20), przy uczeniu 5 próbek/zdanie jednego mówcy. Podano procent rozpoznanych zdań i średnią jakość dopasowania dla prawidłowego reprezentanta (jako wielokrotność średniej jakości):

klasyfikacja / uczenie	Sylwia 1-5	Sylwia 6-10	Michał 1-5	Michał 6-10	Olga 1-5	Olga 6-10	Podsumowanie
Sylwia 1-5	100% ×2.89	98% x2.48	47,5% x1.55		90,5% x1.74		w grupie kobiet ≥ 90%
Michał 1-5	66,0% x1.65		100% ×2.87	100% x2.44	53,0% x1.53		w grupie kobiet: 53 – 66%
Olga 1-5	95,0% x1.86		26,5% x1.45		100% ×3.00	96% x2.15	w grupie kobiet ≥ 95%

Tab. 4.16 Wyniki klasyfikacji 30 zdań (zdania 1-30), przy uczeniu 5 próbek/zdanie jednego mówcy. Podano procent rozpoznanych zdań i średnią jakość dopasowania dla prawidłowego reprezentanta (jako wielokrotność średniej jakości):

klasyfikacja / uczenie	Sylwia 1-5	Sylwia 6-10	Michał 1-5	Michał 6-10	Olga 1-5	Olga 6-10	Podsumowanie
Sylwia 1-5	100% ×2.94	99,3% x2.51	58,3% x1.57		85 % x1.70		w grupie kobiet ≥ 85%
Michał 1-5	66.3% x1.65		100% ×2.95	100% x2.52	52,3% x1.52		w grupie kobiet: 52 – 66%
Olga 1-5	94% x1.81		31,7% x1.44		99,3% ×2.97	93,3% x2.10	w grupie kobiet ≥ 93%

Tab. 4.17 Wyniki klasyfikacji 50 zdań (zdania 1-50), przy uczeniu 5 próbek/zdanie jednego mówcy. Podano procent rozpoznanych zdań i średnią jakość dopasowania dla prawidłowego reprezentanta (jako wielokrotność średniej jakości):

klasyfikacja / uczenie	Sylwia 1-5	Sylwia 6-10	Michał 1-5	Michał 6-10	Olga 1-5	Olga 6-10	Podsumowanie
Sylwia 1-5	100% ×2.85	97,6% x2.44	30,0% x1.48		76,4% x1.67		w grupie kobiet ≥ 76%
Michał 1-5	56,2% x1.45		100% ×2.95	99,2% x2.50	50,2% x1.51		w grupie kobiet: 50 – 56%
Olga 1-5	86,4%% x1.76		18,8% x1.40		98,4% ×2.93	91,6% x2.15	w grupie kobiet ≥ 86,4%

Tab. 4.18 Wyniki klasyfikacji 50 zdań (1-50), przy uczeniu 5 próbkami (1-5) dwoma różnymi głosami jednocześnie.

A) Uczenie 2 głosami żeńskimi – **Sylwia + Olga** - (próbki 1-5) – i klasyfikacja wszystkich głosów i wypowiedzi:

klasyfikacja / uczenie	Sylwia 1-5	Sylwia 6-10	Michał 1-10	Olga 1-5	Olga 6-10	Podsumowanie w grupie kobiet
Sylwia + Olga 1-5	99,6% ×2.22	98,8% x2.10	48,8% x1.51	96.8% ×2.11	92,8% x1.88	96-99% -próbki uczące 92-98% - nowe próbki

B) Uczenie 2 głosami mieszanymi – **Sylwia + Michał** - (próbki 1-5) – i klasyfikacja wszystkich wypowiedzi (1-10):

klasyfikacja / uczenie	Sylwia 1-5	Sylwia 6-10	Michał 1-5	Michał 6-10	Olga 1-10	Podsumowanie - wszyscy
Sylwia + Michał (1-5)	96,8% ×2.04	91,6% x1.91	88,8% ×1.86	90,0% x1.84	75,4% x1.60	88-96% -próbki uczące 75-91% - nowe próbki

C) Uczenie wszystkimi 3 głosami – **Sylwia + Michał + Olga** - (próbki 1-5) – i klasyfikacja wszystkich wypowiedzi (1-10):

klasyfikacja / uczenie	Sylwia 1-5	Sylwia 6-10	Michał 1-5	Michał 6-10	Olga 1-5	Olga 6-10	Podsumowanie - wszyscy
Sylwia + Michał +Olga (1-5)	96,8% ×2.00	94,0% x1.93	76,8% ×1.69	81,6% x1.71	94.8% ×1.84	87,2% x1.73	76-96% -próbki uczące 81-94% - nowe próbki

Zestaw LOT – zdania 51-100 – 3 głosy: sylwia, michał, olga

Tab. 4.19 Pięćdziesiąt klas (zdania nr 51 – 100) wybranych z zestawu LOT:

51 Chcę zapłacić kartą Maestro lub Visa
52 Płacę przelewem
53 Odbiór przez inną osobę
54 Jak odebrać bilet?
55 Czy bilet może odebrać ktoś inny?
56 W jaki sposób stać się uczestnikiem programu?
57 Program fly and drive
58 Jak się zbiera mile
59 W jaki sposób chronione są moje dane osobowe?
60 Czy moja przeglądarka współpracuje z SSL?
61 Czy mogę zmienić adres w profilu użytkownika?
62 Co to jest kod PIN?
63 Tak
64 Nie
65 Jakie są korzyści z obsługi plików cookies?
66 Co się dzieje, gdy przeglądarka nie obsługuje cookies?
67 Jak przebiega odprawa i jakie dokumenty są potrzebne?
68 Jakie są warunki przewozu bagażu?
69 Jakie informacje muszę podać przy logowaniu?
70 Jakich informacji nie mogę zmienić?
71 Czy muszę aktualizować dane?
72 Jak zmienić kraj zamieszkania?
73 Z USA
74 Z innego kraju niż USA
75 Jak chronione są moje dane osobowe?
76 O programie Miles and More
77 Podróż samolotem

78 Jak skorzystać z atrakcyjnych stawek programu?
79 Jakie są stawki przy wynajmie auta w Polsce?
80 Czy możliwe jest zdobycie mil statusowych za przeloty wraz z partnerami Star Alliance?
81 Czy mile można kupić?
82 W jaki sposób mile są naliczane?
83 Czy mogę rozpocząć zbieranie mil przed potwierdzeniem zgłoszenia?
84 Jak naliczane są mile?
85 Jak przewieźć sprzęt sportowy?
86 Jakie są limity bagażu?
87 Czego nie wolno przewozić?
88 Nic mnie nie interesuje
89 Czy mogę zabrać ekwipunek narciarski?
90 Czy mogę przewieźć rower?
91 Jak przewieźć ze sobą zwierzę?
92 Czy mogę przewieźć wózek dziecięcy?
93 Jakie są limity bagażu bezpłatnego?
94 Jaka jest opłata za nadbagaż?
95 Czy wózek jest wliczony w limit bagażu bezpłatnego?
96 Jak przewieźć wózek dziecięcy, gdy brak jest wolnego miejsca na pokładzie?
97 Czego nie wolno przewozić w bagażu podręcznym?
98 Czy mogę przewieźć sprzęt narciarski?
99 Jaka jest stawka specjalna za przewóz sprzętu narciarskiego?
100 Jakie są opłaty za nadbagaż?

Tab. 4.20 Wyniki klasyfikacji 50 zdań (zdania 51-100), przy uczeniu 5 próbkami (1-5) dwoma różnymi głosami jednocześnie.

A) Uczenie 2 głosami żeńskimi – **Sylwia + Olga** - (próbki 1-5) – i klasyfikacja wszystkich głosów i wypowiedzi:

klasyfikacja / uczenie	Sylwia 1-5	Sylwia 6-10	Michał 1-10	Olga 1-5	Olga 6-10	Podsumowanie w grupie kobiet
Sylwia + Olga 1-5	99,6% ×2.29	99,2% x2.21	43,8% x1.57	96.4% ×2.17	90,8% x1.96	96-99% -próbki uczące 90-99% - nowe próbki

B) Uczenie 2 głosami mieszanymi – **Sylwia + Michał** - (próbki 1-5) – i klasyfikacja wszystkich wypowiedzi (1-10):

klasyfikacja / uczenie	Sylwia 1-5	Sylwia 6-10	Michał 1-5	Michał 6-10	Olga 1-10	Podsumowanie - wszyscy
Sylwia + Michał (1-5)	97,2% ×2.04	96,0% x2.00	86,0% ×1.95	90,4% x1.93	69,2% x1.63	86-97% -próbki uczące 69-96% - nowe próbki

C) Uczenie wszystkimi 3 głosami – **Sylwia + Michał + Olga** - (próbki 1-5) – i klasyfikacja wszystkich wypowiedzi (1-10):

klasyfikacja / uczenie	Sylwia 1-5	Sylwia 6-10	Michał 1-5	Michał 6-10	Olga 1-5	Olga 6-10	Podsumowanie - wszyscy
Sylwia + Michał +Olga (1-5)	97,2% ×2.04	95,6% x2.01	75,2% ×1.75	82,8% x1.79	90.4% ×1.88	84,8% x1.78	75-97% -próbki uczące 82-95% - nowe próbki

Wnioski

Podsumowanie i wnioski z testów przedstawiono w rozdziale końcowym. Porównano je także z rozpoznawaniem z modelem HMM. Można stwierdzić, że klasyfikator DTW charakteryzuje się dużą skutecznością w warunkach istnienia małej liczby próbek uczących. Ma w tym zakresie przewagę nad klasyfikatorem HMM i może być efektywnie stosowany wtedy, gdy liczba zdań nie jest zbyt duża (ok. 50 zdań dla wielu mówców lub ok. 100 zdań dla jednego mówcy).

5. Algorytm klasteryzacji i kodowania wektorów cech

Dla rozwiązania problemu kwantyzacji wektorów cech zaproponujemy algorytm wzorowany na **algorytmie LBG** - wyznacza on podobszary w przestrzeni cech i ich prototypy (słownik kodów). Dla podziału przestrzeni cech na klastry (grupy) przy zadanej oczekiwanej liczbie klastrów odwołuje się do funkcji **klasteryzacji**. Podstawowe modyfikacje algorytmy klasteryzacji to: stosowanie ważonej miary Mahalanobisa dla odległości dwóch wektorów cech, i wyróżnienie klasy (o indeksie zero) reprezentującą „ciszę". Dla takiego wektora jego odległość od reprezentanta klasy „cisza" wyznaczana jest jedynie na podstawie różnic jednej składowej wektora – składowej o indeksie „zero" reprezentującej energię okna, dla którego obliczono ten wektor cech.

Istotnymi autorskimi elementami procesu kwantyzacji są:

1) inicjalizacja klastrów dla kategorii podfonemowych w podprzestrzeni cech dodatkowych, w wyniku wstępnego założenia maksymalnej liczby kategorii i stopniowego redukowania tej liczby do rzeczywistych klastrów zawierających próbki wektorowe,

2) zastosowanie w trakcie kwantyzacji dwóch etapów klasteryzacji wektorów cech. W pierwszym etapie uwzględniane są jedynie cechy dodatkowe. Pozwalają one pogrupować (uzyskać klastry) wektory w kategorie odpowiadające niejawnie typom głosek, np. samogłoski, spółgłoski ustne, itd. W drugim etapie, w ramach każdego klastra kategorii, stosujemy ponownie klasteryzację, ale tym razem opartą o cechy podstawowe (MFCC + gradienty MFCC).

3) wprowadzenie szeregu kryteriów dla kończenia iteracji w procesie kwantyzacji. Możliwe jest zakończenie kwantyzacji i ustalenie słownika kodowego w wyniku zbiegania się sumarycznego błędu kwantyzacji lub zmniejszenie się tego błędu poniżej zadanego progu lub przekroczenia zadanej maksymalnej liczby klastrów.

Implementacja procesów kwantyzacji i kodowania wektorów cech w programie **KlaMo** zawarta jest w pliku **KoderCech.cpp**.

5.1 ALGORYTM KLASTERYZACJI DLA PRZEWIDYWANEJ LICZBY KLAS I PRZESTRZENI CECH

Dane są:
- zbiór wektorów cech – próbek uczących: $\boldsymbol{\omega} = \{\mathbf{c}^i \mid i = 1,..., N\}$,
- przewidywana liczba klas K,
- początkowy słownik $\mathbf{Z}^{(0)}$ złożony z wektorów prototypów $\mathbf{Z}^{(0)}=\{\mathbf{z}^{(0)}_\kappa, \kappa=1,...,K\}$,
- początkowy błąd aproksymacji $\mathcal{E}^{(0)}$,
- odchylenia standardowe $\{\sigma_j\}$dla każdej składowej $\{c_j\}$wektora cech, liczone na podstawie zbioru próbek;.
- zadany próg minimalnej energii MIN_ENERGIA, dla wyznaczenia ciszy;
- zadany próg Θ dla względnego błędu aproksymacji.

Algorytm
1) Iteruj kroki (a)-(d) dla $I = 1,2,3, ...$
   (a) Zaklasyfikuj wektory próbek $\mathbf{c}^i \in \boldsymbol{\omega}$ **nie reprezentujące ciszy** zgodnie z aktualnym słownikiem kodowym $\mathbf{Z}^{(I-1)}$ – wybierając klasę odpowiadającą najmniejszej odległości

(w sensie ważonej metryki Mahalanobisa, tzn. sumy ważonych odległości wzdłuż poszczególnych osi normalizowanych (podzielonych) przez odchylenia standardowe poszczególnych cech) wektora cech od reprezentanta klasy – i wyznacz sumaryczny błąd klasyfikacji:

$$\varepsilon^{(I)} = \sum_{i=1}^{N} \min_{\kappa} \mathbf{w} \,|\, \mathbf{c}_i - z_{\kappa}^{(I-1)} \,|_{Mahalanobis} \tag{5.1}$$

gdzie **w** jest wektorem wag poszczególnych cech.

Jeśli (c_0^i < MIN_ENERGIA) to wektor bezwarunkowo reprezentuje „ciszę" i zaliczany jest do klasy o indeksie „zero". Wtedy jego błąd klasyfikacji wynosi :

$$\frac{|\, c_0^i - z_0^{(i-1)} \,|}{\sigma_0} \tag{5.2}$$

(b) Jeśli względna różnica błędów jest poniżej zadanego progu, tzn.

$$\frac{|\, \varepsilon^{(I)} - \varepsilon^{(I-1)} \,|}{\varepsilon^{(I-1)}} < \Theta \tag{5.3}$$

to przejdź do kroku KONIEC.

(c) Oblicz nowy słownik kodowy $\mathbf{Z}^{(I)}$ z prototypami klas $\mathbf{z}^{(I)}{}_{\kappa}$:

$$\mathbf{z}_k^{(I)} = \frac{1}{N_k^{(I)}} \sum_{c \in \Omega_k^{(I)}} \mathbf{c} \tag{5.4}$$

(d) Ustaw i= i+1 i ponów iterację od kroku (a).

(KONIEC) Znaleziony słownik kodowy to $\mathbf{Z}^{(I)}$ przy błędzie aproksymacji $\varepsilon^{(I)}$.

W praktyce wykonujemy powyższy algorytm klasteryzacji ustawiając warunek stopu na 0.05, co pozwala na jego zatrzymanie po wykonaniu kilku iteracji. Kolejne iteracje wprowadzają coraz większy porządek w próbkach. Z czasem ruchy centroidów i klasy próbek przestają się zmieniać i zbiór próbek można uznać za sklasyfikowany.

Implementacja algorytmu klasteryzacji wykonana jest niejako dwukrotnie:

- w postaci funkcji **KlasteryzacjaCech()**, wykonywanej na etapie kwantyzacji dla kategorii i
- funkcji **KlasteryzacjaCechWgKat(),** wykonywanej w celu znalezienia klastrów podfonemowych.

Poniżej przedstawiamy pierwszą z nich. Druga różni się głównie parametrami wywołania – identyfikatorem klastra dla kategorii. Możliwa jest integracja obu metod, której ze względu na zachowanie większej efektywności kodu i zachowania mozliwośći zmiany algorytmu nie wykonaliśmy.

Algorytm kwantyzacji wektorowej (tworzenia słownika kodowego)

Algorytm kwantyzacji wektorowej [DUD02] korzysta z uprzednio przedstawionych funkcji klasteryzacji cech. Jego głównym zadaniem jest odpowiednie sterowanie inicjalizacją klastrów i liczbą klastrów. Jak już wspomnieliśmy, w naszym algorytmie kwantyzacji można wyróżnić trzy etapy:

1. inicjalizacji klastrów dla kategorii głossek (w implementacji oznaczone jako krok 0,
2. klasteryzacji dla kategorii w oparciu o cechy dodatkowe w wektorze cech (krok oznaczony jako 1) i
3. niezależnych względem siebie klasteryzacji wykonywanych w ramach każdej kategorii w oparciu o cechy MFCC i ich gradienty (krok oznaczany jako 2).

Dwa zasadnicze kroki klasteryzacji sterowane są w odmienny sposób. W kroku inicjalizacji liczba kategorii ustawiana jest na maksimum (przyjęliśmy tu 243 maksymalne kategorie). Jest to dużo więcej niż oczekujemy (ok. 8-10). Następnie w wyniku kolejnych klasteryzacji i redukcji ”pustych” klastrów ta liczba jest stopnowo zmiejszana. W kroku klasteryzacji do klas „podfonemowych” postępujemy odwrotnie z liczba klastrów. Rozpoczynamy od jednego klastra w ramach każdej kategorii i kolejno zwiększamy tę liczbę wywołując iteracyjnie algorytm klasteryzacji dla coraz większej liczby klastrów.

Implementację algorytmu może rozszerzyć wprowadzając sterowanie z poziomu modelu symbolicznego poprzez podanie:

1. liczby fonemów (minimalna liczba klastrów kategorii) i
2. liczby podfonemów (minimalna sumaryczna liczba klastrów podfonemowych).

Iteracje każdej klasteryzacji wykonywane są do momentu spełnienia któregoś z warunków końcowych: przekroczenia maksymalnej zadanej liczby klastrów lub osiągniecie zbieżności błędu kwantyzacji lub zmniejszenie się tego błędu poniżej zadanego progu.

Powiększanie liczby klastrów ma charakter „rozszczepiania” wybranego reprezentanta na dwóch równo odległych od niego. O wyborzez rozszczepianego klastra decyduje liczba próbek należących do danego klastra i wariancja tych próbek względem reprezentanta klastra.
Zasada rozszczepiania klastra: do rozszczepienia wybierana jest jedna cecha (powodująca największy błąd kwantyzacji) w jednym reprezentacie klastra (o największej liczbie przynależnych próbek).

Ideę jednego kroku kwantyzacji przedstawiono poniżej. Ilustruje on m.in. sposób korzystania z funkcji klasteryzacji cech i warunków kończenia iteracji.

Algortym pojedynczego kroku **kwantyzacji wektorowej**
DANE wejściowe:

- Zbiór wektorów cech – próbek uczących: $\boldsymbol{\omega} = \{\mathbf{c}_i \mid i = 1,..., N\}$,
- Zadany próg minimalnej energii MIN_ENERGIA, dla wyznaczenia ciszy;
- Próg dla minimalnego błędu Γ.
- Próg dla minimalnego błędu względnego Γ_1.
- Maksymalna liczba klastrów: MAX_Klastry .

KROKI przetwarzania:
1) Oblicz centroidy dwóch podzbiorów zbioru próbek $\boldsymbol{\omega}$: dla wektorów cech reprezentujących ciszę (c_0^i < MIN_ENERGIA) i nie-ciszę (w przeciwnym razie):

$$\boldsymbol{\mu}_0^{(0)} = \frac{1}{N}\sum_i \mathbf{c}_{(i)} \,, (c_0 < MIN\_ENERGIA) \qquad \boldsymbol{\mu}^{(0)} = \frac{1}{N}\sum_i \mathbf{c}_i \tag{5.5}$$

2) Oblicz odchylenia standarowe wszystkich składowych wektora cech dla próbek „nie-ciszy”..
3) Ustaw początkowy słownik $\mathbf{Z}^{(0)}$ dla K_0=2 klastrów i „ciszy”, gdzie prototypy obu klastrów „nie będących ciszą” obliczane są jako:

$$\mathbf{z}_{1,2} = (1 \pm \boldsymbol{\sigma})\,\boldsymbol{\mu}\,, \text{ gdzie } \boldsymbol{\sigma} \text{ - wektor odchylenia standardowego} \tag{5.6}$$

4) Iteraruj kroki (4)-(6) dla $K = K_0 , 2{\cdot}K_0 , 4{\cdot}K_0 , \ldots$
5) Wołaj **Klasteryzacja**(K, $\mathbf{Z}^{(0)}$) → da to nowy słownik $\mathbf{Z}^{(I)}$ opatrzony błędem $\varepsilon^{(I)}$.
6) IF ($\varepsilon^{(I)} < \Gamma$) THEN STOP.
7) IF ($\varepsilon^{(I)} / \varepsilon^{(I-1)} < \Gamma_1$) THEN STOP.
8) IF K > MAX_Klastry THEN STOP.
9) Zgodnie z zasadą rozszczepinaia wybierz reprezentanta $\mathbf{z}_\kappa$ klastra i cechę do rozszczepienia.

10) Ustaw nowy słownik $\mathbf{Z}^{(0)}$ zastępując $\mathbf{z}_\kappa$ przez 2·reprezentantów o prototypach:

$$\mathbf{z}_{\kappa,K+\kappa} = (1 \pm \boldsymbol{\sigma}\,)\,\mathbf{z}_\kappa \text{ , gdzie } \kappa=1,...,K \text{ i } \delta << 1. \qquad (5.7)$$

Przejdź do (4) i wykonuj następną iterację kroków (4-8).

Funkcja KwantyzatorCech()

Ta funkcja stanowi implementację naszego algorytmu kwantyzacji wektorowej w programie **KlaMo**.

```
//////////////////////
// Metoda klasy CKLAMODoc przeznaczona do kwantyzacja wektorowej
// wywołuje powyższe funkcje pomocnicze do 2-etapowej klasteryzacji
// ustala reprezentantów klastrów - słownik kodowy dla wektorów cech
int CKLAMODoc::KwantyzatorCech(double warStopu, int liczbaCechPodst, int liczbaCechDod,
                               long lProbek, int lWierszy,
                               double** probki,
                               int* klasaProbki, int *katProbki, int* katKlasy,
                               double** reprezent, double** reprezentKat)
{ …}
```

```
Ind 1; Elementy: [0] 6.4,  [1] -0.754, [2] -0.187, [12] -2.281, [1
Ind 2; Elementy: [0] 8.1,  [1] -0.771, [2] -0.213, [12] -1.540, [1
Ind 3; Elementy: [0] 8.7,  [1] -0.765, [2] -0.189, [12] -1.635, [1
Ind 4; Elementy: [0] 9.1,  [1] -0.776, [2] -0.234, [12] -0.993, [1
Ind 5; Elementy: [0] 17.2,  [1] -0.857, [2] -0.287, [12] -0.472, [
Ind 6; Elementy: [0] 19.5,  [1] -0.772, [2] -0.186, [12] -0.439, [
Ind 7; Elementy: [0] 8.9,  [1] -0.102, [2] -0.070, [12] -0.482, [1
Ind 8; Elementy: [0] 18.8,  [1] -0.400, [2] -0.079, [12] -0.378, [
Ind 9; Elementy: [0] 27.8,  [1] -0.801, [2] -0.236, [12] 0.133, [1
Ind 10; Elementy: [0] 36.5,  [1] -0.801, [2] -0.170, [12] 0.019, |
Ind 11; Elementy: [0] 34.0,  [1] -0.861, [2] -0.269, [12] -0.098,
Ind 12; Elementy: [0] 30.5,  [1] -0.802, [2] -0.220, [12] -0.144,
Ind 13; Elementy: [0] 8.1,  [1] -0.594, [2] -0.135, [12] 0.786, [1
Ind 14; Elementy: [0] 14.7,  [1] -0.629, [2] -0.093, [12] -0.214,
Ind 15; Elementy: [0] 22.1,  [1] -0.746, [2] -0.208, [12] -0.124,
Ind 16; Elementy: [0] 23.1,  [1] -0.696, [2] -0.106, [12] -0.122,
Ind 17; Elementy: [0] 11.7,  [1] -0.686, [2] -0.212, [12] 0.845, |
Ind 18; Elementy: [0] 13.5,  [1] -0.622, [2] -0.172, [12] 0.840, |
Ind 19; Elementy: [0] 20.6,  [1] -0.810, [2] -0.248, [12] 0.489, |
Ind 20; Elementy: [0] 20.9,  [1] -0.816, [2] -0.181, [12] 0.497, |
```

Rys. 5.1. Przykład wyników klasteryzacji: reprezentacji (cechy o indeksach 0, 1, 2, 12) klastrów od 1 do 20. W tym przykładzie wygenerowano reprezentantów dla 256 klas pod-fonemów.

Kodowanie sekwencji cech

W procesie klasyfikacji z wykorzystaniem dyskretnego modelu HMM wymagane są sekwencje kodów (klas pod-fonemów), najlepiej odpowiadających sekwencji numerycznych wektorów cech aktualnej obserwacji. Do tego celu służy funkcja programu KlaMo **WykonajJednoKodowanie**(). Posługuje się ona tą samą miarą odległości (ważona miara Mahalanobisa) wektora cech od reprezentanta klasy, co miara stosowana w algorytmie kwantyzacji wektorowej. Wynikiem działania funkcji jest wygenerowanie sekwencji kodów i miar jakości kodowania w pliku z rozszerzeniem **pfo** (funkcja **ZapiszJedneKody**()). Pliki tego typu są czytane przez program klasyfikatora HMM, stanowiąc dla niego źródło informacji o wypowiedziach - sygnale mowy.

5.2 MODEL FONETYCZNY JĘZYKA POLSKIEGO

Transkrypcja fonemowa

Fonemy (głoski) to podstawowe kategorie dźwięków mowy (fonów). Dwa dźwięki należą do tej samej kategorii (fonemu), jeśli są sobie wystarczająco bliskie - takie sformułowanie może powodować różne podejścia fonetyków.

Na potrzeby klasyfikatora mowy fonemy mogą być reprezentowane w dowolnym formacie. Istotne jest jedynie jednoznaczność zbioru fonemów i transkrypcji fonetycznych dla rozpoznawanych zdań i słów kluczowych. Przykładami takich zapisów są alfabety Worldbet [HIE94] i SAMPA [GRO01]. Ponieważ pojedynczy fonem może być reprezentowany więcej niż jedną literą, potrzebne jest ustalenie znaków oddzielających fonemy. Worldbet ujmuje symbole fonemów w nawiasy typu "slash", np. /y/, /E/, /s/, na potrzeby tej pracy wprowadźmy myślnik, czyli znak „-„. Poszczególne słowa oddzielać będziemy znakiem „Pauzy” „#”.

Zbiór podstawowych fonemów dla języka polskiego

Podamy zbiór symboli SAMPA opisujących 37 fonemów języka polskiego wyróżniony przez prof. Rocławskiego [WYD09] i dokonamy ich klasyfikacji na 1 z 8 interesujących nas klas fonetycznych:

Samogłoski

1. Dyftongi i normalne samogłoski

Lp.	Fonem	Notacja SAMPA	Notacja w pracy	Podział na trzy-fony
1	i	i	i	3
2	y	I	I	3
3	e	e	e	3
4	a	a	a	3
5	o	o	o	3
6	u	u	u	3
7	ę	eN	ę	3
8	ą	oN	ą	3

2. Skrócone monoftongi

Lp.	Fonem	Notacja SAMPA	Podział na trzy-fony
			2

Spółgłoski

3. Ustne spółgłoski, tzn. pół-samogłoski i sonanty. Są to dźwięki pośrednie między samogłoskami i spółgłoskami - występują tylko nieznaczne przeszkody w trakcie głosowym.

Lp.	Fonem	Notacja SAMPA	Notacja w pracy	Podział na trzy-fony
9.	j	j	j	2
10.	ł	w	w	2
11.	l	l	l	2
12.	r	r	r	2

4. Nosowe.

Zablokowanie przepływu powietrza przez jamę ustną, ale jednoczesne obniżenie podniebienia miękkiego umożliwia słaby przepływ przez nos.

Lp.	Fonem	Notacja SAMPA	Notacja w pracy	Podział na trzy-fony
13.	m	m	m	2
14.	n	n	n	2
15.	ń	n’	ń	2

5. Spiranty.
Świszczące dźwięki tworzone w wyniku zbliżania do siebie elementów artykulacji dźwięku powodujące zawirowania przepływu powietrza.

Lp.	Fonem	Notacja SAMPA	Notacja w pracy	Podział na trzy-fony	Uwaga
16.	f	f	f	1	bezdźwięczna
17.	s	s	s	1	bezdźwięczna
18.	w	v	v	2	dźwięczna
19.	z	z	z	2	dźwięczna
20.	h	x	x	1	bezdźwięczna
21.	ś	s’	ś	1	bezdźwięczna
22.	ź	z’	ź	2	dźwięczna
23.	sz	S	S	1	bezdźwięczna
24.	ż	Z	Z	2	dźwięczna

6. Zwarte.
Wybuchowy charakter dźwięku spowodowany najpierw zupełnym zamknięciem traktu wymowy a następnie raptownym zwolnieniem zamknięcia.

Lp.	Fonem	Notacja SAMPA	Podział na trzy-fony	Uwaga	Uwaga
25.	p	p	zw. +1	zwarcie krtaniowe p^h	bezdźwięczna
26.	t	t	zw. +2	zwarcie krtaniowe t^h	bezdźwięczna
27.	k	k	zw. +1	zwarcie krtaniowe k^h	bezdźwięczna
28.	b	b	zw. + 2	zwarcie krtaniowe p^h	dźwięczna
29.	D	D	zw. + 2	zwarcie krtaniowe t^h	dźwięczna
30.	g	g	zw. + 2	zwarcie krtaniowe k^h	dźwięczna

7. Afrykaty.
Głoski zwarte powodujące świst podczas zwalniania.

Lp.	Fonem	Notacja SAMPA	Notacja w pracy	Podział na trzy-fony	Uwaga
31.	c	ts	c	zw. +2	zwarcie krtaniowe t^h
32.	ć	ts’	ć	zw. +2	zwarcie krtaniowe t^h
33.	cz	tS	C	zw. +2	zwarcie krtaniowe t^h
34.	Dz	Dz	Dz	zw. + 2	zwarcie krtaniowe t^h
35.	dź	dz’	dź	zw. + 2	zwarcie krtaniowe t^h
36.	dż	dZ	dZ	zw. + 2	zwarcie krtaniowe t^h

8. Cisza (zdefiniowana z góry)
Pauza i (ewentualnie) zwarcia krtaniowe /t^h/, /k^h/, /p^h/.

Lp.	Fonem	Notacja SAMPA	Podział na trzy-fony
37.	pauza	#	1
38.	/t^h/		1
39.	/k^h/		1
40.	/p^h/		1

Dekompozycja na pod-fonemy

Efektem wspólnej artykulacji sąsiadujących ze sobą fonemów w słowie jest potrzeba zastosowania reprezentacji kontekstowej dla fonemów. W reprezentacji trzy-fonowej dzielimy każdą głoskę na jedną, dwie lub trzy części, w zależności od tego jak duży może być na nią wpływ sąsiednich dźwięków.

Wyróżniając grupy kontekstowe dla fonemów - zamiast rozpatrywać wpływ indywidualnych głosek rozpatrujemy wpływ grup głosek na wymawianą głoskę. Wyróżnimy osiem grup kontekstowych - w ich wyniku otrzymamy kilkaset (ok. 200-400) trzyczęściowych modeli fonemów. Dla lepszej czytelności notacji nazwę grupy poprzedza znak "$":

- **$Przed** - przednie samogłoski i zbliżone spektralnie ustne spółgłoski;
- **$Centr** - centralne samogłoski i zbliżone spektralnie ustne spółgłoski;
- **$Tylne** - tylne samogłoski i zbliżone spektralnie ustne spółgłoski;
- **$Cisza** - cisza, przerwa, zwarcie krtaniowe;
- **$Nos**- nosowe głoski;
- **$Retro** - głoski typu "retroflex" - kolorowane tylnym "r";
- **$Fric** – grupa frykatywów, czyli spirantów i afrykatów;
- **$Inne** - głoski zwarte i pozostałe ustne.

Przykład.
Model głoski /a/ w słowie "tak":
- /$Inne<a/ ("a" w kontekście poprzedzającej ją głoski zwartej lub pozostałej),
- /<a>/ ("a" ´środkowe, bez kontekstu),
- /a>$Inne/ ("a" w kontekście następującej głoski zwartej lub pozostałej).

```
Komenda 1: zero
Kolumna;  IndPFonemu; NzwPFonemu; i0, i1, i2, i13, i14. Wektor cech: i0, i1, i2, i13, i14
 0;  0;  /.pau/; 2.43, -0.181, -0.022, 111.441, -53.236. |||  1.77, -0.220, -0.049, 0.564, -0.256
 1;  18;  /$cisza<z/; 13.47, -0.622, 0.373, 0.840, -0.709. |||  5.83, -0.300, 0.112, 1.046, -0.651
 2;  17;  /$cisza<z/; 11.71, -0.686, 0.384, 0.845, -0.759. |||  9.27, -0.352, 0.310, 1.175, -0.917
 3;  30;  /z/; 14.95, -0.702, 0.349, 0.392, -0.300. |||  25.10, -0.710, 0.450, 0.453, -0.367
 4;  10;  z>$Przed/; 36.52, -0.801, 0.461, 0.019, 0.015. |||  40.33, -0.781, 0.476, 0.121, -0.037
 5;  12;  z>$Przed/; 30.51, -0.802, 0.479, -0.144, 0.188. |||  37.75, -0.690, 0.385, 0.034, 0.197
 6;  8;  /$Fric<e/; 18.81, -0.400, 0.100, -0.378, 0.652. |||  23.40, -0.382, 0.169, 0.046, 0.453
 7;  191;  /e/; 35.99, 0.145, -0.219, 0.262, 0.054. |||  36.22, 0.086, -0.231, 0.178, 0.174
 8;  191;  /e/; 35.99, 0.145, -0.219, 0.262, 0.054. |||  44.83, 0.084, -0.260, 0.207, 0.040
 9;  189;  /e>$Retro/; 49.72, 0.069, -0.228, 0.111, 0.006. |||  51.31, 0.063, -0.288, 0.151, -0.003
 10;  160;  /$Przed<r/; 57.69, -0.025, -0.375, 0.083, -0.012. |||  56.56, 0.035, -0.306, 0.076, -0.009
 11;  175;  /r/; 55.49, 0.046, -0.341, -0.104, 0.007. |||  59.69, 0.025, -0.329, -0.030, 0.009
 12;  175;  /r/; 55.49, 0.046, -0.341, -0.104, 0.007. |||  59.29, 0.030, -0.340, -0.131, 0.014
 13;  175;  /r/; 55.49, 0.046, -0.341, -0.104, 0.007. |||  52.08, 0.063, -0.345, -0.197, 0.016
 14;  169;  /r>Tylne$; 31.18, 0.125, -0.309, -0.240, -0.006. |||  32.59, 0.147, -0.322, -0.263, 0.044
 15;  184;  /$Retro<o/; 36.24, 0.207, -0.312, 0.056, 0.019. |||  30.42, 0.224, -0.294, 0.023, 0.036
 16;  184;  /$Retro<o/; 36.24, 0.207, -0.312, 0.056, 0.019. |||  38.22, 0.204, -0.289, 0.118, 0.020
 17;  181;  /$Retro<o/; 44.48, 0.126, -0.322, 0.211, 0.006. |||  39.93, 0.180, -0.281, 0.070, 0.008
 18;  241;  /o/; 36.19, 0.204, -0.220, -0.088, -0.013. |||  39.57, 0.183, -0.243, -0.059, -0.003
 19;  241;  /o/; 36.19, 0.204, -0.220, -0.088, -0.013. |||  36.05, 0.211, -0.211, -0.143, -0.012
 20;  241;  /o/; 36.19, 0.204, -0.220, -0.088, -0.013. |||  32.20, 0.220, -0.186, -0.205, -0.030
 21;  228;  /o>$cisza/; 20.60, 0.281, -0.089, -0.400, -0.093. |||  22.47, 0.240, -0.130, -0.391, -0.088
 22;  228;  /o>$cisza/; 20.60, 0.281, -0.089, -0.400, -0.093. |||  15.15, 0.241, -0.022, -0.537, -0.130
 23;  120;  ; 7.31, 0.125, 0.211, -0.860, -0.252. |||  6.66, 0.118, 0.151, -0.869, -0.244
 24;  0;  /.pau/; 2.43, -0.181, -0.022, 111.441, -53.236. |||  4.05, 0.027, 0.128, -0.942, -0.292
 25;  0;  /.pau/; 2.43, -0.181, -0.022, 111.441, -53.236. |||  2.35, -0.066, 0.038, -0.647, -0.211
 26;  0;  /.pau/; 2.43, -0.181, -0.022, 111.441, -53.236. |||  1.94, -0.152, -0.029, -0.257, -0.099
 27;  0;  /.pau/; 2.43, -0.181, -0.022, 111.441, -53.236. |||  2.00, -0.249, -0.005, -0.047, -0.033
```

Rys. 5.2 Model słowa “zero”, złożony z sekwencji 28 wektorów cech (kolumny po prawej stronie), i dopasowanej do niego sekwencji reprezentantów klas pod-fonemów (kolumny po lewej stronie). Automatyczna kwantyzacja cech wprowadziła pewną nadmiarowość liczby klastrów pod-fonemów.

6. Podsumowanie. Wnioski.

Opracowano prosty algorytm klasyfikatora mowy obejmujący analizę akustyczną i fonetyczną, przy założeniu ograniczenia jego zastosowania do rozpoznawania słów i sekwencji słów (izolowanych zdań). Wykonano implementacje elementów tego algorytmu pozwalające na badanie sygnałów mowy, ich testowanie, optymalizowanie ustawień parametrów tych algorytmów i poszukiwanie lepszych rozwiązań niektórych etapów analizy sygnału mowy.

6.1 ALGORYTM

W raporcie ograniczonio się do algorytmu klasyfikacji mowy o postaci klasyfikatora sekenecji cech z „marszczeniem czasu" (algorytm DTW). Wyróżniono dwa tryby pracy algorytmu – tryb uczenia (klasyfikatora sekwencji cech) i tryb aktywnej klasyfikacji zdań i słów.
W pierwszym etapie prac rozwinięto algorytm dostosowany do jednego mówcy i jednego mikrofonu, tzn. wymagający osobnego trenowania dla każdego mówcy i sprzętu użytego do akwizycji sygnału mowy. Skuteczność rozpoznawania znacznie spadnie, jeśli wystąpią duże zmienności czasu trwania tej samej wypowiedzi lub istotnej zmiany intonacji tego samego zdania, np. zamiast intonacji oczekiwanej dla zdania informującego wystąpi intonacja dla pytania.
W drugi etapie uzupełniono algorytm o etap normalizacji spektrogramu (w celu zmniejszenie rozrzutu wypowiedzi dla różnych głosów) i etap normalizacji cech MFCC (w celu zmniejszenia wpływu różnych mikrofonów).

Algorytm klasyfikacji jest niezależny od języka – wymaga jedynie wstępnego skonfigurowania klasyfikatora obejmującego:
1. podanie zestawu fonemów języka i ich przynależności do 7 ogólnie przyjętych kategorii fonetycznych,
2. podanie opisów rozpoznawanych zdań i słów kluczowych w postaci sekwencji wcześniej zdefiniowanych fonemów.

Pomimo korzystania z dość powszechnie w środowisku fachowym znanych metod podawanych w literaturze przedmiotu, opracowano algorytm klasyfikatora mowy o autorskim charakterze, modyfikując wiele etapów analizy sygnału mowy w unikalny sposób.

W wyniku własnej implementacji algorytmu i przeprowadzonych testów dopracowano się rozwiązań o praktycznym charakterze, zazwyczaj nigdzie nie publikowanych, takich, jak – liczba i zakres filtrów melowych, liczba współczynników MFCC, budowa wektora cech, wagi elementów wektora cech, dwu albo trzy-etapowa kwantyzacja wektorów cech.

Algorytm umożliwia zarówno klasyfikację całych zdań, jak i wyszukiwanie zadanych słów w zdaniach. Do obu zadań stosuje się zasadniczo te same funkcje uczenia i dopasowywania cech. Zmienia sie jedynie sposób oceny wiarygodności dopasowania. Wynika to z różnicy polegającej na tym, że w klasyfikacji zdań oczekiwaną długość wzorca wyznacza wektor obserwacji a w klasyfikacji słów – oczekiwana długość wzorca wynika z modelu danego słowa.

6.2 PODSUMOWANIE TESTÓW

Przedstawiono wyniki eksperymentów przeprowadzone na dwóch zestawach danych (WAT i LOT) o różnych liczbach zdań (klas) (jednorazowo klasyfikowano maksymalnie 25 typów zdań dla WAT i 50 dla LOT). Proces uczenia prowadzono na jednej połowie (lub części) zestawu danych, a proces klasyfikacji i rozpoznawania – na całości zestawu.

W warunkach uczenia i klasyfikacji pojedynczego mówcy skuteczność rozpoznawania wynosiła:

- dla nowych próbek - ponad 90%,
- a dla próbek uczących zwykle sięgała 98-100%.

Są to wyniki uzyskane dla stosunkowo niedużych zestawów danych i w praktyce należy liczyć się z dużo gorszą skutecznością klasyfikatora DTW ale z polepszeniem wyników klasyfikatora HMM. .

Podsumowanie testów klasyfikacji DTW

1. Klasyfikator DTW charakteryzuje się dużą skutecznością w warunkach istnienia małej liczby próbek uczących. Nie wymaga żadnej wiedzy o fonetyce języka, czyli jego stosowanie nie zależy od języka.

2. Skuteczność rozpoznawania klasyfikatora DTW zależy od liczby klas, ale przede wszystkim od tego próbki ilu głosów stosowane są podczas tworzenia modeli DTW.

3. Gdy model powstał na podstawie jednego głosu, to jedynie próbki tego głosu były rozpoznawane z wysoką skutecznością.
   Dla próbek WAT (25 klas) uczenie próbkami jednego głosu (pkt. 1.1) prowadziło do skuteczności średnich dla próbek tego samego mówcy wynoszących 97%-100% dla próbek uczących.
   Dla próbek LOT (50 zdań) odpowiednia skuteczność wynosiła (pkt. 1.2): 98%-100% dla próbek uczących i 91-97% dla nowych próbek.
   Także próbki dla głosów tej samej płci co głos „uczący" są z reguły rozpoznawane z lepszą skutecznością niż próbki dla głosu odmiennej płci. Jednak ta skuteczność znacznie się waha (w próbkach WAT: 64-93% dla głosów męskich i 51-90% dla głosów żeńskich, oraz w próbkach LOT: 76%-97% dla głosów żeńskich).

4. Gdy model powstał na podstawie wielu głosów jednej płci, to następuje pewne „rozmycie" modeli zdań w przestrzeni cech. Skutkuje to z jednej strony obniżeniem się 100% skuteczności maksymalnej, zwykle obserwowanej dla głosu uczącego, ale z drugiej strony podnosi się skuteczność „dolna" i średnia skuteczność rozpoznawania głosów danej płci znacznie wzrasta.
   Dla próbek WAT (25 klas) uczenie próbkami głosów męskich względnie żeńskich (pkt. 1.1) prowadziło do skuteczności średnich dla mówców tej samej płci, wynoszących: 88%-100% dla próbek uczących i 78-98% dla nowych próbek.
   Dla próbek LOT (50 zdań) uczenie próbkami 2 głosów żeńskich (pkt. 1.2, 1.3) prowadziło do skuteczności średnich dla tych głosów wynoszących: 96%-99% dla próbek uczących i 90-99% dla nowych próbek.

5. Najlepszą średnią skuteczność rozpoznawania uzyskujemy stosując próbki uczące pochodzące od głosów obu płci. Następuje dalsze „rozmycie" powstających modeli, ale wtedy też głosy obu płci rozpoznawane są jednocześnie z wyższą średnią skutecznością.
   - Dla próbek WAT (25 klas) uczenie próbkami wszystkich 10 głosów (pkt. 1.1) prowadziło do skuteczności średnich dla mówców wynoszących: 72%-100% dla próbek uczących i 62-98% dla nowych próbek.
   - Dla próbek LOT (50 zdań) uczenie próbkami 3 głosów (pkt. 1.2 i 1.3) prowadziło do skuteczności średnich dla mówców wynoszących: 75%-97% dla próbek uczących i 81-95% dla nowych próbek.

6.3 WNIOSKI

1. Wyłania się strategia pracy z dwoma, odrębnymi modelami i słownikami kodowymi cech, utworzonymi na podstawie próbek kilku osób:
    a. kobiet (F0: 200 – 250 Hz, średnia DP (dolnopasmowość): 0.1-0.5) lub
    b. mężczyzn (F0: 100 – 150 Hz, śDP: 0.5-0.9).

2. Uczenie początkowe należy przeprowadzić na podstawie próbek kilku osób jednocześnie. Przy korzystaniu z programu przez jedną osobę warto zastosować „adaptowanie się" się modelu DTW lub słownika kodowego dla modelu HMM do próbek głosu danej osoby.

3. Dla wystarczająco dużej liczby próbek uczących dyskretny klasyfikator HMM powinien osiągnąć skuteczność prezentowaną przez klasyfikator DTW, pomimo, że korzysta on ze znacznie zredukowanej informacji o sygnale (dyskretne kody) w porównaniu z klasyfikatorem DTW (wielowymiarowe wektory cech).

Literatura

[ADA00] B. Adamczyk, K. Adamczyk, K. Trawiński: *Zasób mowy ROBOT*. Biuletyn IAiR nr 12, Wojskowa Akademia Techniczna, Warszawa, 2000 (+2 CD-romy).

[BEN08] J. Benesty, M.M. Sondhi, Y. Huang (Eds.): *Springer Handbook of Speech Processing*. Springer-Verlag Berlin Heidelberg, 2008.

[CSL00] (Praca zbiorowa). *The CSLU Speech Toolkit*. Centre for Spoken Language Understanding, Oregon Graduate Institute of Science and Technology, USA, 2000.

[Dav07] A. David, S. Vassilvitskii, k-means++: the advantages of careful seeding, Proceedings of the 18th ACM-SIAM symposium on Discrete algorithms, 2007, 1027-1035.

[DUD02] R.O. Duda, P.E. Hart P.E., Stark: *Pattern classification and scene analysis*, 2nd edition, John Wiley & Sons, New York, 2002.

[GRO01] S. Grocholewski: *Statystyczne podstawy systemu ARM dla języka polskiego*, Rozprawy Nr. 362, Wyd. Politechniki Poznańskiej, 2001.

[HIE94] J.L. Hieronymous at al.: *Worldbet*. AT&T Bell Laboratories, 1994.

[HTK06] P. Woodland, G. Evermann, M. Gales: HTKBook, Cambridge University Engineering Department (CUED), 2000-2006, http://htk.eng.cam.ac.uk

[INT10] InteliWise SA: Zestaw próbek LOT. Warszawa, 2010. (informacja prywatna)

[JUN96] J.-C. Junqua, J.-P. Haton: *Robustness in automatic speech recognition*, Kluwer Academic Publications, Boston etc., 1996.

[KAS09] W. Kasprzak. *Rozpoznawanie obrazów i sygnałów mowy*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2009. http://www.wydawnictwopw.pl/

[PRZ12] P.Przybysz, W.Kasprzak: *Rozpoznawanie zdań w sygnale mowy z wykorzystaniem modelu HMM*. Raport badawczy. IAiIS 12-05, Instytut Automatyki i Informatyki Stosowanej Politechnikmi Warszawskiej, 2012.

[RAB93] L. Rabiner, B. Juang. *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[WAL04] W. Walker, P.Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, J. Woelfel. *Sphinx-4: A Flexible Open Source Framework for Speech Recognition*. SMLI TR2004-0811, SUN Microsystems Inc., 2004. http://cmusphinx.sourceforge.net/sphinx4/#whitepaper

[WYD09] S. Wydra. *Zastosowanie ukrytych modeli Markowa w aplikacjach głosowych dla mowy polskiej*. Rozprawa doktorska. Wydział Elektroniki i Technik Informacyjnych Politechniki Warszawskiej, Warszawa 2009.